1. Random Sampling

In [35]:

```
import random

# Random sampling from a list
data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,23,56,78,90,45,34,33,22,11,56,76,75,
sample = random.sample(data, 10)  # Choose 5 random elements
print("Random Sample:", sample)

```

```
Random Sample: [11, 6, 56, 8, 45, 1, 10, 76, 3, 22]
```

Random sampling on dataset

In [48]:

```
import seaborn as sns
df=sns.load_dataset('iris')
print(df.shape)
random_sample = df.sample(n=10,random_state=42)
print("Random Sample:\n", random_sample)

```

```
(150, 5)
Random Sample:
      sepal_length  sepal_width  petal_length  petal_width     species
73            6.1          2.8           4.7          1.2  versicolor
18            5.7          3.8           1.7          0.3      setosa
118           7.7          2.6           6.9          2.3   virginica
78            6.0          2.9           4.5          1.5  versicolor
76            6.8          2.8           4.8          1.4  versicolor
31            5.4          3.4           1.5          0.4      setosa
64            5.6          2.9           3.6          1.3  versicolor
141           6.9          3.1           5.1          2.3   virginica
68            6.2          2.2           4.5          1.5  versicolor
82            5.8          2.7           3.9          1.2  versicolor
```

2. Systematic Sampling

In [52]:

```
import numpy as np

# Systematic sampling
population = np.arange(1, 101)  # Data from 1 to 100
population
```

Out[52]:

```
array([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,
        14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,
        27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,  39,
        40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,  52,
        53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  65,
        66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,
        79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,
        92,  93,  94,  95,  96,  97,  98,  99, 100])
```

In [53]:

```
n = 10  # Sample every 10th element
systematic_sample = population[::n]
print("Systematic Sample:", systematic_sample)
```

```
Systematic Sample: [ 1 11 21 31 41 51 61 71 81 91]
```

Systematic sampling on dataset

In [54]:

```
interval = 10
systematic_sample = df.iloc[::interval]
print("Systematic Sample:\n", systematic_sample)
```

```
Systematic Sample:
     sepal_length  sepal_width  petal_length  petal_width     species
0             5.1          3.5           1.4          0.2      setosa
10            5.4          3.7           1.5          0.2      setosa
20            5.4          3.4           1.7          0.2      setosa
30            4.8          3.1           1.6          0.2      setosa
40            5.0          3.5           1.3          0.3      setosa
50            7.0          3.2           4.7          1.4  versicolor
60            5.0          2.0           3.5          1.0  versicolor
70            5.9          3.2           4.8          1.8  versicolor
80            5.5          2.4           3.8          1.1  versicolor
90            5.5          2.6           4.4          1.2  versicolor
100           6.3          3.3           6.0          2.5   virginica
110           6.5          3.2           5.1          2.0   virginica
120           6.9          3.2           5.7          2.3   virginica
130           7.4          2.8           6.1          1.9   virginica
140           6.7          3.1           5.6          2.4   virginica
```

3.Stratified Sampling

In [25]:

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

# Stratified sampling using scikit-learn
data = pd.DataFrame({
    'Category': ['A', 'A', 'A', 'B', 'B', 'C', 'C', 'C'],
    'Values': [1, 2, 3, 4, 5, 6, 7, 8]
})
train, test = train_test_split(data, test_size=0.5, stratify=data['Categor
print("Train Sample:\n", train)
print("Test Sample:\n", test)
```

```
Train Sample:
   Category  Values
2        A       3
5        C       6
3        B       4
0        A       1
Test Sample:
   Category  Values
7        C       8
4        B       5
1        A       2
6        C       7
```

Starified Sampling on Dataset

In [27]:

```
X = df.drop(columns=["species"])
y = df["species"]

# Stratified sampling: dividing the dataset into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3, stratify=df['species'],
                 random_state=42)
print("Train Sample:\n", X_train)
print("Test Sample:\n", X_test)

```

```
Train Sample:
      sepal_length  sepal_width  petal_length  petal_width
98             5.1          2.5           3.0          1.1
68             6.2          2.2           4.5          1.5
19             5.1          3.8           1.5          0.3
143            6.8          3.2           5.9          2.3
99             5.7          2.8           4.1          1.3
..             ...          ...           ...          ...
37             4.9          3.6           1.4          0.1
79             5.7          2.6           3.5          1.0
33             5.5          4.2           1.4          0.2
94             5.6          2.7           4.2          1.3
3              4.6          3.1           1.5          0.2

[105 rows x 4 columns]
Test Sample:
      sepal_length  sepal_width  petal_length  petal_width
107            7.3          2.9           6.3          1.8
63             6.1          2.9           4.7          1.4
133            6.3          2.8           5.1          1.5
56             6.3          3.3           4.7          1.6
127            6.1          3.0           4.9          1.8
140            6.7          3.1           5.6          2.4
53             5.5          2.3           4.0          1.3
69             5.6          2.5           3.9          1.1
20             5.4          3.4           1.7          0.2
141            6.9          3.1           5.1          2.3
14             5.8          4.0           1.2          0.2
38             4.4          3.0           1.3          0.2
108            6.7          2.5           5.8          1.8
116            6.5          3.0           5.5          1.8
28             5.2          3.4           1.4          0.2
148            6.2          3.4           5.4          2.3
57             4.9          2.4           3.3          1.0
10             5.4          3.7           1.5          0.2
23             5.1          3.3           1.7          0.5
18             5.7          3.8           1.7          0.3
97             6.2          2.9           4.3          1.3
7              5.0          3.4           1.5          0.2
75             6.6          3.0           4.4          1.4
104            6.5          3.0           5.8          2.2
138            6.0          3.0           4.8          1.8
51             6.4          3.2           4.5          1.5
84             5.4          3.0           4.5          1.5
93             5.0          2.3           3.3          1.0
66             5.6          3.0           4.5          1.5
35             5.0          3.2           1.2          0.2
134            6.1          2.6           5.6          1.4
132            6.4          2.8           5.6          2.2
85             6.0          3.4           4.5          1.6
49             5.0          3.3           1.4          0.2
111            6.4          2.7           5.3          1.9
40             5.0          3.5           1.3          0.3
42             4.4          3.2           1.3          0.2
2              4.7          3.2           1.3          0.2
43             5.0          3.5           1.6          0.6
77             6.7          3.0           5.0          1.7
```

```
55              5.7            2.8            4.5            1.3
22              4.6            3.6            1.0            0.2
106             4.9            2.5            4.5            1.7
147             6.5            3.0            5.2            2.0
58              6.6            2.9            4.6            1.3
```

4. Cluster Sampling

In [29]:

```python
import numpy as np

# Cluster sampling: Select clusters randomly
data = {
    'Cluster1': [1, 2, 3,7,8,9,0],
    'Cluster2': [4, 5, 6,6,5],
    'Cluster3': [7, 8, 9,8,9],
    'Cluster4':[6,8,9,11,22]
}
selected_clusters = np.random.choice(list(data.keys()), 2, replace=False)
cluster_sample = [data[cluster] for cluster in selected_clusters]
print("selected clusters:", selected_clusters)
print("Cluster sample:",cluster_sample)
```

```
selected clusters: ['Cluster2' 'Cluster1']
Cluster sample: [[4, 5, 6, 6, 5], [1, 2, 3, 7, 8, 9, 0]]
```

Cluster sampling on dataset

In [57]:

```python
import random

# Cluster sampling: divide dataset into clusters based on target labels
clusters = [df[df["species"] == label]
            for label in df["species"].unique()]

# Select one cluster randomly
selected_cluster = random.choice(clusters)
print("Cluster Sample:\n", selected_cluster)
#clusters
#df["species"].unique()
```

```
Cluster Sample:
     sepal_length  sepal_width  petal_length  petal_width    species
100           6.3          3.3           6.0          2.5  virginica
101           5.8          2.7           5.1          1.9  virginica
102           7.1          3.0           5.9          2.1  virginica
103           6.3          2.9           5.6          1.8  virginica
104           6.5          3.0           5.8          2.2  virginica
105           7.6          3.0           6.6          2.1  virginica
106           4.9          2.5           4.5          1.7  virginica
107           7.3          2.9           6.3          1.8  virginica
108           6.7          2.5           5.8          1.8  virginica
109           7.2          3.6           6.1          2.5  virginica
110           6.5          3.2           5.1          2.0  virginica
111           6.4          2.7           5.3          1.9  virginica
112           6.8          3.0           5.5          2.1  virginica
113           5.7          2.5           5.0          2.0  virginica
114           5.8          2.8           5.1          2.4  virginica
115           6.4          3.2           5.3          2.3  virginica
116           6.5          3.0           5.5          1.8  virginica
117           7.7          3.8           6.7          2.2  virginica
118           7.7          2.6           6.9          2.3  virginica
119           6.0          2.2           5.0          1.5  virginica
120           6.9          3.2           5.7          2.3  virginica
121           5.6          2.8           4.9          2.0  virginica
122           7.7          2.8           6.7          2.0  virginica
123           6.3          2.7           4.9          1.8  virginica
124           6.7          3.3           5.7          2.1  virginica
125           7.2          3.2           6.0          1.8  virginica
126           6.2          2.8           4.8          1.8  virginica
127           6.1          3.0           4.9          1.8  virginica
128           6.4          2.8           5.6          2.1  virginica
129           7.2          3.0           5.8          1.6  virginica
130           7.4          2.8           6.1          1.9  virginica
131           7.9          3.8           6.4          2.0  virginica
132           6.4          2.8           5.6          2.2  virginica
133           6.3          2.8           5.1          1.5  virginica
134           6.1          2.6           5.6          1.4  virginica
135           7.7          3.0           6.1          2.3  virginica
136           6.3          3.4           5.6          2.4  virginica
137           6.4          3.1           5.5          1.8  virginica
138           6.0          3.0           4.8          1.8  virginica
139           6.9          3.1           5.4          2.1  virginica
140           6.7          3.1           5.6          2.4  virginica
141           6.9          3.1           5.1          2.3  virginica
142           5.8          2.7           5.1          1.9  virginica
143           6.8          3.2           5.9          2.3  virginica
144           6.7          3.3           5.7          2.5  virginica
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica
```

In [30]:

```
df["species"].unique()
```

Out[30]:

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

In [ ]:

```

```

In [ ]:

```

```