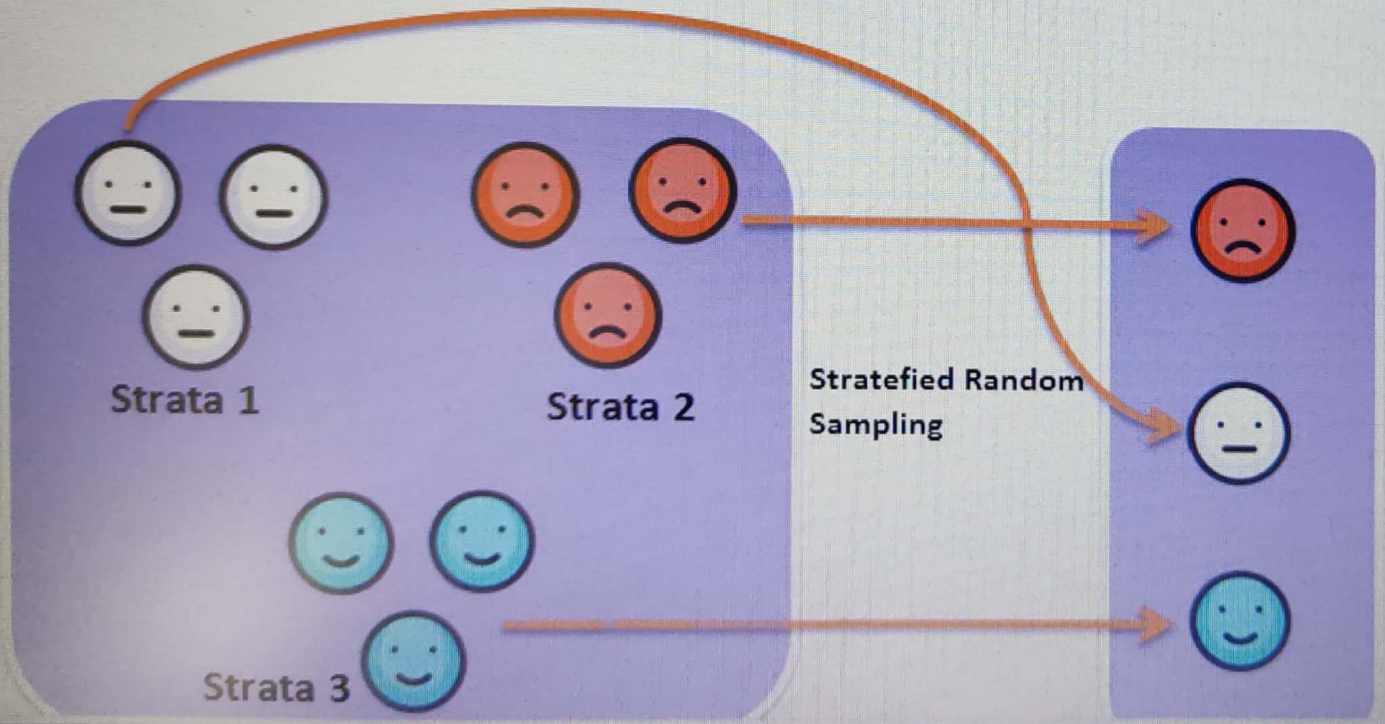
Sampling

- A ***population*** is the group of all items of interest. Ex: All the students on university campus
- A ***sample*** is a set of data drawn from the studied population. Ex: 500 students from the campus
- Few sampling techniques are:
  - ***Random Sampling***: Probability of each record being selected into your sample will be equal. If there are n records, probability of choosing any one record is 1/n.
  - ***Stratified Sampling***: Involves dividing the entire population into homogeneous groups called strata.



[2]:
```
import pandas as pd
```

[3]:
```
data=pd.read_excel('CreditCardData.xlsx')
data.head()
```

[3]:

	Card_ID	Campaign_Responce	Registration_Date	Gender	Birth_Date
0	100005950	False	1998-11-18	M	1984-02-06
1	100022191	True	1999-09-15	F	1959-09-11
2	100025442	False	1998-05-12	M	1970-08-25
3	100026513	False	1999-02-12	M	1951-03-12
4	100039145	False	2000-08-12	M	1949-06-08

[4]:
```
data.shape
```

[4]:
```
(297, 5)
```

[5]:
```
data.dtypes
```

[5]:
```
Card_ID                int64
Campaign_Responce       bool
Registration_Date    datetime64[ns]
Gender                 object
Birth_Date           datetime64[ns]
dtype: object
```

Simple Random Sampling

- Random sampling without replacement: Repeatation of samples are not allowed.
- Random sampling with replacement: Repeatation of samples are allowed.

```
# Task - Select 5 records randomly from Binom sheet and save it as a datafram df1

df1 = data.sample(n = 5, random_state= 44)
df1
```

	Card_ID	Campaign_Responce	Registration_Date	Gender	Birth_Date
252	105607063	False	1998-05-27	F	1940-03-21
175	103828605	False	1998-05-29	M	1966-04-26
192	104235958	False	1998-01-27	F	1951-04-25
43	100867201	True	1999-12-17	F	1957-05-10
28	100614020	True	1998-07-21	M	1958-11-19

```
# Check the indexes of the selected sample
df1.index
```

```
Int64Index([252, 175, 192, 43, 28], dtype='int64')
```

```
4]: # Check the counts of campaign response
    df1.Campaign_Responce.value_counts(normalize= True)
```

```
4]: False    0.6
    True     0.4
    Name: Campaign_Responce, dtype: float64
```

```
5]: data.Campaign_Responce.value_counts(normalize= True)
```

```
5]: False    0.835017
    True     0.164983
    Name: Campaign_Responce, dtype: float64
```

```
6]: df1 = data.sample(n = 100, random_state= 44)
    df1
```

6]:

	Card_ID	Campaign_Responce	Registration_Date	Gender	Birth_Date
252	105607063	False	1998-05-27	F	1940-03-21
175	103828605	False	1998-05-29	M	1966-04-26
192	104235958	False	1998-01-27	F	1951-04-25
43	100867201	True	1999-12-17	F	1957-05-10
28	100614020	True	1998-07-21	M	1958-11-19
...	...	...	...	...	...
157	103475707	True	2000-03-15	F	1984-08-09

[17]:
```
df1.Campaign_Responce.value_counts(normalize= True)
```

[17]:
```
False    0.86
True     0.14
Name: Campaign_Responce, dtype: float64
```

[18]:
```
#Task - Select 10% of records randomly, and create a dataframe df2

df2 = data.sample(frac = 0.1, random_state= 44)
df2.shape
```

[18]:
```
(30, 5)
```

[19]:
```
# Check indexes
df2.index
```

[19]:
```
Int64Index([252, 175, 192,  43,  28,  33,  12, 134, 185, 278,  20, 287, 179,
            148, 201,  46, 294,  95, 110, 107, 174, 211, 226,  36,  65, 135,
            277,   7, 229,  88],
           dtype='int64')
```

[20]:
```
# Check the counts of campaign response
df2.Campaign_Responce.value_counts(normalize= True)
```

[20]:
```
False    0.8
True     0.2
Name: Campaign_Responce, dtype: float64
```

```
#Task - Select 35 records with replacement. Create a dataframe df3
df3 = data.sample(n = 35, random_state= 44, replace= True)
```

```
# Check indexes
df3.index
```

```
Int64Index([276, 241, 173,  59,  96,  84, 239, 120, 151, 195, 199,  67, 227,
            109, 245, 100,  57, 257,  14, 120, 213,  96, 287,  72, 189,  72,
             86, 242, 144, 116,  50,  18,  92, 285,   1],
           dtype='int64')
```

```
sum(df3.index.duplicated())
```

```
3
```

Stratified Sampling

```
# Find out what is the % distribution by Gneder
data['Gender'].value_counts()/data.shape[0]
```

```
M    0.572391
F    0.427609
Name: Gender, dtype: float64
```

```
# Task - Select 30% of records stratified according to Gender

from sklearn.model_selection import train_test_split
```

```
# Create sample - df1 and df2
df1, df2 = train_test_split(data, test_size = 0.3, stratify = data['Gender'], random_state = 44)
```

```
#Train
df1.shape[0]/data.shape[0]
```

```
0.696969696969697
```

```
#Test
df2.shape[0]/data.shape[0]
```

```
0.30303030303030304
```

```
df2.head()
```

	Card_ID	Campaign_Responce	Registration_Date	Gender	Birth_Date
102	102366360	False	2000-05-14	M	1967-12-05
199	104449587	False	2000-01-15	F	1968-11-23
140	103241485	False	1999-05-08	M	1959-06-22
258	105820678	False	2001-09-14	F	1948-03-27
75	101746024	False	1999-06-30	M	1950-10-12

```
df2['Gender'].value_counts(normalize= True)
```

```
M    0.577778
F    0.422222
```

```
df2['Gender'].value_counts(normalize= True)
```

```
M    0.577778
F    0.422222
Name: Gender, dtype: float64
```

```
df1['Gender'].value_counts(normalize= True)
```

```
M    0.570048
F    0.429952
Name: Gender, dtype: float64
```

```
# Task - Select 25% of records from 'data' stratified based on 'Campaign_Responce' variable.
data['Campaign_Responce'].value_counts(normalize= True)
```

```
False    0.835017
True     0.164983
Name: Campaign_Responce, dtype: float64
```