In [1]:
```
import numpy as np
import pandas as pd
```

In [11]:
```
# Define total number of groups
number_of_groups = 15
# Create data dictionary
data = {'Id': np.arange(1, number_of_groups+1).tolist(),
        'Group Number': np.arange(1,number_of_groups+1).tolist(),
        'Value': [19, 71, 58, 62, 12, 91, 60, 75, 38, 51,23,97,8,34,65]
        }
# Transform dictionary into a data frame
df1 = pd.DataFrame(data)

display(df1)
```

| | Id | Group Number | Value |
|---|---|---|---|
| **0** | 1 | 1 | 19 |
| **1** | 2 | 2 | 71 |
| **2** | 3 | 3 | 58 |
| **3** | 4 | 4 | 62 |
| **4** | 5 | 5 | 12 |
| **5** | 6 | 6 | 91 |
| **6** | 7 | 7 | 60 |
| **7** | 8 | 8 | 75 |
| **8** | 9 | 9 | 38 |
| **9** | 10 | 10 | 51 |
| **10** | 11 | 11 | 23 |
| **11** | 12 | 12 | 97 |
| **12** | 13 | 13 | 8 |
| **13** | 14 | 14 | 34 |
| **14** | 15 | 15 | 65 |

In [12]:
```
# 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
```

In [13]:
```
# Obtain a systematic sample and save it in a new variable
systematic_sample = systematic_sampling(df1, 4)
# View sampled data frame
display(systematic_sample)
```

| | Id | Group Number | Value |
|---|---|---|---|
| **0** | 1 | 1 | 19 |
| **4** | 5 | 5 | 12 |
| **8** | 9 | 9 | 38 |
| **12** | 13 | 13 | 8 |

In [16]:
```
# Create a dictionary of students
students = {
    'Name': ['Jayesh','Abhishek','Zaid','Anish','Aditya','Lisa', 'Kate', 'B
             'Alex', 'Evan', 'Greg', 'Sam', 'Ella'],
    'ID': ['018', '016', '017', '010', '058','020','030','040','050', '060'
           '007', '008', '009', '070','080'],
    'Grade': ['A', 'A', 'C','A', 'C','B', 'B', 'B', 'B', 'C',
              'A', 'A','B','C','B'],

    'Category': [2, 3, 1, 3, 2, 3, 3, 1, 2, 1,3,2,1,3,1]
}

# Create dataframe from students dictionary
df = pd.DataFrame(students)

# View the dataframe
df
```

Out[16]:

| | Name | ID | Grade | Category |
|---|---|---|---|---|
| **0** | Jayesh | 018 | A | 2 |
| **1** | Abhishek | 016 | A | 3 |
| **2** | Zaid | 017 | C | 1 |
| **3** | Anish | 010 | A | 3 |
| **4** | Aditya | 058 | C | 2 |
| **5** | Lisa | 020 | B | 3 |
| **6** | Kate | 030 | B | 3 |
| **7** | Ben | 040 | B | 1 |
| **8** | Kim | 050 | B | 2 |
| **9** | Josh | 060 | C | 1 |
| **10** | Alex | 007 | A | 3 |
| **11** | Evan | 008 | A | 2 |
| **12** | Greg | 009 | B | 1 |
| **13** | Sam | 070 | C | 3 |
| **14** | Ella | 080 | B | 1 |

In [17]:
```
df.groupby('Category', group_keys=False).apply(lambda x: x.sample(2))
```

Out[17]:

| | Name | ID | Grade | Category |
|---|---|---|---|---|
| **9** | Josh | 060 | C | 1 |
| **12** | Greg | 009 | B | 1 |
| **4** | Aditya | 058 | C | 2 |
| **8** | Kim | 050 | B | 2 |
| **6** | Kate | 030 | B | 3 |
| **10** | Alex | 007 | A | 3 |

In [18]:
```
df.groupby('Grade', group_keys=False).apply(lambda x: x.sample(frac=0.6))
```

Out[18]:

| | Name | ID | Grade | Category |
|---|---|---|---|---|
| **10** | Alex | 007 | A | 3 |
| **0** | Jayesh | 018 | A | 2 |
| **3** | Anish | 010 | A | 3 |
| **12** | Greg | 009 | B | 1 |
| **8** | Kim | 050 | B | 2 |
| **6** | Kate | 030 | B | 3 |
| **7** | Ben | 040 | B | 1 |
| **4** | Aditya | 058 | C | 2 |
| **9** | Josh | 060 | C | 1 |

In [19]:
```
#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)})
df.head(21)
```

Out[19]:

| | tour | experience |
|---|---|---|
| **0** | 1 | 8.764052 |
| **1** | 1 | 7.400157 |
| **2** | 1 | 7.978738 |
| **3** | 1 | 9.240893 |
| **4** | 1 | 8.867558 |
| **5** | 1 | 6.022722 |
| **6** | 1 | 7.950088 |
| **7** | 1 | 6.848643 |
| **8** | 1 | 6.896781 |
| **9** | 1 | 7.410599 |
| **10** | 1 | 7.144044 |
| **11** | 1 | 8.454274 |
| **12** | 1 | 7.761038 |
| **13** | 1 | 7.121675 |
| **14** | 1 | 7.443863 |
| **15** | 1 | 7.333674 |
| **16** | 1 | 8.494079 |
| **17** | 1 | 6.794842 |
| **18** | 1 | 7.313068 |
| **19** | 1 | 6.145904 |
| **20** | 2 | 4.447010 |

In [20]:
```
#Randomly choose 4 tour groups out of the 10
clusters = np.random.choice(np.arange(1,16), size=3, replace=False)
print(clusters)
#Define sample as all members who belong to one of the 4 tour groups
cluster_sample = df[df['tour'].isin(clusters)]

#View first six rows of sample
#cluster_sample.head(60)
```

```
[12  3  5]
```

In [21]:
```
#Find how many observations came from each tour group
cluster_sample['tour'].value_counts()
```

Out[21]:
```
3    20
5    20
Name: tour, dtype: int64
```