In [1]:
```
import os
os.environ['OMP_NUM_THREADS'] = '1'

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import scale
```

In [2]:
```
csv_in = 'car-data.csv'
df = pd.read_csv(csv_in, sep=',', skiprows=0, header=0)
print(df.shape)
print(df.info())
display(df.head())
display(df.tail())
```

```
(36, 5)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Car     36 non-null     object
 1   Model   36 non-null     object
 2   Volume  36 non-null     int64 
 3   Weight  36 non-null     int64 
 4   CO2     36 non-null     int64 
dtypes: int64(3), object(2)
memory usage: 1.5+ KB
None
```

	Car	Model	Volume	Weight	CO2
0	Toyota	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95
3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105

	Car	Model	Volume	Weight	CO2
31	Volvo	XC70	2000	1746	117
32	Ford	B-Max	1600	1235	104
33	BMW	216	1600	1390	108
34	Opel	Zafira	1600	1405	109
35	Mercedes	SLK	2500	1395	120

In [3]:
```
dfX = df[['Weight', 'CO2']] #1
print(dfX.shape)
display(dfX.head())
```

```
(36, 2)
```

	Weight	CO2
0	790	99
1	1160	95
2	929	95
3	865	90
4	1140	105

In [4]:
```
X_scaled = scale(dfX) #2
```

In [5]:
```
print(type(X_scaled))
print(X_scaled.shape)
```

```
<class 'numpy.ndarray'>
(36, 2)
```

In [6]:
```
emit_df = pd.DataFrame(X_scaled, columns=dfX.columns) #3
print(type(emit_df))
display(emit_df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
```

	Weight	CO2
0	-2.103893	-0.411925
1	-0.554072	-0.956120
2	-1.521663	-0.956120
3	-1.789740	-1.636364
4	-0.637846	0.404367

In [7]:
```
km = KMeans(n_clusters=4, n_init=6, random_state=12) #4~6
clstr = km.fit_predict(emit_df)
print(clstr)
```

```
[1 1 1 1 3 1 1 2 1 1 1 1 1 2 2 2 2 3 3 3 2 2 2 2 2 2 3 0 3 0 0 0 3 3 3 0]
```

In [8]:
```
print(km.inertia_) #7
```

```
14.956613161341018
```

In [9]:
```
emit_df['cluster_no'] = clstr #8
display(emit_df.head())
```

	Weight	CO2	cluster_no
0	-2.103893	-0.411925	1
1	-0.554072	-0.956120	1
2	-1.521663	-0.956120	1
3	-1.789740	-1.636364	1
4	-0.637846	0.404367	3

In [10]:
```
print(emit_df['cluster_no'].value_counts())
```

```
cluster_no
1    11
2    11
3     9
0     5
Name: count, dtype: int64
```

Ans(9)

5