In [1]:

```
import pandas as pd
df = pd.read_csv('C:/Users/user/Desktop/프로젝트_종합/프로젝트_서울/결합 데이터_2(자치구)/결합_구단
df.columns = ['gid','gu','AVL','Ani_n', 'Ani_rel','traffic']
```

In [2]:

```
ss = df[['Ani_n','AVL', 'Ani_rel','traffic']]
ss.head()
```

Out[2]:

```
   Ani_n         AVL  Ani_rel  traffic
0  16944  2726405.03       42   154307
1  28578  1694321.33       70   127423
2  27770  1406054.97       71   203769
3  14676  1283371.75       40   113104
4  16179  1201495.24       43   118207
```

In [3]:

```
# 변수 정규화
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

#표준화
sc = StandardScaler()
ss_scaled = sc.fit_transform(ss)
ssdf=pd.DataFrame(ss_scaled)
```

In [4]:

```
ssdf
```

Out[4]:

	0	1	2	3
0	-0.254497	-0.334201	-0.714913	-0.885311
1	1.392487	-0.823716	0.112259	-1.286746
2	1.278102	-0.960440	0.141801	-0.146739
3	-0.575570	-1.018629	-0.773997	-1.500559
4	-0.362795	-1.057463	-0.685371	-1.424360
5	0.443851	-0.575459	-0.242243	-0.046903
6	0.124618	-0.724912	-0.064992	-0.603318
7	0.116549	-0.069569	-0.271785	-0.802662
8	-2.274509	-0.185309	-0.271785	0.331686
9	-0.478172	-0.008193	-0.153618	0.528491
10	-0.145066	0.678068	-0.183160	0.638331
11	-1.531710	3.145685	-1.158041	-0.478814
12	-1.272501	0.007392	-1.335293	-0.551563
13	-0.065223	-0.077011	0.348594	0.198805
14	1.662879	1.059301	2.150648	2.053134
15	1.787883	2.566228	3.243698	1.994480
16	0.187191	0.544408	0.909890	2.050804
17	0.889785	-0.769530	-0.183160	0.514544
18	-0.226608	0.007035	-0.803539	-0.277455
19	0.011790	0.199739	-0.389953	0.351127
20	-0.862383	-0.558757	-0.921706	-1.327018
21	-1.390426	-0.472548	-0.685371	-0.756298
22	1.318448	-0.806028	1.146225	0.512319
23	0.122212	0.003296	0.644013	0.478916
24	0.103666	0.230612	0.141801	0.435106

In [5]:

```
ss_ar=ssdf.to_numpy()
#cc_scaled 데이터를 넣어 클러스터링
kmeans = KMeans(n_clusters=3, random_state=0)
clusters = kmeans.fit(ss_ar)

#클러스터링 변수인 clusters 값을 원본 데이터인 'cc'내에 넣기
ss['cluster'] = clusters.labels_
ss
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1332: UserWarni
ng: KMeans is known to have a memory leak on Windows with MKL, when there are less c
hunks than available threads. You can avoid it by setting the environment variable O
MP_NUM_THREADS=1.
  warnings.warn(
C:\Users\user\AppData\Local\Temp\ipykernel_4444\1254553237.py:7: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  ss['cluster'] = clusters.labels_
```

Out[5]:

	Ani_n	AVL	Ani_rel	traffic	cluster
0	16944	2726405.03	42	154307	0
1	28578	1694321.33	70	127423	2
2	27770	1406054.97	71	203769	2
3	14676	1283371.75	40	113104	0
4	16179	1201495.24	43	118207	0
5	21877	2217741.92	58	210455	2
6	19622	1902638.46	64	173192	2
7	19565	3284348.14	57	159842	0
8	2675	3040324.75	57	235809	0
9	15364	3413752.74	61	248989	2
10	17717	4860651.19	60	256345	2
11	7922	10063323.10	27	181530	0
12	9753	3446612.38	21	176658	0
13	18281	3268657.86	78	226910	2
14	30488	5664434.35	139	351094	1
15	31371	8841606.91	176	347166	1
16	20064	4578844.96	97	350938	2
17	25027	1808566.97	60	248055	2
18	17141	3445859.03	39	195015	0
19	18825	3852152.55	53	237111	2

	Ani_n	AVL	Ani_rel	traffic	cluster
20	12650	2252954.65	35	124726	0
21	8920	2434716.58	43	162947	0
22	28055	1731614.89	105	247906	2
23	19605	3437975.88	88	245669	2
24	19474	3917243.72	71	242735	2

In [6]:

```
ss.groupby('cluster').count()
```

Out[6]:

	Ani_n	AVL	Ani_rel	traffic
cluster				
0	10	10	10	10
1	2	2	2	2
2	13	13	13	13

In [7]:

```
ss.groupby('cluster').mean()
```

Out[7]:

	Ani_n	AVL	Ani_rel	traffic
cluster				
0	12642.500000	3.317941e+06	40.4	162214.5
1	30929.500000	7.253021e+06	157.5	349130.0
2	21558.384615	2.930017e+06	72.0	232269.0

In [8]:

```
df['cluster'] = clusters.labels_
df
```

Out[8]:

	gid	gu	AVL	Ani_n	Ani_rel	traffic	cluster
0	11410	서대문구	2726405.03	16944	42	154307	0
1	11380	은평구	1694321.33	28578	70	127423	2
2	11350	노원구	1406054.97	27770	71	203769	2
3	11320	도봉구	1283371.75	14676	40	113104	0
4	11305	강북구	1201495.24	16179	43	118207	0
5	11290	성북구	2217741.92	21877	58	210455	2
6	11260	중랑구	1902638.46	19622	64	173192	2
7	11230	동대문구	3284348.14	19565	57	159842	0
8	11215	광진구	3040324.75	2675	57	235809	0
9	11200	성동구	3413752.74	15364	61	248989	2
10	11170	용산구	4860651.19	17717	60	256345	2
11	11140	중구	10063323.10	7922	27	181530	0
12	11110	종로구	3446612.38	9753	21	176658	0
13	11740	강동구	3268657.86	18281	78	226910	2
14	11710	송파구	5664434.35	30488	139	351094	1
15	11680	강남구	8841606.91	31371	176	347166	1
16	11650	서초구	4578844.96	20064	97	350938	2
17	11620	관악구	1808566.97	25027	60	248055	2
18	11590	동작구	3445859.03	17141	39	195015	0
19	11560	영등포구	3852152.55	18825	53	237111	2
20	11545	금천구	2252954.65	12650	35	124726	0
21	11530	구로구	2434716.58	8920	43	162947	0
22	11500	강서구	1731614.89	28055	105	247906	2
23	11470	양천구	3437975.88	19605	88	245669	2
24	11440	마포구	3917243.72	19474	71	242735	2