03. Clustering

1. Introduction

2주차와 마찬가지로 Clustering에 Kaggle의 Divorce Prediction 데이터를 활용했다. 2주차 PCA와 FA의 결과 54개의 Question(Feature)에 대해서, FA(Factor Analysis) – roation : Varimax를 통해 진행한 결과

Factor #	각 요인의 Loading 절대값이 0.6이상인 변수
Factor 1	Q1, Q2, Q3, Q4, Q5, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q26, Q27, Q28, Q29, Q30
Factor 2	Q31, Q32, Q33, Q34, Q35, Q36, Q37, Q38, Q51, Q53, Q54
Factor 3	Q43, Q45, Q46, Q47
Factor 4	Q6, Q7 *(Factor4의 경우 절대값 제일 높은 두가지를 선택함)*

다음 과 같이 Factor가 4개로 나왔다. Factor 4개를 통해 줄인 결과값을 바탕으로 이번에 군집분석을 진행하였다. 각 Factor 별로 'Affection', 'Aggression', 'Silence', 'Home-Distance' Naming을 해준 뒤, 아래 과정을 통해 Outlier 값을 제거하고 난 후의 데이터를 통해 Cluster를 진행하였다.

```
# 이상치 제거 함수 (1Q - 1.5*IQR, 3Q + 1.5*IQR 를 넘어가는 값을 가진 행을 제거)
import numpy as np

def get_outlier(df=None, column=None, weight=1.5):
  quantile_25 = np.percentile(df[column].values, 25)
  quantile_75 = np.percentile(df[column].values, 75)

  IQR = quantile_75 - quantile_25
  IQR_weight = IQR*weight

  lowest = quantile_25 - IQR_weight
  highest = quantile_75 + IQR_weight

  outlier_idx = df[column][ (df[column] < lowest) | (df[column] > highest) ].index
  return outlier_idx
```

```
# 함수 사용해서 이상치 값 삭제
outlier_idx = get_outlier(df=data, column='Affection', weight=1.5)
data.drop(outlier_idx, axis=0, inplace=True)
```

```
# 함수 사용해서 이상치 값 삭제
outlier_idx = get_outlier(df=data, column='Aggression', weight=1.5)
data.drop(outlier_idx, axis=0, inplace=True)
```

```
# 함수 사용해서 이상치 값 삭제
outlier_idx = get_outlier(df=data, column='Silence', weight=1.5)
data.drop(outlier_idx, axis=0, inplace=True)
```

```
# 함수 사용해서 이상치 값 삭제
outlier_idx = get_outlier(df=data, column='Home-Distance', weight=1.5)
data.drop(outlier_idx, axis=0, inplace=True)
```

```
data
```

Divorce	Affection	Aggression	Silence	Home-Distance
1	-1.124784	0.747916	-1.086777	1.183849
1	1.540539	-0.780734	-1.039879	2.025911
1	0.141439	-1.120123	0.185957	-0.707625
1	-0.474497	-0.702579	-0.605910	1.317738
1	0.876337	-0.852976	0.464757	0.954184
...	...	...	...	...
0	-0.460458	0.133171	-1.566689	0.022206
0	-1.670673	0.229122	0.425698	0.203356
0	-0.308820	-0.779170	-0.124893	-0.317710
0	-0.991156	-0.383782	0.417958	-0.420596
0	-0.762844	-0.240818	0.083437	-1.142981

159 rows × 4 columns

<Outlier를 제거한 데이터>

2. 군집분석(Cluster Analysis)

I. 군집분석(Cluster Analysis)의 의미

군집분석은 동일한 성격을 가진 여러 개의 그룹으로 대상을 분류하는 것이다. 이때 나눠지는 그룹들을 Clustering 이라고 부른다. 각 집단의 성격을 파악함으로써 데이터 전체의 구조에 대한 이해를 돕고자하는 탐색적 분석방법이다.

군집분석은 분류(Classification)이랑은 다르다. 군집 분석은 종속변수에 대한 독립변수의 영향과 같이 사전에 정의된 특수한 목적이 없으며, 데이터 자체를 통해 데이터 구조와 자료를 탐색하고 요약하는 기법이다. 전체를 유사한 군집으로 구분한다면, 전체에 대한 의미 있는 정보를 얻어낼 수 있다. 동일 군집 내 개체들은 유사한 성격을 갖고, 다른 군집은 다른 성격을 갖도록 군집이 형성이 된다. 군집분석은 군집 내 차이를 줄이고, 군집 간 차이를 최대화 화도록 하여 대표성을 찾는 원리로 구현된다.

군집분석은 계층적 군집분석(Hierarchical Clustering)과 비계층적 군집분석(Non-Hierarchical Clustering)으로 나뉜다. 본 분석은 계층적 군집분석과 비계층벅 군집분석 중 K-Means Clustering

으로 분석을 실시하였다.

II. Hierarchical Clustering Method

계층적 군집분석은 관측치, 혹은 군집 간의 유사도를 통해 군집을 만든 후, 그 군집과 가까운 관측치를 찾아 하나씩 덴드로그램을 타고 올라가는 군집 분석 방법이다. 덴드로그램에서 먼저 합쳐진 부분이 아래쪽에 있을수록 유사도가 높으며, 위로 올라갈수록 유사도가 낮아진다. 일반적으로 데이터 사이의 거리는 유클리디안 방식을 활용해 구하며, 관측치와 군집 간의 거리를 구하는 방법에 따라 single linkage, complete linkage, average linkage, ward, centroid로 나뉜다. 해당 과제에서는 ward와 complete linkage 방법을 통해 군집 분석을 실행하고, 그 결과를 비교해보기로 결정했다.

i. Ward Method

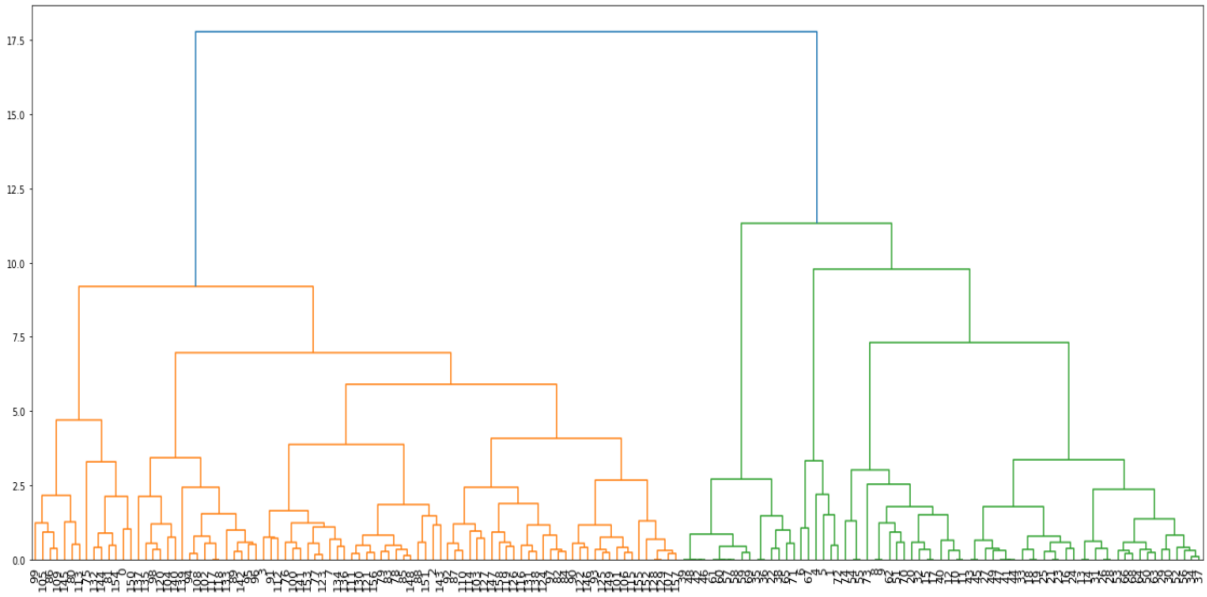
Ward method는 두 cluster간의 유사성을 두 cluster가 합쳐졌을 때의 ESS(Minimum Error sum of Square, 오차 제곱합)의 증가분에 기반하여 측정한다. 즉, distance matrix를 구할 때, ESS의 증분을 두 군집사이의 거리로 측정하게 되고, 이것을 최소화하는 방향으로 clustering이 이루어진다. 이때 ESS는 각 cluster별 ESS를 다 더해준 것의 총합이다. 세부적으로 i는 cluster의 index를 나타내고, j는 i-cluster의 관측치의 index를, p는 개체들의 특징을 나타내는 변수들의 개수를 의미한다. 우리는 4개의 Factor를 사용할 것이므로 p=4가 될 것이다. 이렇게 모든 cluster, 관측치, 변수의 관점에서 ESS를 다 더한 것이 총 ESS가 된다.

$$ESS = \sum_{i=1}^{g} ESS_i$$

$$ESS_i = \sum_{j=1}^{N_i} \sum_{k=1}^{p} (x_{ijk} - \bar{x}_{ik})^2$$

i=1, …, g : Cluster
j=1, …, N_i : j^{th} observation of i^{th} cluster
k=1, …, p : variable
$\bar{x}_{ik}$: mean of variable k within i^{th} cluster

A. Ward Clustering 결과 & Dendrogram



위 그림은 Cut-Off를 하기전의 clustering결과를 dendrogram으로 나타낸 것이다.

B. Cut-off & clustering

■ 실루엣 실루엣 기법은 군집의 성능을 평가하는 지표를 계산하는 방법이다. 앞선 Elbow기법을 통해 구해진 k값들 중 어떤 K가 가장 성능이 좋은 군집 개수인지 평가 지표를 통해 판단하고자 한다.

$$s^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max\{a^{(i)}, b^{(i)}\}}$$

위 식이 실루엣 계수 s(i)를 계산하는 식이다. 위 식의 a는 군집 내 데이터 응집도 (cohesion)을 나타낸 값으로, 데이터와 동일한 군집 내 다른 데이터들과의 평균 거리이다. 또한, b는 군집 간 분리도(separation)을 나타내는 값으로, 데이터와 가장 가까운 군집 내의 모든 데이터들과의 평균 거리를 의미한다. 이때 만약 군집 개수가 최적화되어 있을수록 분리도 B는 커지고 응집도 a는 작아진다. 반대로 응집도와 분리도가 같으면 실루엣 계수는 0이 된다. 따라서 실루엣 계수는 0과 1 사이의 값을 갖으며, 최적화 되어있을수록 1에 가까운 값을 갖는다.

■ **Cut-Off Value = 9.5 일 때,**

```
from scipy.cluster.hierarchy import fcluster # 지정한 클러스터 자르기
cut_tree = fcluster(clusters_ward, t=9.5, criterion='distance') # 본 과제에서는 9.5(y축)에서 cut
cut_tree # prediction
```

```
array([1, 3, 1, 1, 3, 3, 3, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 4, 2, 2, 4, 4, 2, 4,
       4, 4, 2, 4, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 4, 4, 4, 2,
       4, 3, 4, 2, 4, 2, 3, 4, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1], dtype=int32)
```

```
data_1['hc_cluster'].value_counts()
```

```
1    88
4    49
2    16
3     6
Name: hc_cluster, dtype: int64
```

```
from sklearn.metrics import silhouette_samples, silhouette_score
score_samples = silhouette_samples(data_1.iloc[:,[0,1,2,3]], data_1['hc_cluster'])
data_1['silhouette_coeff'] = score_samples
average_score = silhouette_score(data_1.iloc[:,[0,1,2,3]], data_1['hc_cluster'])
print('데이터셋 Silhouette Analysis Score:{0:.3f}'.format(average_score))
```

```
데이터셋 Silhouette Analysis Score:0.384
```

```
# 군집별 평균 silhouette_score 값
data_1.groupby('hc_cluster')['silhouette_coeff'].mean()
```

```
hc_cluster
1    0.341731
2    0.655200
3    0.410665
4    0.367066
Name: silhouette_coeff, dtype: float64
```

위 코드의 결과는 Cut-Off value를 9.5로 진행해 4개의 Cluster로 진행한 결과다. 실루엣 점수는 0.384가 나왔다.

■ **Cut-Off Value = 8 일 때,**

```
from scipy.cluster.hierarchy import fcluster # 지정한 클러스터 자르기
cut_tree = fcluster(clusters_ward, t=8, criterion='distance') # 본 과제에서는 8(y축)에서 cut
cut_tree # prediction
```

```
array([1, 4, 2, 2, 4, 4, 4, 2, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 3, 3, 5, 3, 3, 5, 5, 3, 5,
       5, 5, 3, 5, 3, 5, 5, 5, 5, 5, 5, 5, 5, 3, 3, 3, 3, 3, 5, 5, 5, 3,
       5, 4, 5, 3, 5, 3, 4, 5, 5, 1, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1,
       2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2,
       1, 2, 2, 2, 2], dtype=int32)
```

```
data_1['hc_cluster'].value_counts()
```

```
2     74
5     49
3     16
1     14
4      6
Name: hc_cluster, dtype: int64
```

```
from sklearn.metrics import silhouette_samples, silhouette_score
score_samples = silhouette_samples(data_1.iloc[:,[0,1,2,3]], data_1['hc_cluster'])
data_1['silhouette_coeff'] = score_samples
average_score = silhouette_score(data_1.iloc[:,[0,1,2,3]], data_1['hc_cluster'])
print('데이터셋 Silhouette Analysis Score:{0:.3f}'.format(average_score))
```

```
데이터셋 Silhouette Analysis Score:0.373
```

```
# 군집별 평균 silhouette_score 값
data_1.groupby('hc_cluster')['silhouette_coeff'].mean()
```

```
hc_cluster
1    0.244798
2    0.342382
3    0.655200
4    0.397388
5    0.362155
Name: silhouette_coeff, dtype: float64
```

위 코드는 cut-off value 8로 진행해 5개의 cluster로 진행한 것을 나타낸다. 실루엣 점수가 0.373이 나왔다.

- cut-off value가 9.5인 경우의 전체 Silhouette Score 평균값이 더 높게 나오고, 각 cluster의 Silhouette Score와 전체 Silhouette Score 평균을 비교 했을 때, cut-off value가 9.5인 경우에 4개의 cluster중 2개의 cluster가 더 높게나옴을 확인할 수있었고 따라서 cut-off value를 9.5로 결정하기로 하였다.

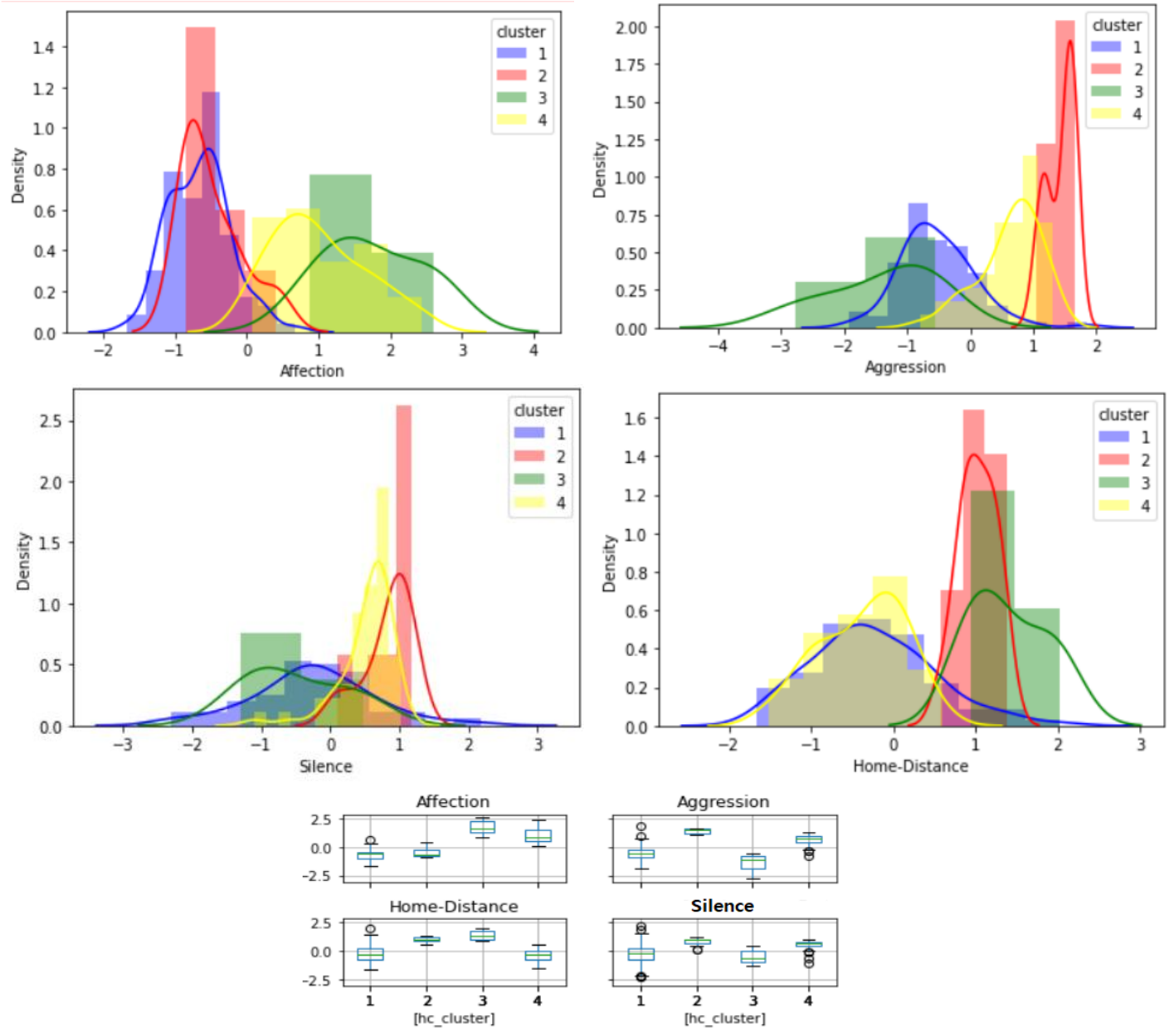
```
# 군집별 통계(평균)
cluster_g = data_1.groupby('hc_cluster')
cluster_g.mean()
```

hc_cluster	Affection	Aggression	Silence	Home-Distance
1	-0.655044	-0.522949	-0.262649	-0.251451
2	-0.484207	1.431613	0.839318	1.024817
3	1.755908	-1.394384	-0.503938	1.398282
4	1.012363	0.637316	0.532882	-0.385392

```
# 군집별 통계(표준편차)
cluster_g.std()
```

hc_cluster	Affection	Aggression	Silence	Home-Distance
1	0.428925	0.596120	0.881843	0.737378
2	0.422110	0.217806	0.350341	0.227104
3	0.695215	0.864678	0.707468	0.468785
4	0.650611	0.494799	0.393610	0.528834

➢ 클러스터별 특징을 분석하기위해 각 클러스터로 그룹화해서 Factor별 평균과 분산을 구해보았고, 두가지 Plotting방법으로(ex.Boxplot) 시각화하였다.



<통계치와 시각화 과정을 통해 특성을 분석한 표>

	Affection	Aggression	Silence	Home-Distance
Cluster1	**조금 낮음**	**조금 낮음**	-	-
Cluster2	**조금 낮음**	**매우 높음**	**높음**	**높음**
Cluster3	**매우 높음**	**매우 낮음**	-	**매우 높음**
Cluster4	**높음**	-	-	-

- 각 cluster의 평균, 표준편차, plot을 바탕으로 했을 때 다음처럼 분석할 수 있다.

Cluster 1

해당 군집은 5가지 군집 중 Affection이 가장 낮은 평균치를 기록했고, 표준편차를 고려했을 때, 낮은 Affection을 이 군집의 주요한 특징 중 하나라고 판단.

Aggression의 경우 2번째로 작았지만, 표준편차 값이 크기 때문에 제외했다. Silence는 값이 0에 매우 가까웠고 표준편차 역시 0.8로 가장 컸다.

따라서 해당 군집을 나타내는 특징에서 Silence 역시 제외했다. 같은 이유로 Home-Distance도 특징에서 제외했다.

그러므로 Cluster를 가장 잘 나타내는 특징으로 낮은 Affection을 선정했다.

- 배우자에게 관심이나 긍정적인 태도를 가지는정도(Affection)는 조금 낮지만, 논쟁 시 부정적인 태도를 보이는정도(Aggression)도 조금 낮음을 보인다.

Cluster 2

해당 군집은 Aggression 부분에서 다른 군집들보다 훨씬 높은 평균치를 기록했고, 표준편차 값 역시 약 0.2로 상당히 낮은 수치를 기록했다.

그러므로 높은 Aggression을 이 군집을 나타내는 확실한 특징으로 선정했다.

Silence의 경우에도 4가지 군집 중 가장 큰 평균치를 기록했다. 표준편차 값 역시 작았기 때문에 높은 silence를 해당 군집을 나타내는 특징으로 선정했다.

Home-Distance는 4가지 군집 중 2번째로 높은 수치를 기록했다. 표준편차의 값이 0.2로 매우 작았기 때문에 높은 Home-Distance도 해당 군집을 나타내는 특징으로 선정했다. 반면, Affection의 경우 평균값이 0과 가까웠고 분산값을 고려했을 때 해당 군집을 나타내기엔 적합하지 않다고 판단해 제외했다.

그러므로 Cluster를 가장 잘 나타내는 특징으로 높은 Aggression, 높은 Silence, 높은

Home-Distance를 선정했다.

- 논쟁 시 부정적인 태도를 보이는정도(Aggression)도 매우 높고, 논쟁 시 서로 침묵을 하는 정도(Silence)와 집안에서 서로 거리를 두는정도(Home-Distance)가 높음을 보인다.

Cluster3

해당 군집은 4가지 군집 중 Affection이 가장 높은 평균치를 기록했고, 표준편차는 크지않음을 보아 Affection이 해당 군집을 나타내는 주요한 특징 중 하나라고 판단할 수 있었다. Aggression또한 매우 낮은 평균치를 보이지만 표준편차가 높아, -영역에 고루 퍼져있고, 이또한 군집을 나타내는 특징으로 판단할 수 있었다.Silence는 다른 3 factor에 비해 낮은 절댓값을 보이고, 표준편차 또한 크다고 판단했기에 이 군집을 나타내는 특성이라고 생각하지않았다. 또 Home-Distance도 매우 높고, 표준편차가 작아 이 군집을 나타내는 특성이라고 선정할수있었다.

- 배우자에게 관심이나 긍정적인 태도를 가지는정도(Affection)가 매우 높고, 논쟁 시 부정적인 태도를 보이는정도(Aggression)가 매우 낮고, 집안에서 서로 거리를 두는정도(Home-Distance)가 매우 높음을 보인다.

Cluster4

해당 군집은 4개의 군집 중 Affection에서 두번째로 높은 수치를 가지며, 표준편차가 크지 않음을 보아 이 군집을 나타내는 특성이라고 판단할수있다. 또한 Aggression, Silence, Home-Distance의 평균은 Affection에 비해 50%정도의 값을 지니고, 표준편차 또한 작지않은 값을 지니기에 이 군집의 특성을 명확히 드러내지않는다고 판단했다.

- 배우자에게 관심이나 긍정적인 태도를 가지는정도(Affection)가 높음을 보인다.

C. Ward Cluster 결과

```
from sklearn.metrics import silhouette_samples, silhouette_score
score_samples = silhouette_samples(data_1.iloc[:,[0,1,2,3]], data_1['hc_cluster'])
data_1['silhouette_coeff'] = score_samples
average_score = silhouette_score(data_1.iloc[:,[0,1,2,3]], data_1['hc_cluster'])
print('데이터셋 Silhouette Analysis Score:{0:.3f}'.format(average_score))
```

```
데이터셋 Silhouette Analysis Score:0.384
```

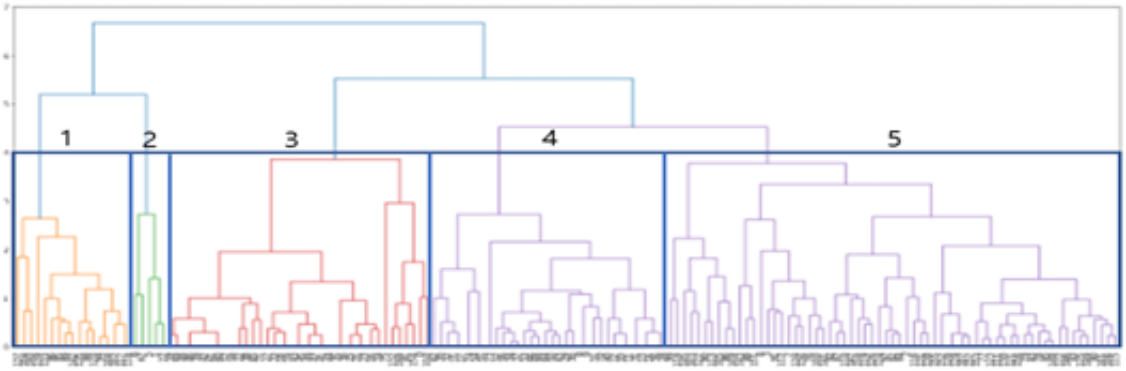
```
# 군집별 평균 silhouette_score 값
data_1.groupby('hc_cluster')['silhouette_coeff'].mean()
```

```
hc_cluster
1    0.341731
2    0.655200
3    0.410665
4    0.367066
Name: silhouette_coeff, dtype: float64
```

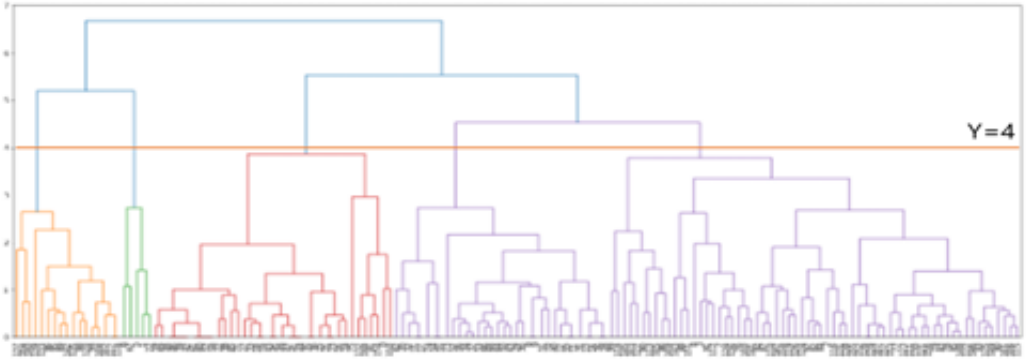
그리고 각 cluster별 Silhouette score의 평균을 위의 코드로 구하여 볼 수 있다.
따라서 전체의 Silhouette score와 cluster별 Silhouette score을 비교해보았을 때, 2, 3번 cluster는 전체 Silhouette score보다 높기에 clustering이 더 잘 되었다고 말할 수 있고, 1, 4번 cluster는 전체 Silhouette score보다 낮기에 조금 더 아쉽게 clustering되었다고 말할 수 있다.

ii. Complete Linkage Method

Complete method는 군집 간 거리를 잴 때 다른 군집의 점들 중에서 가장 멀리 떨어진 두 점 간의 거리를 측정하는 방식을 의미한다. 다음은 FA 를 통해 만든 새로운 변수들로 나눈 데이터를 Complete method를 통해 군집화를 진행한 후 나타낸 덴드로그램이다.



해당 덴드로그램을 해석의 편의성과 군집의 적절한 분류를 위해 Y=4 값을 기준으로 잘라 5개의 군집을 선정하였다.



A. 군집 별 분석

hc_cluster	Affection	Aggression	Silence	Home-Distance
1	-0.924776	-1.107302	0.903788	-0.139204
2	1.931822	-1.502666	-0.697677	1.487101
3	-0.178212	1.087447	0.398268	0.632163
4	1.296275	0.433311	0.439592	-0.609064
5	-0.561372	-0.461071	-0.439273	-0.378588

hc_cluster	Affection	Aggression	Silence	Home-Distance
1	0.310627	0.332065	0.585104	0.616132
2	0.609969	0.920135	0.586624	0.464254
3	0.563933	0.557447	0.896280	0.557389
4	0.563203	0.448568	0.431900	0.460699
5	0.473612	0.483852	0.661706	0.685117

<왼쪽은 군집 별 평균, 오른쪽은 군집 별 표준편차를 나타낸 DataFrame이다.>

■ Cluster 1

해당 군집은 5가지 군집 중 Affection에서 가장 낮은 평균치를 기록했고, 표준편차 또한 작은 것으로 보아 낮은 Affection이 해당 군집을 나타내는 주요한 특징 중 하나라고 판단할 수 있었다. Aggression에서는 두번째로 낮은 수치를 기록했고 표준편차는 가장 작았다. 낮은 Aggression 역시 이 군집을 나타내는 특징으로 판단할 수 있었다. 한편 silence에서는 5개의 군집 중 가장 높은 수치를 기록했다. 표준편차 역시 크지 않아 높은 silence 또한 해당 군집을 나타내는 주요한 특징으로 선정할 수 있었다. 반면 Home-Distance에서는 평균은 0에 가까웠고 분산 또한 꽤 컸다. 그러므로 Home-Distance는 해당 군집을 나타내는 주요한 특성에서 제외할 수 있었다.

- Cluster 1은 **낮은 Affection, 낮은 Aggression, 높은 Silence**가 나타나는 군집이라고 판단했다.

■ Cluster 2

해당 군집은 5가지 군집 중 가장 높은 Affection 수치를 기록했다. 표준편차가 다른 군집에 비해 컸지만, 평균이 상대적으로 많이 높은 것으로 보아 높은 Affection은 이 군집을 나타내는 주요한 특징이라고 판단할 수 있었다. Aggression에서는 5개의 군집 중 가장 낮은 평균을 기록했다. 이 또한 분산이 비교적 컸지만 상대적으로 다른 군집보다 값이 많이 낮았기에 낮은 Aggression 역시 해당 군집을 나타내는 특징이라 판단할 수 있었다. Home-Distance에서는 가장 높은 평균치를 기록했고 표준편차 값도 크지 않았다. 높은 Home-Distance 역시 해당 군집을 나타내는 특징이라 판단했다.

- Cluster 2는 **높은 Affection, 낮은 Aggression**, **높은 Home-Distance**가 나타나는 군집

이라 판단했다.

■ Cluster 3

해당 군집에서 Affection 은 가장 0에 가까웠고, 표준편차 역시 작지 않았다. 그러므로 Affection 은 이 군집을 특징짓는 부분이 아니라고 판단했다. 반면 Aggression 에서 가장 높은 평균치를 기록했다. 표준편차 또한 크지 않았기 때문에 높은 Aggression 은 확실히 해당 군집을 나타내는 특징이라 판단했다. Silence 는 5개의 군집 중 0에 가장 가까웠고, 표준편차의 값은 가장 컸다. 이를 통해 Silence 는 이 군집을 나타내는 특징에 포함시키지 않았다. Home-Distance 역시 평균이 0에 가까웠고 표준편차가 작지 않았기 때문에 이 군집을 나타내는 주요한 특징에서 제외시켰다.

- Cluster 3은 **높은 Aggression** 이 나타나는 군집이라 판단했다.

■ Cluster 4

해당 군집은 Affection에서 두번째로 높은 평균치를 기록했다. 표준편차 값을 고려했을 때, 높은 Affection은 해당 군집을 나타내는 주요한 특징 중 하나라고 판단할 수 있었다. Aggression 은 다섯개의 군집 중 가장 0에 가까웠다. 음의 방향으로 치우쳐져 있었지만 평균이 워낙 0에 가까웠기 때문에 주요한 특징에서 제외했다. Silence는 5개중 2번째로 높은 평균치를 기록했다. 하지만 상대적으로 그 값들이 음의 방향보단 0에 가깝다. 그러므로 Silence는 주요 특징에서 제외했다. Home-Distance의 경우 가장 낮은 평균치를 기록했고, 표준편차의 값 역시 작았지만, 평균값이 0에 상당히 가까웠기 때문에 Home-Distance는 해당 군집을 나타내는 주요한 특징에서 제외했다.

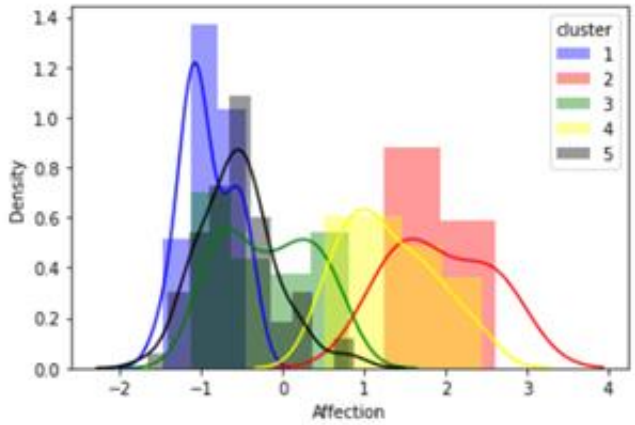
- Cluster 4는 **높은 Affection**이 나타나는 군집이라 판단했다.

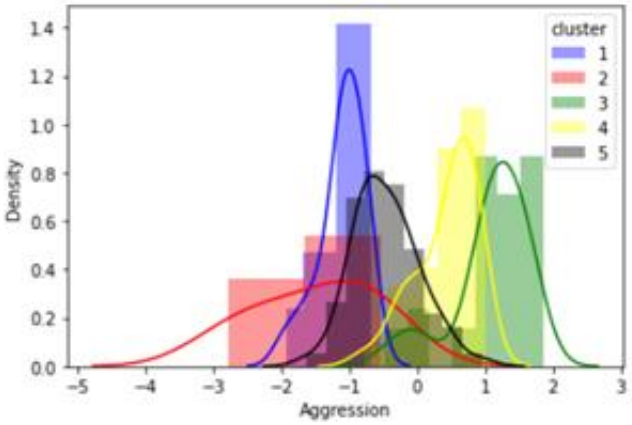
■ Cluster 5

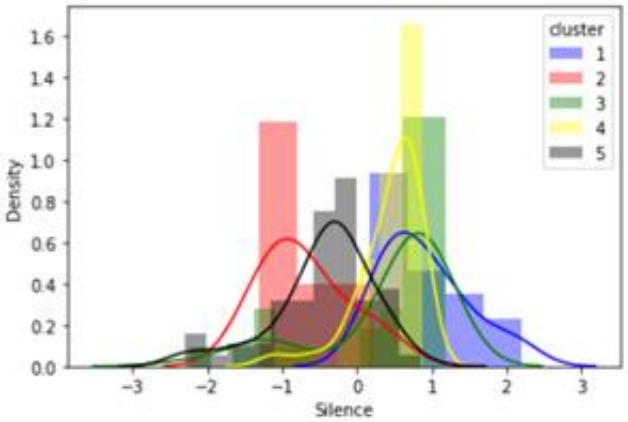
해당 군집은 Affection에서 두번째로 낮은 평균치를 기록했다. 표준편차 값을 고려했을 때, 낮은 Affection은 해당 군집을 나타내는 주요한 특징 중 하나라고 판단할 수 있었다. Aggression 의 평균치는 0에 가장 가까웠고, 표준편차의 값을 고려해 이 군집을 나타내는 특징에서 제외했다. Silence는 5개중 두번째로 낮은 평균치를 기록했다. 그러나 그 값이 가장 작은 Cluster2 보다는 0에 더 가까웠고 표준편차 역시 작지 않았기 때문에 Silence는 이 군집을 나타내는 특징에서 제외했다. Home-Distance도 두번째로 낮은 평균치를 기록했다. 하지만 표준편차가 가장 크기 때문에 주요 특징에서 제외했다.

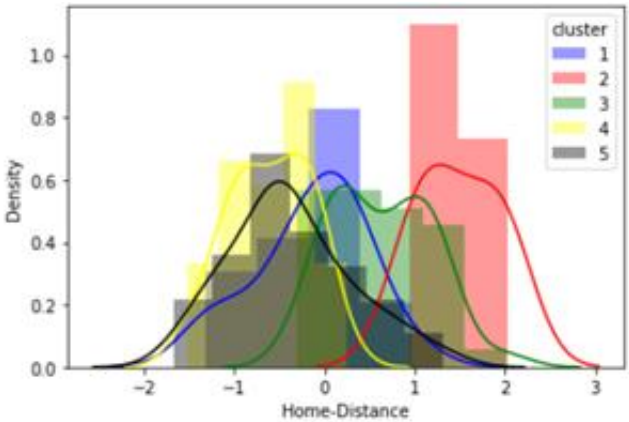
➢ cluster 5는 **낮은 Affection**이 나타나는 군집이라 판단했다.

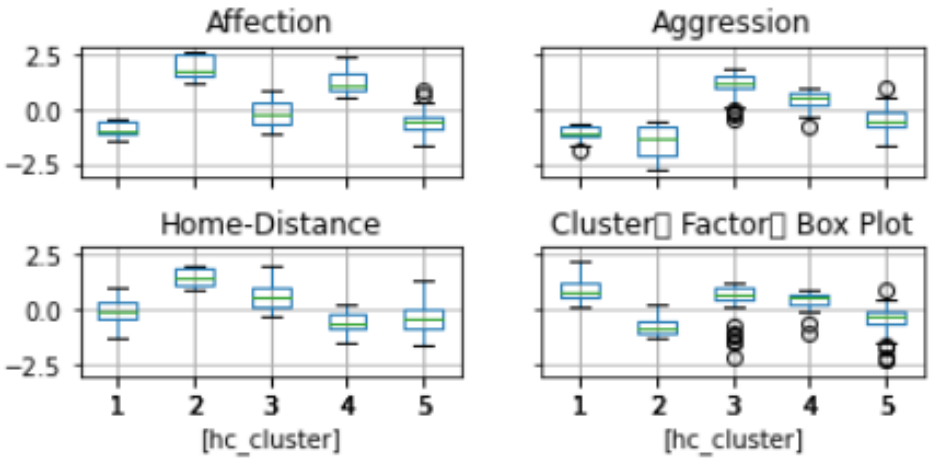
B. 군집 별 시각화











➢ 시각화를 통해 기존에 평균과 표준편차로만 도출했던 특징을 더욱 더 자세하게

검증하고 수정하였다.

■ Cluster 1

기존 평균과 표준편차 분석을 통해 **낮은 Affection, 낮은 Aggression, 높은 Silence**를 Cluster 1이 갖는 특징으로 선정했었다. 세가지의 경향성 모두 Factor별 분포 그래프와 Box plot에서 분명하게 드러났다. 특히 Affection에서 다른 군집들에 비해 매우 낮은 값을 굉장히 특징적으로 가진 것으로 드러났기 때문에, Affection에서 **매우 낮은 Affection**을 가진다고 수정했다.

■ Cluster 2

기존 평균과 표준편차 분석을 통해 **높은 Affection, 낮은 Aggression, 높은 Home-Distance**를 Cluster 2가 갖는 특징으로 선정했었다. 이 중 Aggression의 Factor별 분포 그래프를 통해 봤을 때 평균치는 가장 낮음에도 불구하고 분산이 굉장히 커 데이터 전체를 덮고 있었다. 그럼에도 왼쪽으로 치우쳐져 있었기 때문에, **조금 낮은 Aggression**의 경향성정도만 나타낸다고 수정했다.

■ Cluster 3

기존 평균과 표준편차 분석을 통해 **높은 Aggression**을 Cluster 3이 갖는 특징으로 선정했었다. 이런 특징은 Factor별 분포 그래프를 통해 확연히 확인할 수 있었다.

■ Cluster 4

기존 평균과 표준편차 분석을 통해 **높은 Affection, 낮은 Home-Distance**를 Cluster 4가 갖는 특징으로 선정했었다. Aggression의 경우 평균값이 0에 가까워 특징에서 제외했었다. 하지만 분포 그래프를 통해 값이 0보다는 1쪽에 훨씬 치우쳐 있는 것으로 나타났다. 이를 통해 **조금 높은 Aggression**을 cluster 4가 갖는 특징으로 추가했다.

■ Cluster 5

기존 평균과 표준편차 분석을 통해 낮은 Affection을 Cluster 5가 갖는 특징으로 선정했었다. Home-Distance의 경우 높은 표준편차를 이유로 특징에서 제외했었다. 해당 특징을 그

래프와 Box plot이 잘 드러내고 있는 것으로 보여 특별한 수정을 하지 않았다.

C. Silhouette 기법

마지막으로 Complete method가 얼마나 잘 군집화를 진행했는지 파악하기 위해 Silhouette Analysis를 진행하였다. 데이터셋 전체의 Silhouette Analysis Score는 **0.473점** 이었고, 각 군집 별 Silhouette 상관계수는 다음과 같이 나타났다.

```
hc_cluster
1    0.659192
2    0.538490
3    0.348856
4    0.521105
5    0.469839
Name: silhouette_coeff, dtype: float64
```

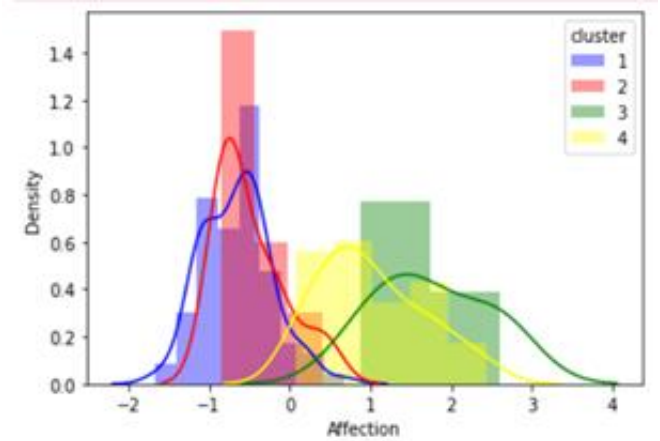
상대적으로 cluster 1, 2, 4가 더 군집화가 잘 되어있다고 볼 수 있다. 반면 3과 5는 평균에 비해 작기 때문에 상대적으로 군집화가 덜 되어있다고 결론지을 수 있었다.

D. Ward Method와 Complete Method의 비교 및 결론

■ **Ward Method**

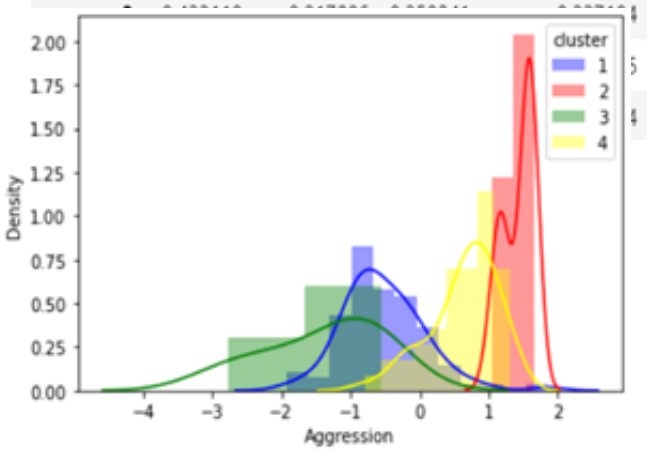
```
# 군집별 통계(평균)
cluster_g = data_1.groupby('hc_cluster')
cluster_g.mean()
```

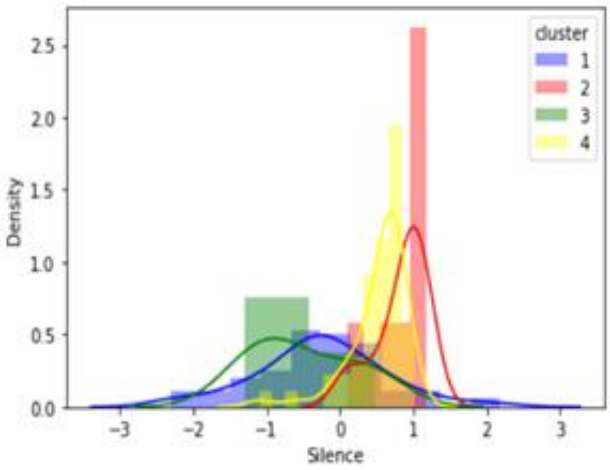
hc_cluster	Affection	Aggression	Silence	Home-Distance
1	-0.655044	-0.522949	-0.262649	-0.251451

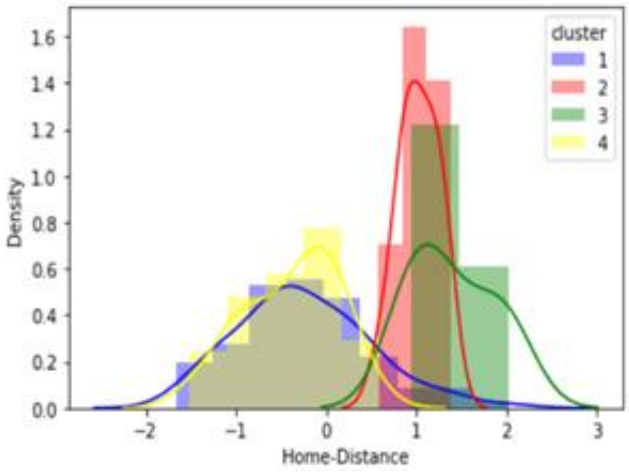



```
# 군집별 통계(표준편차)
cluster_g.std()
```

hc_cluster	Affection	Aggression	Silence	Home-Distance
1	0.428925	0.596120	0.881843	0.737378





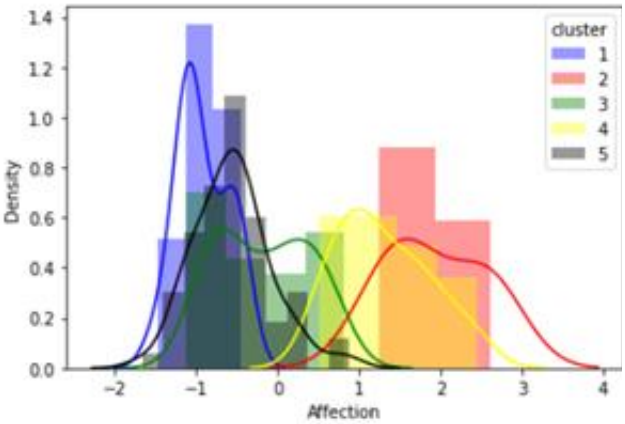


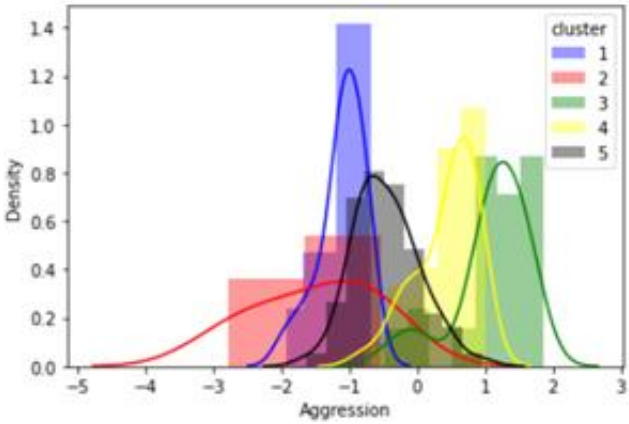
	Affection	Aggression	Silence	Home-Distance
Cluster1	조금 낮음	조금 낮음	-	-
Cluster2	조금 낮음	매우 높음	높음	높음
Cluster3	매우 높음	매우 낮음	-	매우 높음
Cluster4	높음	-	-	-

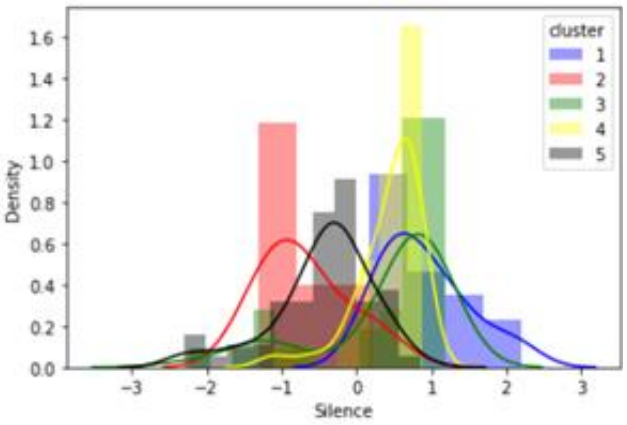
■ **Complete Method**

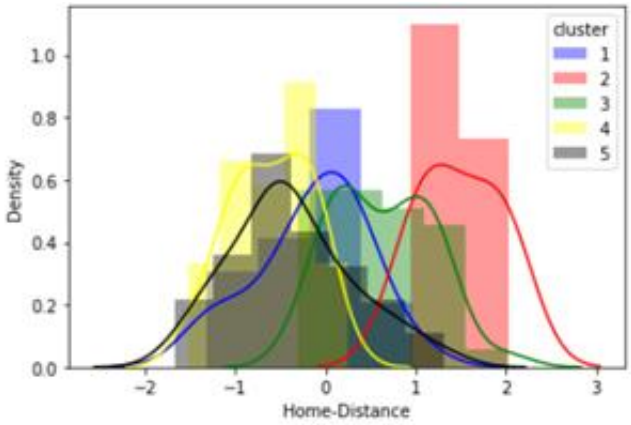
hc_cluster	Affection	Aggression	Silence	Home-Distance
1	-0.924776	-1.107302	0.903788	-0.139204
2	1.931822	-1.502666	-0.697677	1.487101
3	-0.178212	1.087447	0.398268	0.632163
4	1.296275	0.433311	0.439592	-0.609064
5	-0.561372	-0.461071	-0.439273	-0.378588

hc_cluster	Affection	Aggression	Silence	Home-Distance
1	0.310627	0.332065	0.585104	0.616132
2	0.609969	0.920135	0.586624	0.464254
3	0.563933	0.557447	0.896280	0.557389
4	0.563203	0.448568	0.431900	0.460699
5	0.473612	0.483852	0.661706	0.685117









	Affection	Aggression	Silence	Home-Distance
Cluster 1	매우 낮음	낮음	높음	-
Cluster 2	높음	조금 낮음	-	높음
Cluster 3	-	높음	-	-
Cluster 4	높음	-	-	-
Cluster 5	낮음	-	-	-

E. Hierarchical Clustering 결론

Ward method 의 데이터셋 전체의 Silhouette Analysis Score 는 0.373이었고, Complete method 의 데이터셋 전체의 Silhouette Analysis Score 는 0.473이었다. 이를 통해 상대적으로 **Complete Method 가 해당 데이터를 더 잘 군집화하는 방법이라고 결론** 지을 수 있었다. 다음은 각각 Ward 와 Complete method 의 각 군집별 Silhouette Analysis Score 를 나타낸 자료이다.

```
hc_cluster
1    0.244798
2    0.342382
3    0.655200
4    0.397388
5    0.362155
Name: silhouette_coeff, dtype: float64
```

<Ward Method Silhouette Score>

```
hc_cluster
1    0.659192
2    0.538490
3    0.348856
4    0.521105
5    0.469839
Name: silhouette_coeff, dtype: float64
```

<Complete Method Silhouette Score>

- Ward method 의 경우 3번 군집의 Silhouette Analysis Score 가 대략 0.65로 상대적으로 다른 군집과 차이가 많이 나타났고 나머지는 다 0.2~0.3 사이의 상대적으로 많이 작음

- 반면 Complete Cluster 의 경우 Cluster 간 실루엣 스코어 비교적 비슷했다. 이는

Ward method 는 3번 군집을 제외한 군집은 제대로 군집이 이루어지지 않았지만, Complete method 는 특정 군집에 치우치지 않고 모든 군집에 대해 군집화가 잘 이루어졌다고 판단할 수 있다.

III. K-Means Clustering

비계층적 군집분석(Non-Hierarchical Clustering)이란, 계층을 두지 않고 그룹화를 할 유사도 측정 방식에 따라 최적의 군집(cluster)를 계속적으로 찾아나가는 방법이다. 비계층적 군집분석에는 대표적으로 K-means Clustering과 DBSCAN 방식이 있는데, 본 과제에서는 이 중 K-means Clustering 방식을 이용한 군집 분석을 진행하고자 한다.

K-means는 가장 일반적인 방식으로, 군집 중심점이라는 특정한 임의의 지점을 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 군집화 기법이다. 군집 중심점(Centroid)는 선택된 포인트의 평균 지점으로 이동하고, 이동된 중심점에서 다시 가까운 포인트를 선택하고, 다시 중심점을 평균 지점으로 이동하는 프로세스를 반복적으로 수행한다. 이후 모든 데이터 포인트에서 더 이상 중심점의 이동이 없을 경우 반복을 멈추고 해당 중심점에 속하는 데이터 포인트들을 군집화하는 기법이다. K-means는 거리 기반 알고리즘으로 반복적으로 수행하기 때문에 속성이 많을 경우 정확도가 떨어져 차원 축소를 진행하고 해야하는 단점이 있지만, 본 과제에서는 위 계층적 군집분석과 마찬가지로 Factor Analysis를 진행한 데이터셋을 이용하여 수행하려고 한다.

K-Means Clustering의 특징은 다음과 같다.

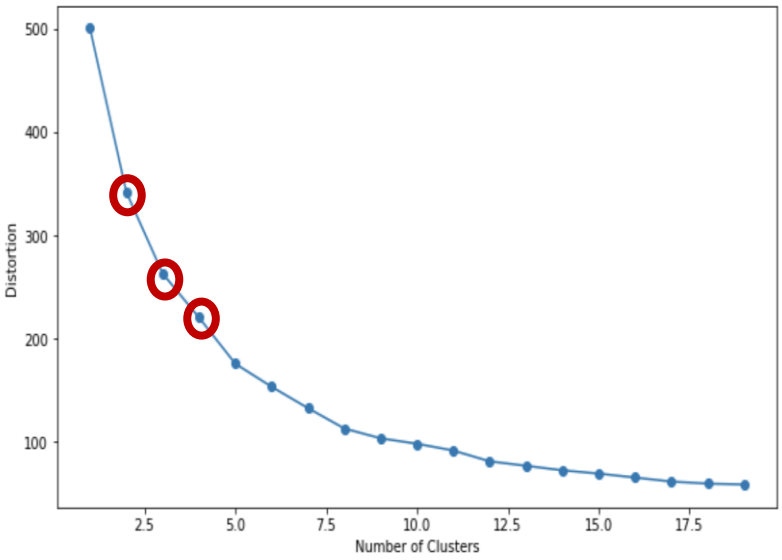
장점	단점
Scalability(확장성) – 큰 데이터에 사용 가능	서로 다른 크기, 모양의 군집을 잘 찾아내지 못한다.
짧은 계산 시간과 낮은 비용	Cluster의 수를 직접 정해줘야함(K)
이해하기 쉽다	특이값(Outlier)에 취약

i. 최적의 군집 수 결정

K-means Clustering을 진행하기 위해서는 최적의 군집 수 k값을 알아야한다. 군집 수를 결정하기 위해서는 대표적으로 Elbow 기법, Shilhouette 기법, 손실함수 기법이 있다. 손실함수 기법은 K-means clustering 결과값이 결국 머신러닝 기법에 사용되기 때문에, k 값을 하나의 파라미터로 보고 평가점수가 가장 좋게 나오는 K를 선택해서 사용한다. 하지만 본 분석을 위해 사용되는 데이터는 FA를 진행하여 Factor만을 속성으로 하는 데이터이고, 본 데이터의 종속변수가 범주형 변수이기 때문에 해당 기법을 사용하기 어렵다고 판단하였다. 따라서 본 과제에서는 Elbow 기법과 실루엣 기법을 사용하고자 한다.

1) Elbow 기법

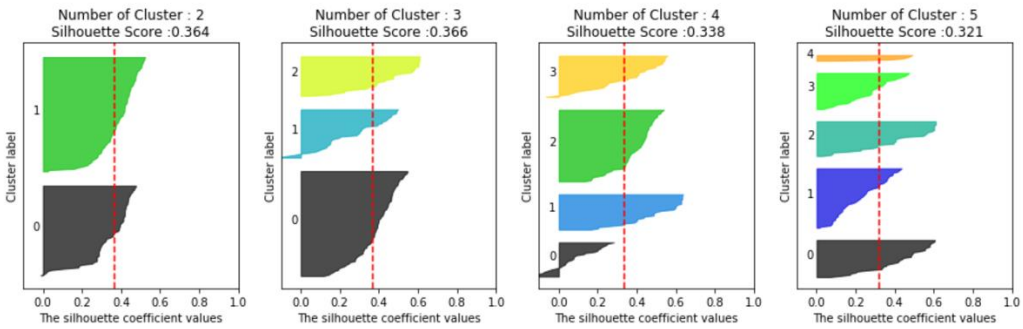
K-means 클러스터링은 군집 내 오차제곱합(SSE)의 값이 최소가 되도록 군집의 중심을 반복적으로 결정해 나가는 기법이다. 이때 SSE 값이 작다면 더 적합한 군집이라 할 수 있는데, 군집의 개수를 늘려나가면서 그린 그래프에서 SSE의 값이 줄어들다가 줄어드는 비율이 급격하게 작아지는 부분이 있는데, 이때 최적의 클러스터 개수가 된다.



데이터를 이용하여 Elbow 그래프를 그린 결과이다. 그래프 결과, 줄어드는 비율이 급격하게 작아지는 부분이 한 지점이 아니라 K=2,3,4 총 3가지 지점이 나왔다. 따라서 실루엣 기법을 통해 어떤 K가 가장 최적의 K 값인지 판단하고자 하였다.

2) Silhouette 기법

최적의 K를 구하기 위해 위에서 나온 2,3,4의 실루엣 계수를 구한 다음, 평균값이 가장 높은 것을 구하도록 하였다. 다음 그림은 실루엣 계수를 시각화한 결과이다.



실루엣 계수를 도출하여 비교 분석해본 결과, K-means에 최적화된 군집 개수는 3개인 것으로 나왔다. 먼저, 군집 수가 4이상이 되면 실루엣 계수의 편차가 큰 것으로 그래프를 통해 확인할 수 있었다. 예를 들어 4번의 경우 0번 군집은 전체 평균값보다 다 낮은 수치

를 갖고, 다른 군집과의 편차가 큰 것으로 확인 할 수 있다. 군집 개수가 2일 때와 3일 때의 평균값이 0.002로 근소한 차이가 나지만 k=3일 때가 최적값인 것으로 알 수 있었다.

3) 최종 결정

실루엣 기법을 통해서 K=3일 때가 최적 수인 것으로 파악되었지만, 실제 K값에 따른 각 군집별 통계값을 기반으로 2일 때와 3일 때 어떻게 다른지 파악하고자 한다.

k_means_cluster	Affection	Aggression	Silence	Home-Distance
0	-0.620492	-0.575556	-0.243232	-0.258218
1	0.695118	0.709823	0.490254	0.091521

k_means_cluster	Affection	Aggression	Silence	Home-Distance
0	1.365220	0.159157	0.298062	-0.307962
1	-0.651388	-0.565328	-0.230239	-0.294503
2	-0.116346	1.272740	0.644898	0.649108

왼쪽은 K=2일 때의 군집별 평균 값이고, 오른쪽은 k=3일 때 군집별 평균 값이다. K=2일 때 속성별 평균값을 보면 Cluster0은 전체가 음의 값, cluster1은 전체가 양의 값으로 상반되게 나와있어 속성별 설명이 어려운 것으로 볼 수 있다. 반면, k=3일 때에는 군집 내에 평균 값들이 음,양 고르게 분포되어있는 것을 확인 할 수 있었다. 실루엣 값 자체가 k=3일 때 제일 높았으며, 통계 값을 보고도 k=3이 최적화된 군집 개수임을 확인 할 수 있었다.

ii. Cluster 별 분석

A. 통계치를 통한 분석

k_means_cluster	Affection	Aggression	Silence	Home-Distance
0	1.365220	0.159157	0.298062	-0.307962
1	-0.651388	-0.565328	-0.230239	-0.294503
2	-0.116346	1.272740	0.644898	0.649108

(군집 별 평균)

k_means_cluster	Affection	Aggression	Silence	Home-Distance
0	0.598322	0.838417	0.581743	0.857142
1	0.430596	0.525940	0.861133	0.685521
2	0.570928	0.288489	0.664924	0.565215

(군집 별 표준편차)

K=3일 때 군집별 통계량을 나타낸 표들이다. 각 Cluster 별 속성의 평균을 분석한 결과, 먼저 Cluster 0은 Affection의 평균의 절대값이 높아 Affection에 관하여 잘 설명하고 있다고 할 수 있고, 상대적으로 나머지 3개의 요인들은 절대값이 비슷하게 낮은 것으로 확인할 수 있었다. 이때,

Aggression과 Home-Distance의 경우엔 절대값도 낮고 표준편차도 높아 Cluster0으로 설명하기 어렵다고 볼 수 있다.

Cluster1의 경우 Affection과 Aggression 두 개의 속성의 절대값이 높게 나왔으며, 나머지 두개의 절대값은 상대적으로 낮은 것을 확인할 수 있었다. 표준편차를 확인했을 때 나머지 두개 Silence와 Home-Distance가 상대적으로 높게 나왔다. 따라서 Cluster1으로 Affection과 Aggression을 잘 설명할 수 있으며, 나머지 두개 요인은 어려운 것으로 판단 할 수 있다.

Cluster 2의 경우 Aggression 속성의 절대값이 제일 높은 것으로 확인 할 수 있다. Affection 요인의 절대값이 제일 낮은 것을 볼 수 있었다. 특이한 부분은 나머지 두 Silence와 Home-Distance 요인의 절대값도 Affection과 비교하면 높은 수치임을 알 수 있었지만, Aggression의 평균 절대값이 훨씬 큰 점과, 표준편차가 작다는 점에서 Cluster 2는 Aggression에 대한 것으로만 하는 것이 좋을 것이라고 판단하였다.

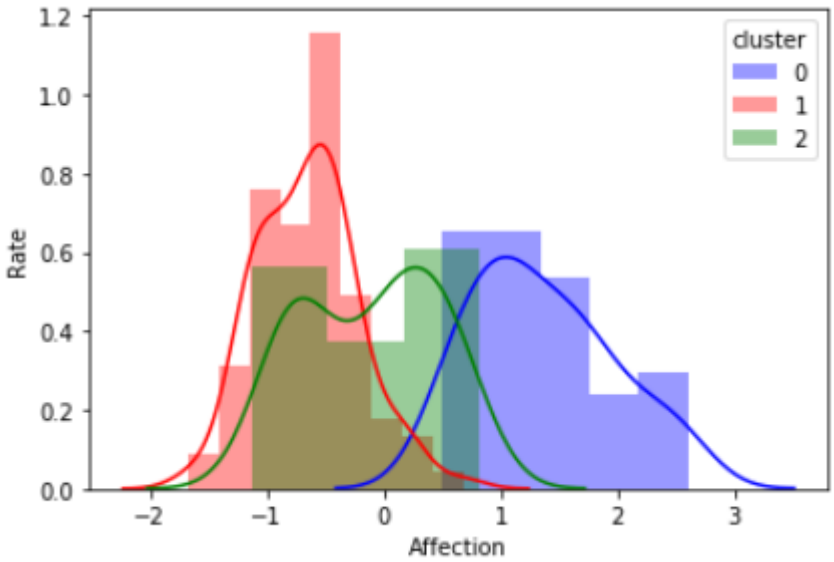
통계량을 통해 확인한 각 클러스터의 특성은 다음과 같다.

Cluster #	**특성**
Cluster 0	배우자에 대한 애정이 강함 사람 (0번: +Aff)
Cluster 1	배우자에 대한 애정과 공격성이 낮은 사람 (0번: -Aff, 1번: -Agg)
Cluster 2	배우자에 대한 공격성이 높은 사람 (1번: -Agg)

B. 시각화를 통한 비교

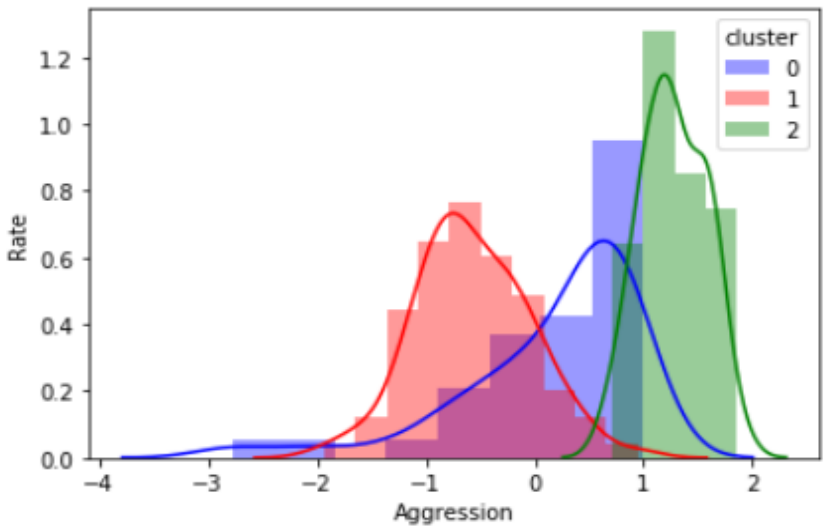
B.1 군집간 Factor 1개로 비교

1) Cluster x Affection



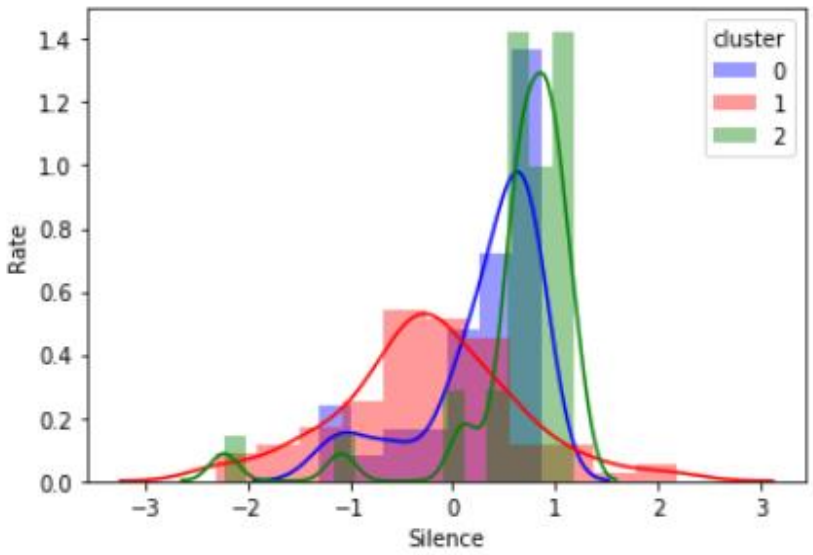
Cluster간 Affection을 비교한 것을 그래프로 나타냈다. 클러스터 0은 양의 경향, 클러스터 1는 음의 경향을 보이고, Cluster2는 음양에 걸쳐 고르게 분포되어 있는 것을 확인 할 수 있다. **따라서 Affection 속성을 통해 군집은 0과 1을 잘 설명할 수 있는 것으로 확인**이 된다. 그리고 클러스터 0은 나머지 1,2에 비해 양의 경향이 강한 것을 알 수 있어, Affection 속성의 관점에서 본 클러스터 0은 나머지 클러스터 1,2와 구분 지을 수 있다. 더불어, 해당 속성의 유의미한 클러스터 0과 1이 각각 양과 음으로 확실하게 구분되어 있는 것도 확인할 수 있었다.

2) Cluster x Aggression



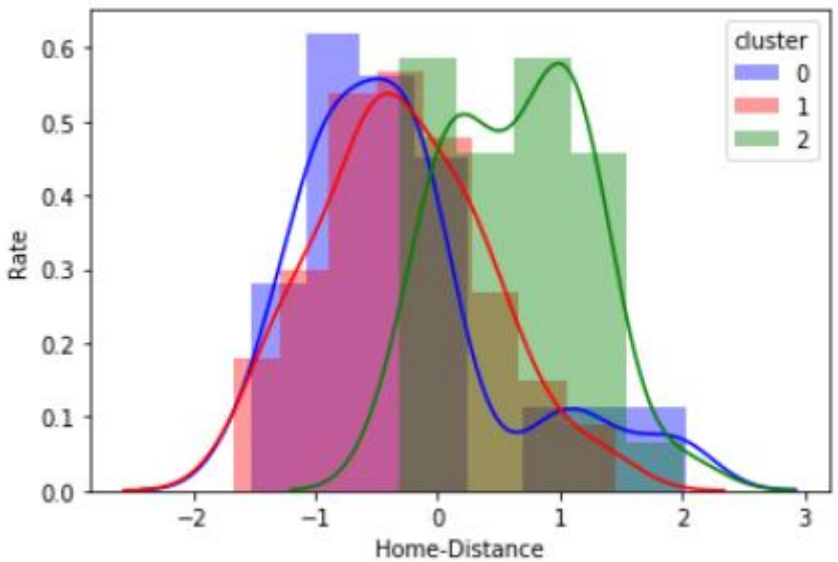
Cluster간 Aggression을 비교한 것을 그래프로 나타냈다. **Cluster 0과 2는 양의 경향을 띄지만, Cluster 2는 특히나 Aggression 속성을 통해 잘 나타날 수 있는 것으로 보인다**. Cluster 1은 -0.5의 값을 평균으로 정규분포를 이루고 있다. Cluster 2의 경우, Aggression 특성에 영향을 특히나 많이 받고 있다. Cluster2를 다른 군집과 구분짓는 특성으로 활용할 수 있을 것으로 보인다.

3) Cluster x Silence



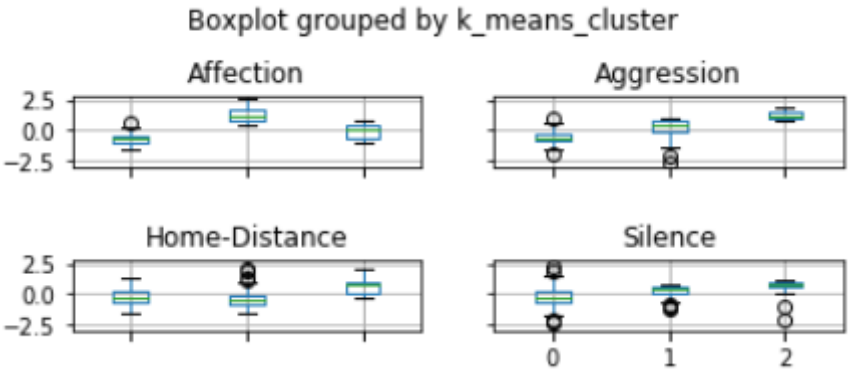
Cluster간 Silence를 비교한 것을 그래프로 나타냈다. **Cluster 0,2 는 양의 경향, Cluster 1은 음의 경향을 띈다.** 하지만 전반적으로 모든 Cluster에서 데이터가 퍼져있는 것을 확인할 수 있다. 각각의 Cluster를 Silence의 특성을 설명할 수 있지만, 이 특성을 통해 군집을 구분하기는 다소 어려워 보이지만, 음의 값이 나온 경우는 대부분 Cluster 1이라고 할 수 있을 것이다.

4) Cluster x Home-Distance



Cluster간 Home-Distance를 비교한 것으로 그래프로 나타냈다. **Cluster 0,1은 음의 경향, Cluster 2는 양의 경향을 띄는 것을 확인**할 수 있다. 하지만, 전반적으로 겹쳐지는 구간이 많은 것을 볼 수 있다. 특히, **Cluster 0,1은 Home-Distance로 구분하기는 매우 어려워 보인다. 양의 값을 가질 때, 대부분의 데이터가 Cluster 2**라고 할 수 있을 것이다.

5) **Factor 1가지로 분석한 결과**

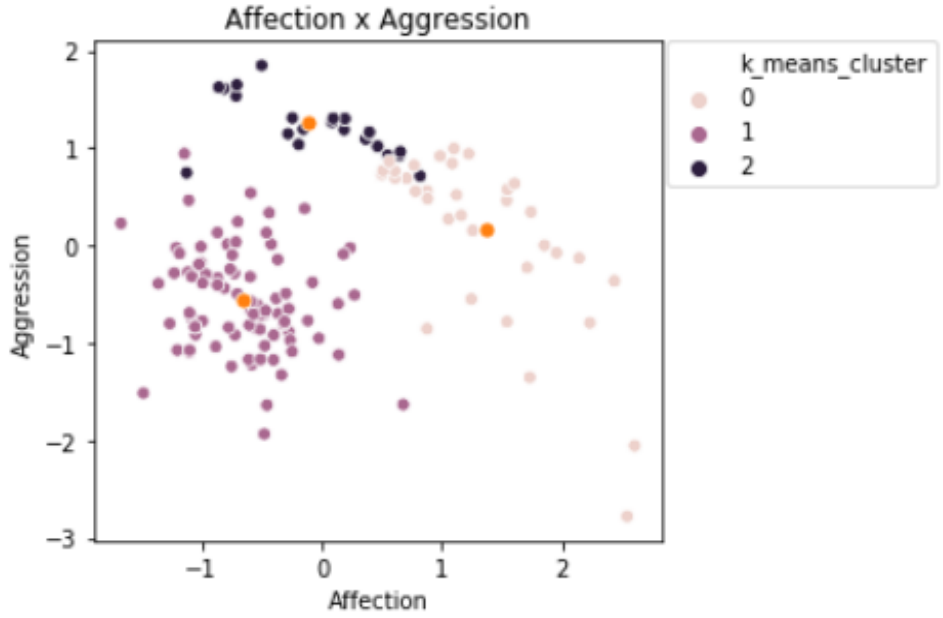


- **Affection과 Aggression의 속성의 경우**엔 해당 속성들을 잘 설명하는 Cluster가 명확하게 보이는 것으로 확인함.

- **silence랑 home-distance는 확인하기 어려움**. 속성 별로 잘 설명할 수 있는 군집을 확인할 수 있었지만, 군집간 명확한 차이가 보이기는 어려웠음

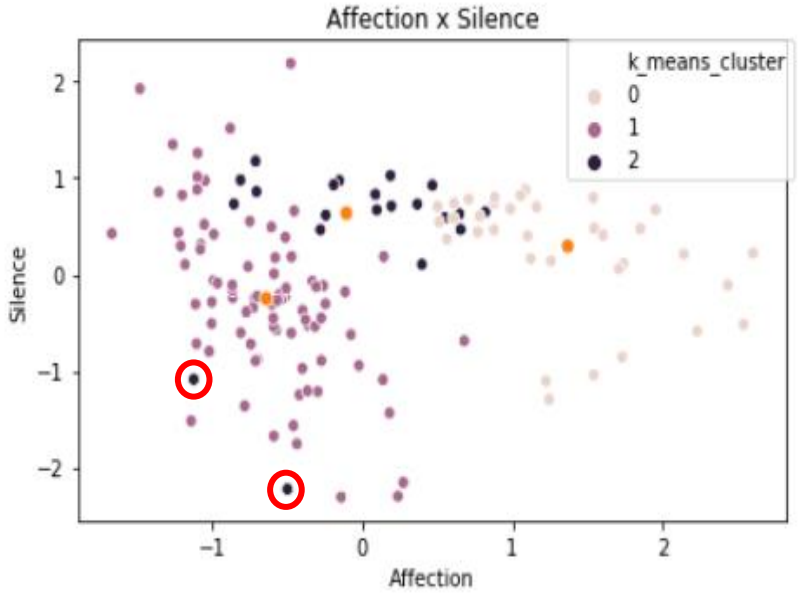
B.2 군집간 Factor 2개로 비교

1) Affection x Aggression



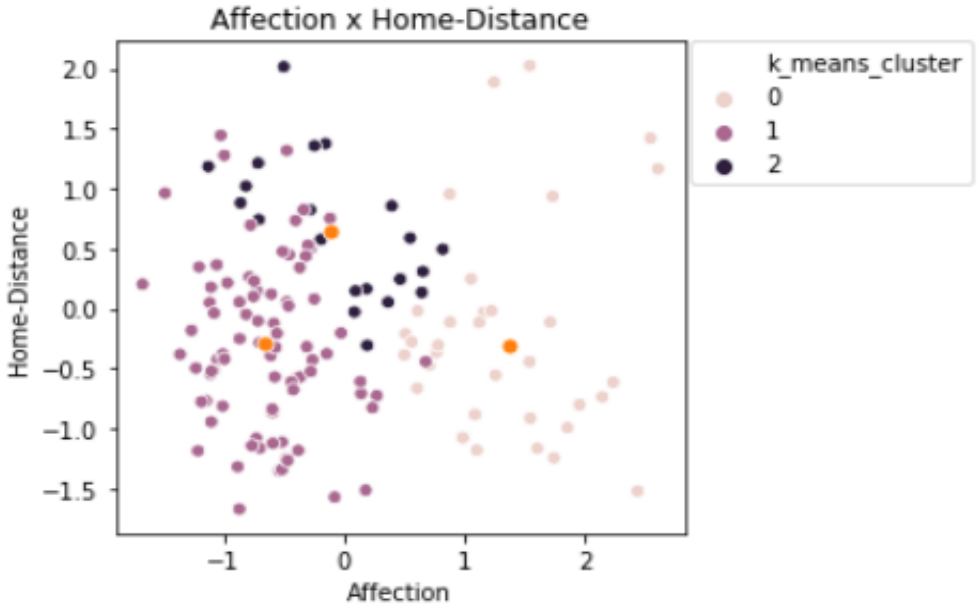
Affection과 Aggression의 특성에 따라 Cluster를 구분지었다. Cluster 2의 경우 Cluster 0과 Cluster 1 간의 겹치는 데이터가 존재하지만, 4개의 데이터를 제외하면, Cluster 간 구분을 뚜렷이 볼 수 있다. **두 Factor로 Cluster를 구분지을 수 있어보인다.**

2) Affection x Silence



Affection과 Silence의 특성에 따라 Cluster를 구분지었다. Cluster 2의 경우 Cluster 0과 1에서 겹치는 데이터가 여럿 확인할 수 있었다. 빨간색으로 표시한 데이터는 Affection과 Silence로 확인했을 경우는 Cluster 2보다 Cluster 1에 가까운 경향을 보였다. **Cluster 0과 Cluster 2간의 구분은 겹치는 구간 없이 보여주는 모습을 보여주었다.**

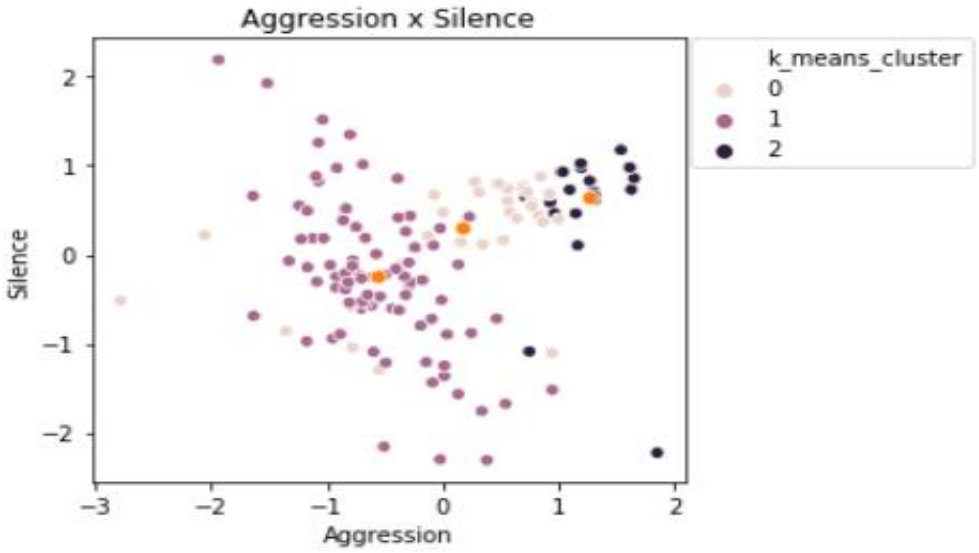
3) Affection x Home-Distance



Affection과 Home-Distance의 특성에 따라 Cluster를 구분지었다. 두 Factor와 Cluster를

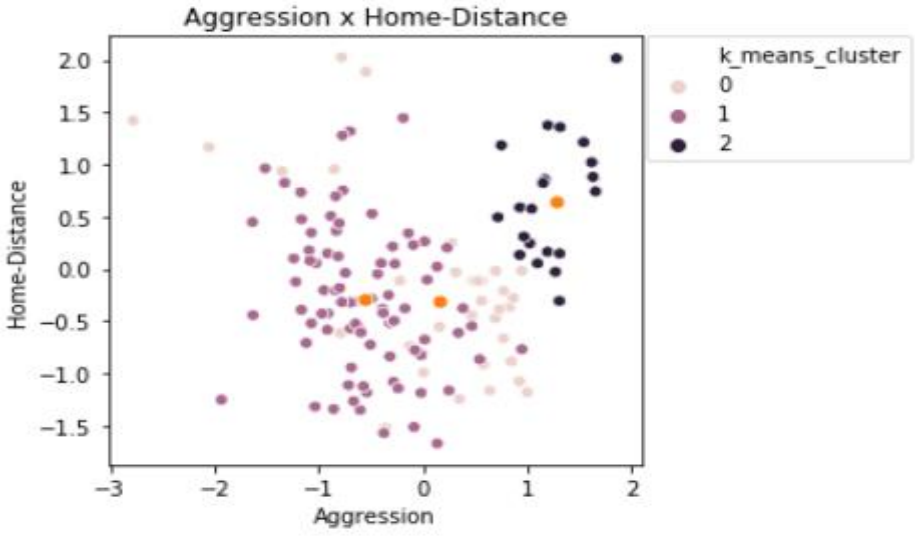
Scatterplot으로 확인했을 때, Cluster 2와 나머지 Cluster간 겹치는 데이터 구간이 존재했다. Cluster 1과 Cluster 2간의 구분은 잘 짓는 것을 볼 수 있었다.

4) Aggression x Silence



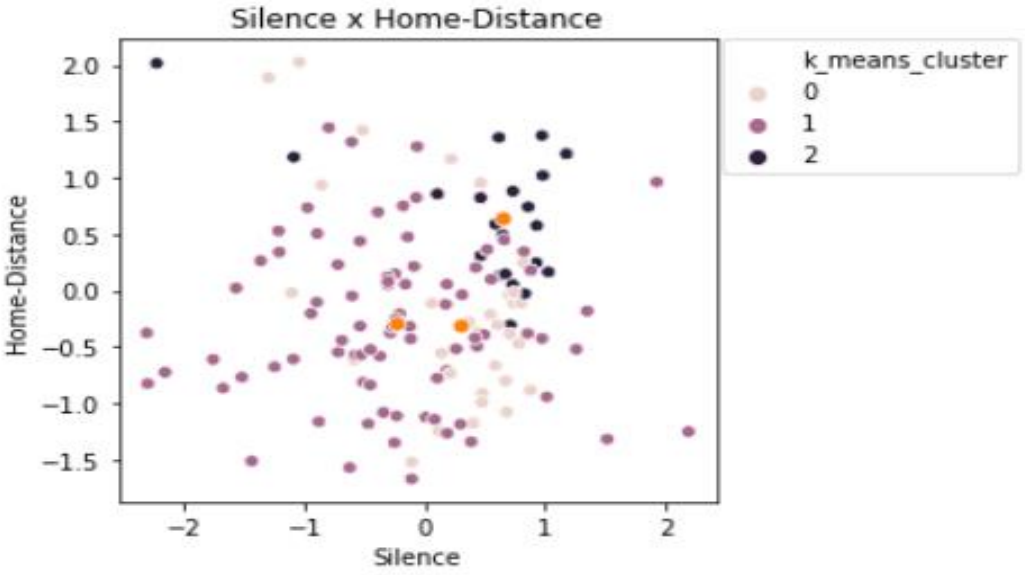
Aggression과 Silence의 특성에 따라 cluster를 구분지었다. 두 특성에 따라 Cluster를 비교했을 때는, Cluster 0에 해당하는 데이터는 골고루 분포하고 있어서, Cluster 0을 나머지 Cluster와 두 특성으로 비교하기는 어려워보였다. Cluster 1과 Cluster 2만 비교했을 경우는 겹쳐지는 데이터가 두 데이터가 존재하는 것으로 확인이 되지만, **Aggression과 Silence에 따라 Cluster 1과 2는 명확히 구분되는 것을 볼 수 있다.**

5) Aggression x Home-Distance



Aggression과 Home-Distance의 특성에 따라 Cluster를 구분지었다. 두 특성에 따라 Cluster를 비교했을 때는, Cluster 0에 해당하는 데이터는 골고루 분포하고 있어서, **Cluster 0을 나머지 Cluster와 두 특성으로 비교하기는 어려워보였다**. Cluster 1과 2를 비교했을 경우 겹쳐지는 데이터 구간이 없고, Aggression과 Home-Distance 간으로 구분이 될 것으로 보인다.

6) Silence x Home-Distance



Silence와 Home-Distance의 특성에 따라 Cluster를 구분지었다. Silence와 Home-Distance로 Cluster간 구분 짓기는 어려워 보였다. **두 특성에 따라 세 군집 모두 고르게 분포되어 있는 형태를 띄었고, 데이터가 서로 교차되는 구간이 상당히 존재했다**.

7) Factor 2가지로 분석한 결과

- Factor 2가지를 통해 Cluster 간 분석한 결과, **Affection과 Aggression 두 가지를 했을 때, 가장 뚜렷하게 Cluster간 구분을 지을 수 있는 모습**을 볼 수 있었다. 다른 Plot들도 확인 했을 경우, Affection과 다른 Factor간의 조합을 통해서는 어느정도 Cluster간 구분이 지어지는 모습을 확인할 수 있었다.

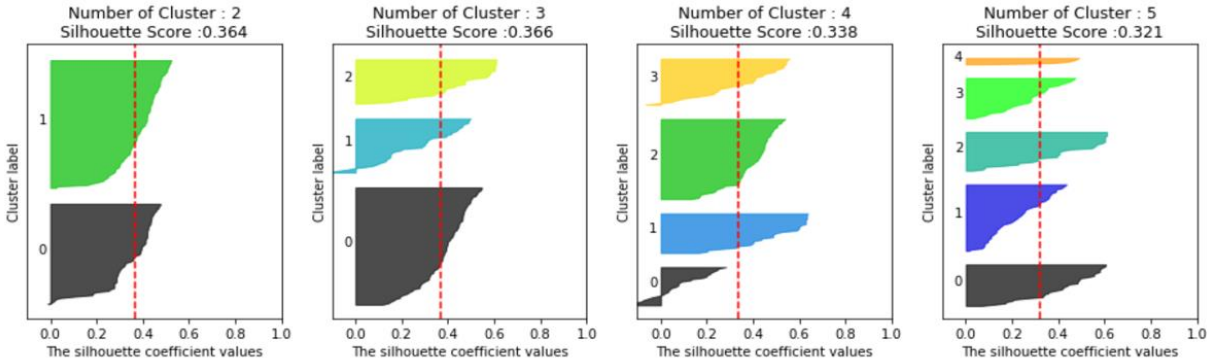
- Affection을 제외한 나머지 Aggression, Silence, Home-Distance 특성들 간의 조합 결과는 Cluster간 명확성을 보여주진 않았다.

C. K-Means Clustering 결과

- 통계랑으로 Cluster별 특성에 따른 Naming을 시각화 과정을 통해 확인했다. 그 결과, Cluster 별 특성을 갖고 있다고 결론 지었다. 또한 특성에 따라 Cluster의 Naming을 다음과 같이 하였다. Cluster 0을 배우자에 대해 애정이 강하기 때문에 '사랑꾼 그룹', Cluster 1을 배우자에 대한 애정과 공격이 낮기 때문에 '무관심한 그룹', Cluster 2를 배우자에 대한 공격성이 높은 사람이기 때문에 '위험한 배우자 그룹' 으로 지었다.

Cluster #	**특성**
Cluster 0	배우자에 대한 애정이 강함 사람 (0번: +Aff)
Cluster 1	배우자에 대한 애정과 공격성이 낮은 사람 (0번: -Aff, 1번: -Agg)
Cluster 2	배우자에 대한 공격성이 높은 사람 (1번: -Agg)

- 실루엣 점수를 통해 타당성을 확인한 결과, K=3 일 때 가장 점수가 높았다. 하지만 일반적으로 실루엣 점수가 일반적으로 0.5 이상일 때 군집 결과가 타당한 것으로 평가한다고 하는데, 실루엣 점수로는 약간 아쉬운 모습을 보였다.



4. 결과

■ 계층적 클러스터링 vs 비계층적 클러스터링

본 과제에서는 계층적 클러스터링 중 Ward와 Complete 방식을 진행하였고, 비계층적 클러스터링으로는 K-means 클러스터링을 진행하였다. 계층적 클러스터링은 두 가지 방식 중 실루엣 점수가 높았던 **Complete 방식이 가장 최적의 군집임을 결론**을 내었다.

Cluster #	**특성 (K-means 클러스터링)**

Cluster 0	배우자에 대한 애정이 강함 사람 (0번: +Aff)
Cluster 1	배우자에 대한 애정과 공격성이 낮은 사람 (0번: -Aff, 1번: -Agg)
Cluster 2	배우자에 대한 공격성이 높은 사람 (1번: -Agg)

Cluster #	**특성 (Complete)**
Cluster 1	배우자에 대한 애정이 강하고 공격성이 낮고 논쟁 시 침묵을 유지하는 사람 사람 (0번: +Aff, 1번: -Agg, 2번: +Sil)
Cluster 2	배우자에 대한 애정이 높고, 집 내에 거리감을 낮게 느끼는 사람 (0번: +Aff, 3번: -HD)
Cluster 3	배우자에 대한 공격성이 높은 사람 (1번: +Agg)
Cluster 4	배우자에 대한 애정이 높은 사람 (0번: +Aff)
Cluster 5	배우자에 대한 애정이 낮은 사람 (0번: -Aff)

두 클러스터링 방식을 실루엣 점수를 도출한 결과, K-means로 한 군집들의 실루엣 계수 평균 점수는 0.366 (K=3)이었으며, Complete은 0.473(K=3)이었다. 따라서 실루엣 점수가 더 높은 complete method로 진행한 계층적 클러스터링이 최적 클러스터링으로 결론을 내렸다. 클러스터링 실루엣 점수는 보통 0.5 이상이 나와야 좋은 클러스터링이라고 하지만, 두 방식 모두 0.5 미만의 숫자가 나와 아쉬운 부분이 있었다. 시각화된 그래프를 보면 클러스터 간 구분이 확실한 경우도 있지만 그렇지 않은 경우도 있는 것으로 확인 할 수 있었다.