데이터마이닝 이론 및 응용 2주차 과제

PCA & FA

요조 [곽지운, 박현준, 이정현, 최동훈]

1. 서론

1.1데이터 설명

이번 2주차 과제에 활용할 데이터로 Kaggle의 Divorce Prediction 데이터를 선정하였다. 해당 데이터는 터키의 다양한 지역에서 170명의 기혼자, 이혼자를 대상으로 면대면 설문조사를 진행해 얻어낸 데이터로, 모든 응답은 0과 4 사이로 응답되었다. 0은 절대 그렇지 않다, 1은 가끔 그렇다, 2는 일반적으로 그렇다, 3은 자주 그렇다, 그리고 4는 항상 그렇다를 의미한다.

이 자료는 총 9350개의 관측치로 구성되어 있고, 변수는 독립변수(질문) 54개와 종속변수(이혼 여부) 1개로 이루어져 있다. 종속변수는 이혼한 경우 1, 그렇지 않은 경우 0으로 범주형 변수이다. 독립변수는 '다툼 중 한 명이 사과하면 다툼이 끝난다.', '서로의 차이를 인정할 수 있다.', '상대와 보내는 시간이 특별하다.' 등등 총 54개의 질문에 대한 응답 값으로 0과 4 사이의 연속형 변수고, 각 질문을 Q1~Q54로 명명하였다.

각 질문의 내용은 다음과 같다.

Question 내용
1 | 토론이 악화될 때 우리 중 한 명이 사과하면 토론이 끝납니다.
2 | 때때로 상황이 어려워지더라도 우리의 차이점을 무시할 수 있다는 것을 알고 있습니다.
3 | 우리가 필요로 할 때 그것을 우리는 처음부터 내 배우자와 우리의 토론을 하고 수정할 수 있습니다.
4 | 내가 배우자 와 연락하여 연락하면 결국 효과가 있을 것입니다.
5 | 아내와 함께 보낸 시간은 우리에게 특별합니다.
6 | 우리는 파트너로서 집에서 시간이 없습니다.
7 | 우리는 가족보다는 집에서 같은 환경을 공유하는 두 명의 낯선 사람과 같습니다.
8 | 저는 아내와 함께 휴가를 즐깁니다.
9 | 저는 아내와 함께 여행하는 것을 좋아합니다.
10 | 대부분의 목표는 배우자에게 공통적입니다.

11 | 언젠가 돌아 보면 배우자와 나는 서로 조화를 이루고있는 것을 보게 될 것 같습니다.
12 | 내 배우자와 나는 개인적 자유 측면에서 비슷한 가치관을 가지고 있습니다.
13 | 배우자와 나는 비슷한 오락 감각을 가지고 있습니다.
14 | 사람 (어린이 친구 등)에 대한 대부분의 목표 는 동일합니다.
15 | 배우자와의 꿈은 비슷하고 조화 롭습니다.
16 | 우리는 사랑이 무엇인지에 대해 배우자와 양립 할 수 있습니다.
17 | 우리는 배우자와 우리 삶에서 행복 해지는 것에 대해 같은 견해를 공유합니다.
18 | 배우자와 나는 결혼이 어떻게되어야하는지에 대해 비슷한 생각을 가지고 있습니다.
19 | 배우자와 나는 결혼 생활에서 역할이 어떻게되어야하는지 비슷한 생각을 가지고 있습니다.
20 | 배우자와 나는 신뢰에 대해 비슷한 가치를 가지고 있습니다.
21 | 아내가 뭘 좋아하는지 정확히 알고 있습니다.
22 | 배우자가 아플 때 어떻게 보살핌을 받고 싶어하는지 알고 있습니다.
23 | 배우자가 좋아하는 음식을 알고 있습니다
24 | 내 배우자가 자신의 삶에서 어떤 스트레스를 받고 있는지 말해 줄 수 있습니다.
25 | 배우자의 내면에 대해 알고 있습니다.
26 | 배우자의 기본적인 불안을 알고 있습니다.
27 | 제 배우자의 현재 스트레스 원인이 무엇인지 압니다.
28 | 배우자의 희망과 소원을 알고 있습니다.
29 | 저는 제 배우자를 아주 잘 압니다.
30 | 배우자의 친구와 그들의 사회적 관계를 알고 있습니다.
31 | 배우자와 논쟁 할 때 공격적으로 느껴집니다.
32 | 배우자와 대화 할 때 보통 '당신은 항상'또는 '당신은 결코'와 같은 표현을 사용합니다.
33 | 토론 중에 배우자의 성격에 대해 부정적인 말을 할 수 있습니다.
34 | 토론 중에 불쾌한 표현을 사용할 수 있습니다.
35 | 토론 중에 배우자를 모욕할 수 있습니다.
36 | 저는 토론할 때 수치스러울 수 있습니다.
37 | 배우자와의 대화가 차분하지 않습니다.
38 | 배우자가 주제를 여는 방식이 싫습니다.
39 | 우리의 토론은 종종 갑자기 발생합니다.
40 | 우리는 무슨 일이 일어나고 있는지 알기 전에 토론을 시작하고 있습니다.
41 | 내가 배우자와 이야기 할 때 내 차분함이 갑자기 깨집니다.
42 | 내가 배우자와 말다툼을하면 나가기 만하고 아무 말도하지 않습니다.
43 | 저는 환경을 조금 진정시키기 위해 대부분 침묵합니다.39| 우리의 토론은 종종 갑자기 발생합니다.

44 | 때로는 잠시 집을 떠나는 것이 좋다고 생각합니다
45 | 배우자와상의하는 것보다 묵비권을 행사하고 싶습니다.
46 | 내가 토론에서 옳다고해도 나는 내 배우자를 해치기 위해 침묵을 지킵니다.
47 | 배우자 와상의 할 때 분노를 다 스릴 수 없을까 두려워 침묵을 지킵니다.
48 | 저는 토론에서 옳다고 느낍니다
49 | 나는 내가 기소 된 것과 관련이 없습니다.
50 | 실제로 내가 기소 된 것에 대해 유죄가되는 사람은 아닙니다.
51 | 나는 가정의 문제에 대해 틀린 사람이 아닙니다.
52 | 배우자에게 자신의 부적절함에 대해 이야기하는 것을 주저하지 않을 것입니다.
53 | 토론할 때 배우자에게 부적절 함을 상기시킵니다.
54 | 배우자에게 자신의 무능력에 대해 말하는 것이 두렵지 않습니다.

1.2데이터의 특징

독립변수들의 공분산 행렬과 상관계수 행렬을 확인해보았다. 다음은 공분산행렬의 일부를 나타낸 행렬이다.

```
In [14]: df.cov().head()
```

Out[14]:

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	...	Q45	Q46	Q47	Q48
Q1	2.647964	1.957466	1.917508	2.019701	2.340620	0.422416	0.625896	2.019005	2.144588	1.827845	...	1.245179	0.893561	1.504629	1.172572
Q2	1.957466	2.156944	1.675252	1.748277	1.964079	0.136547	0.551201	1.962861	1.893909	1.633206	...	1.077341	0.784755	1.437661	1.075322
Q3	1.917508	1.675252	2.003481	1.717717	1.849983	0.336582	0.590324	1.657501	1.800905	1.515141	...	0.907414	0.598329	1.223808	1.027497
Q4	2.019701	1.748277	1.717717	2.263000	2.009607	0.252906	0.641907	1.857153	1.943056	1.868221	...	1.008145	0.702123	1.318413	1.078246
Q5	2.340620	1.964079	1.849983	2.009607	2.663975	0.439471	0.559415	2.214967	2.330108	1.911034	...	1.448451	1.054020	1.864532	1.223738

5 rows × 54 columns

같은 방식으로 구한 상관계수 행렬은 다음과 같다.

```
In [15]: df.corr().head()
```

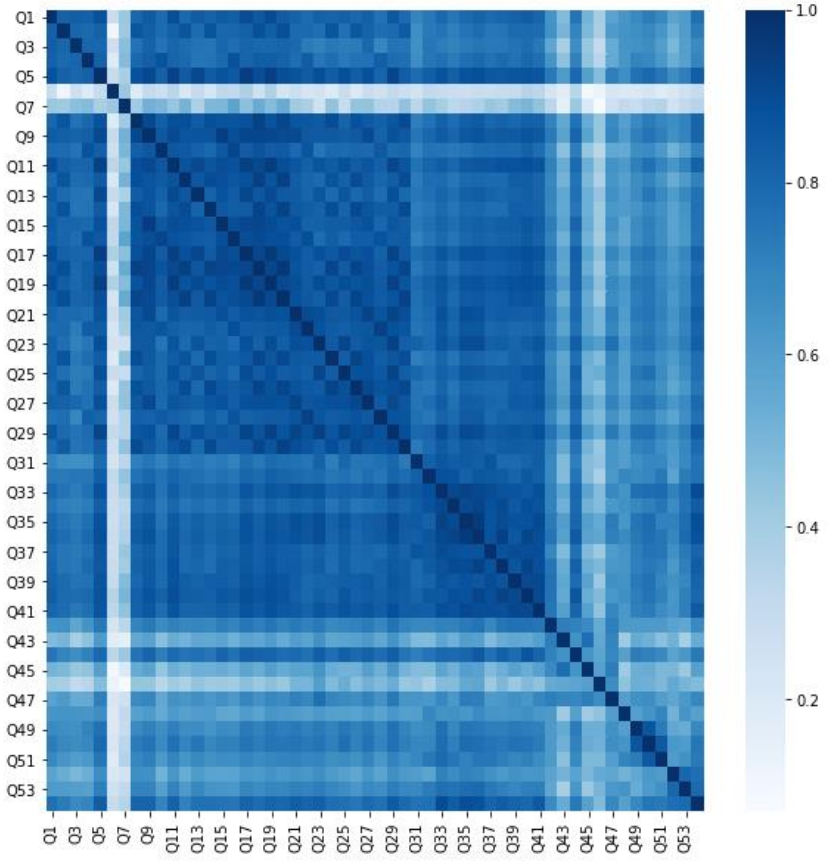
Out[15]:

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	...	Q45	Q46	Q47	Q48
Q1	1.000000	0.819066	0.832508	0.825066	0.881272	0.287140	0.427989	0.802357	0.845916	0.790183	...	0.510160	0.400296	0.582693	0.633564
Q2	0.819066	1.000000	0.805876	0.791313	0.819360	0.102843	0.417616	0.864284	0.827711	0.782286	...	0.489062	0.389519	0.616884	0.643762
Q3	0.832508	0.805876	1.000000	0.806709	0.800774	0.263032	0.464071	0.757264	0.816653	0.753017	...	0.427409	0.308149	0.544863	0.638256
Q4	0.825066	0.791313	0.806709	1.000000	0.818472	0.185963	0.474806	0.798347	0.829053	0.873636	...	0.446798	0.340240	0.552301	0.630205
Q5	0.881272	0.819360	0.800774	0.818472	1.000000	0.297834	0.381378	0.877584	0.916327	0.823659	...	0.591656	0.470758	0.719899	0.659220

5 rows × 54 columns

각 데이터의 평균이 다르고 크기가 다르기 때문에 해당 과제에서는 표준화한 데이터를 기준으로 PCA를 진행하기로 결정했다. 이때 필요한 것이 바로 상관계수 행렬이며, 이를 히트맵으로 시각화하여 대략적인 분포를 알아보고자 했다. 다음은 데이터의 상관계수 행렬을 시각화한 그림이다.

Q11과 Q20, Q34와 Q35는 다른 데이터들보다 높은 상관계수를 가지는 것으로 확인되었고, Q6, Q7은 나머지 변수들과 상관계수가 굉장히 낮게 나왔다. Q43, Q45, Q46 역시 Q6, Q7보단 덜하지만 다른 변수들과 상관계수가 낮게 나왔다. Q11과 Q20은 각각 배우자간 화합을 묻는 질문과 배우자간 신뢰를 묻는 질문이었다. 이를 통해 배우자간 화합과 신뢰가 높은 상관관계를 가짐을 알 수 있었다. Q6, Q7은 '가정에서



파트너로서 시간을 갖지 않는다.', '둘이 같이 집에 있을 땐 가족이 아니라 서로 낯선 사람 같다.' 였다. 해당 질문은 해석의 여지가 다양해서 다른 독립변수들과의 상관관계가 유독 낮게 나온 것이라 추론할 수 있었다. 이 데이터 중 상관관계를 갖는 변수들이 있기 때문에 PCA 분석에 사용하기 적합한 데이터라 판단하였다.

2. Principal Components Analysis(주성분분석)

2.1PCA의 정의와 원리

PCA란 변수의 수가 많은 고차원 데이터는 시각적으로 표현하기도 어려우며 불필요한 변수도 존재하기 때문에 이를 해결하고자 새로운 변수를 추출해 차원을 낮추는 분석법을 의미한다. 원래 데이터의 분산을 최대한 보존하면서 새로운 변수를 추출하는데, 이때 새로운 변수는 기존 변수의 선형조합으로 생성된다. 기존 변수들의 선형결합으로 만들어낸 새로운 변수를 '주성분'이라고 하며, 이 주성분에서 기존 독립변수의 계수들을 통해 각 변수들이 새로운 주성분에서 차지하는 영향을 대략적으로 파악할 수 있다.

새로 만들어진 주성분에서 기존 독립변수들의 계수는 Eigen vector의 값으로 이루어지며, Eigen value의 값이 가장 큰 고유벡터가 제1주성분의 계수가 된다. 이후 두번째로 큰 Eigen value을 가지는 Eigen vector로 제2주성분을 만들며 각 주성분들은 모두 선형적으로 독립이다.

PCA를 진행하는 방법으로 크게 두가지 방법이 있다. 하나는 원변수를 통해 PCA를 진행하는 방법이다. 이 방법은 Covariance Matrix(공분산 행렬)을 통해 이루어진다. 하지만 이는 독립변수의 규모가 서로 다른 경우를 반영하지 못할 수도 있다. 이를 보충하기 위해 표준화된 변수를 통해 PCA를 진행하는 방법이 있다. 이 경우 굳이 독립변수 데이터를 다 표준화한 후 진행하지 않아도 되고, Correlation Matrix를 사용해 PCA를 진행하면 된다. 본 과제에서는 Correlation Matrix가 더 적합하다 판단해 Correlation Matrix를 통해 PCA를 진행하였다.

2.2Correlation Matrix를 통한 PCA

데이터의 Correlation Matrix에서 FactorAnalyzer함수를 통해 eigen value값을 구했다. 이제 여기서 데이터 전체 분산을 적절히 반영하는 PC를 선정해야 한다. Eigen Value의 값을 기준으로 선정하는 방법, 누적된 설명가능 분산 비율을 기준으로 선정하는 방법, 두가지 방법 모두 고려하여 선정하는 방법 총 3가지 방법으로 PC를 선정을 해보았다.

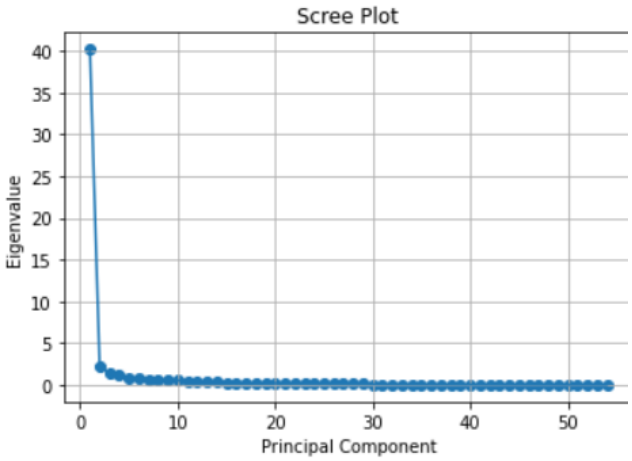
```
# Eigen value
fa = FactorAnalyzer(n_factors=divorce_data_scaled.shape[1], rotation=None) # 54개의 component
fa.fit(divorce_data_scaled)

ev, v = fa.get_eigenvalues()
ev # 54개의 Principal Component에 대응되는 eigen value들
```

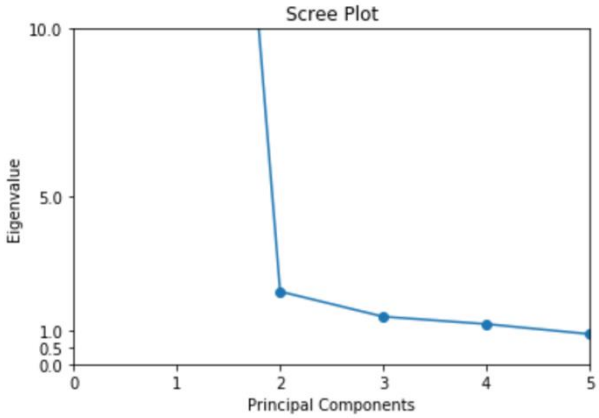
```
array([4.01756867e+01, 2.16531589e+00, 1.41651436e+00, 1.19405368e+00,
       8.96249766e-01, 7.88897467e-01, 6.98636632e-01, 5.95362778e-01,
       5.68366461e-01, 5.29115848e-01, 4.28671399e-01, 3.90369411e-01,
       3.62469358e-01, 3.19198119e-01, 2.83031788e-01, 2.70341457e-01,
       2.52520084e-01, 2.20402276e-01, 2.02203198e-01, 1.91133304e-01,
       1.75596029e-01, 1.65126304e-01, 1.38913064e-01, 1.32199503e-01,
       1.19176457e-01, 1.07826952e-01, 1.03507589e-01, 1.00321611e-01,
       9.48779953e-02, 8.83932265e-02, 7.82819003e-02, 7.10819919e-02,
       6.61273710e-02, 6.21878385e-02, 5.42665704e-02, 5.37969531e-02,
       5.01311101e-02, 4.64030261e-02, 3.99427933e-02, 3.83918809e-02,
       3.53348952e-02, 3.01372243e-02, 2.80488023e-02, 2.49852654e-02,
       2.17052887e-02, 2.16659686e-02, 1.81846648e-02, 1.60713319e-02,
       1.50625288e-02, 1.31196938e-02, 1.19387963e-02, 1.10997856e-02,
       9.73035916e-03, 7.82527366e-03])
```

1) Eigen Value의 값이 1 이상인 PC 선정

표준화된 변수들의 분산은 모두 1이며, 이 변수들의 총분산과 component의 총분산은 동일하다. 따라서 총 54개의 component는 평균 1의 분산값(즉, eigne value)을 가지게 되므로, eigen value값의 PC선정 기준을 1 이상으로 정할 수 있다. 위에서 구한 Eigen Value의 Matrix에서 1 이상인 Eigen Value는 총 4개가 나왔다.

```
array([4.01756867e+01, 2.16531589e+00, 1.41651436e+00, 1.19405368e+00,
       8.96249766e-01, 7.88897467e-01, 6.98636632e-01, 5.95362778e-01,
       5.68366461e-01, 5.29115848e-01, 4.28671399e-01, 3.90369411e-01,
       3.62469358e-01, 3.19198119e-01, 2.83031788e-01, 2.70341457e-01,
       2.52520084e-01, 2.20402276e-01, 2.02203198e-01, 1.91133304e-01,
       1.75596029e-01, 1.65126304e-01, 1.38913064e-01, 1.32199503e-01,
       1.19176457e-01, 1.07826952e-01, 1.03507589e-01, 1.00321611e-01,
       9.48779953e-02, 8.83932265e-02, 7.82819003e-02, 7.10819919e-02,
       6.61273710e-02, 6.21878385e-02, 5.42665704e-02, 5.37969531e-02,
       5.01311101e-02, 4.64030261e-02, 3.99427933e-02, 3.83918809e-02,
       3.53348952e-02, 3.01372243e-02, 2.80488023e-02, 2.49852654e-02,
       2.17052887e-02, 2.16659686e-02, 1.81846648e-02, 1.60713319e-02,
       1.50625288e-02, 1.31196938e-02, 1.19387963e-02, 1.10997856e-02,
       9.73035916e-03, 7.82527366e-03])
```



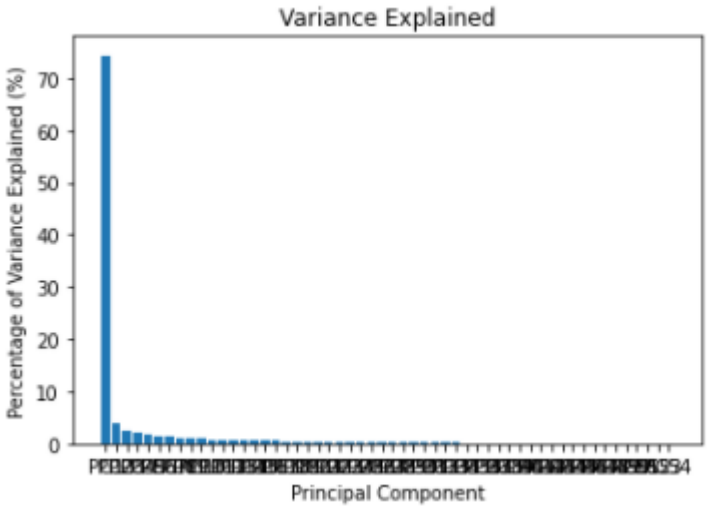


- 선정된 4개의 PC를 PC1, PC2, PC3, PC4라고 명명하였다. Scree plot을 보면 PC2로 가는 순간 eigen value가 급격히 줄어들고, PC4까지 확실한 하향곡선을 그리고 있다.

2) 누적된 설명가능 분산 비율을 통한 PC 선정

Explained_variance_ratio함수를 이용해 각 component의 기여도를 계산한 후, 누적 기여도를 확인해보았다.

0	0.743994	28	0.983076
1	0.784093	29	0.984713
2	0.810324	30	0.986162
3	0.832436	31	0.987479
4	0.849034	32	0.988703
5	0.863643	33	0.989855
6	0.876581	34	0.990860
7	0.887606	35	0.991856
8	0.898131	36	0.992784
9	0.907930	37	0.993644
10	0.915868	38	0.994383
11	0.923097	39	0.995094
12	0.929809	40	0.995749
13	0.935721	41	0.996307
14	0.940962	42	0.996826
15	0.945968	43	0.997289
16	0.950644	44	0.997691
17	0.954726	45	0.998092
18	0.958470	46	0.998429
19	0.962010	47	0.998726
20	0.965262	48	0.999005
21	0.968320	49	0.999248
22	0.970892	50	0.999469
23	0.973340	51	0.999675
24	0.975547	52	0.999855
25	0.977544	53	1.000000
26	0.979461		
27	0.981319		



이 경우 4번째 component까지 PC로 선정한다면 83.2436%, 대략 83%정도의 분산까지 설명가능하기 때문에 충분히 데이터의 분산을 설명할 수 있을 것이라 판단했다.

3) 두가지 모두 고려한 PC 선정

eigen value의 값을 1을 기준삼음과 동시에 누적기여도를 고려한다면, 역시 마찬가지로 4번째 PC까지 뽑아야 대략적으로 eigen value의 값도 만족시키며 83%정도의 누적 기여도를 가질 것이다.

	PC1	PC2	PC3	PC4
Q1	0.140194	0.137001	0.130086	0.134778
Q2	0.10317	0.116647	0.169542	0.148398
Q3	-0.0123	-0.16128	0.018737	-0.03578
Q4	-0.01637	-0.15502	-0.04252	-0.08397
Q5	0.032704	0.132646	0.158537	0.130858
Q6	-0.14606	0.062012	-0.11529	-0.09953
Q7	0.145958	0.002571	0.079947	-0.05071
Q8	0.226995	0.102937	0.259824	-0.00207
Q9	0.010881	-0.12561	-0.11574	-0.04045
Q10	0.053824	-0.10738	0.062881	0.459175
Q11	-0.12335	-0.00138	0.022367	0.0914
Q12	0.189929	0.105859	0.419573	0.085251
Q13	-0.11666	-0.15242	-0.03914	-0.16
Q14	-0.05867	-0.08291	0.084444	0.111153
Q15	0.341851	-0.07515	-0.15746	-0.01589
Q16	0.072164	0.160451	-0.35333	0.048852
Q17	0.084648	0.017566	0.079649	-0.18759
Q18	0.067391	0.166452	0.24162	0.026484
Q19	-0.30135	-0.12643	0.132763	-0.20317
Q20	0.008784	-0.17655	-0.18794	0.073044
Q21	0.029514	0.018597	0.09465	-0.02112
Q22	-0.015	-0.26425	-0.01611	0.058965
Q23	0.067435	-0.2905	-0.204	0.115525
Q24	0.022743	0.255501	-0.19607	-0.1278
Q25	-0.27068	0.210079	-0.17272	-0.10169
Q26	-0.1173	0.357719	-0.01753	0.008861
Q27	0.098035	0.18961	0.113833	-0.19131
Q28	-0.08922	-0.0514	0.112164	-0.09468
Q29	-0.29949	0.123311	0.036894	0.127728
Q30	-0.20548	0.077881	0.095123	0.050477
Q31	0.020336	0.180963	-0.23327	0.274417
Q32	-0.05993	-0.33773	0.114414	0.235173
Q33	-0.04721	0.056605	0.075004	0.299682
Q34	-0.28026	-0.02352	0.172574	0.185954
Q35	-0.1178	-0.14413	0.167711	-0.16239
Q36	-0.23271	0.093894	0.048662	-0.02978
Q37	-0.06103	-0.11785	0.102433	-0.14943
Q38	-0.25769	-0.01141	0.061546	0.07898
Q39	-0.05432	0.094823	-0.00489	0.106692
Q40	-0.03148	0.136499	-0.03777	-0.02084
Q41	0.056062	-0.03606	0.011471	-0.00568
Q42	0.020982	-0.01989	0.037236	0.106768
Q43	-0.11684	0.016304	0.054519	0.048606
Q44	-0.07393	0.053959	0.062116	-0.10372
Q45	0.038873	0.025485	-0.08927	0.103129
Q46	-0.0124	-0.0167	-0.07671	0.124168
Q47	-0.19954	0.067457	-0.08868	0.11474
Q48	0.106517	-0.07729	0.080747	-0.15071
Q49	0.069744	-0.00819	-0.01151	0.014533
Q50	0.08783	0.047346	-0.09902	0.073037
Q51	0.094816	0.110448	-0.11096	0.058192
Q52	-0.04788	-0.00361	-0.04156	0.161241
Q53	-0.09504	-0.03757	-0.05202	0.134227
Q54	0.007103	-0.06703	0.047021	0.012741

본 과제에서는 1번 기준을 바탕으로 PCA를 진행하였다.

2.3Principal Component 해석

PCA를 진행한 결과로 나온 각 PC의 독립변수들의 Linear Combination을 살펴보자. 각 PC에서 계수가 ±0.2 이상인 값들을 선택해 변수를 해석했다 각 질문의 내용은 하단에 첨부하였고, 질문들의 조합을 통해 각 PC가 의미하는 바를 찾고자 했다.

PC#	계수의 절대값이 0.2이상인 변수
PC1	Q8, Q15, Q19, Q25, Q29, Q30, Q34, Q36, Q38
PC2	Q22, Q23, Q24, Q25, Q26, Q32
PC3	Q8, Q12, Q16, Q18, Q23, Q31
PC4	Q10, Q19, Q31, Q32, Q33

1) Component 1

- **선형식:** $PC1 = Q1 * 0.14019 + Q2 * 0.10317 + \cdots + Q54 * 0.007103$

- 변수 중 Q8, Q15는 0.2보다 큰 값을 가졌고, 나머지는 -0.2보다 작은 값을 가졌다. Q8과 Q15는 배우자와 많은 시간을 보내는지, 배우자와 서로 비슷하고 조화로운 목표를 추구하는지에 관한 질문이었다.

- 높은 음의 값을 가진 Q19, Q25, Q29, Q30, Q34, Q36, Q38 중 19,25,29번 문항은 응답자 입장에서 배우자를 잘 아는지에 대한 내용이고, 30번대는 배우자에게 논쟁 중 공격적인 표현을 사용할 수 있는 가에 대한 내용이다.

- 즉, PC1의 값이 크다면 시간을 많이 보내고, 서로 비슷한 목표를 추구하지만, 동시에 공격적인 표현을 아끼고 배우자에 대해 잘 안다고 생각을 덜 한다고 판단할 수 있다.

➢ 이를 통해 **PC1은 배우자에 대한 존중을 의미하는 지표**라 추정할 수 있다.

2) Component 2

- **선형식:** $PC2 = Q1 * 0.137001 + Q2 * 0.116647 + \cdots + Q54 * 0.06703$

- Q24, Q25, Q26은 0.2보다 큰 값을 가졌고, Q22, Q23, Q32는 -0.2보다 작은 값을 가졌다. 양의 값을 갖는 Q24, Q25, Q26는 배우자가 받는 스트레스를 알고 있는가, 배우자의 내면을 알고 있는가, 배우자가 어떤 것에 불안해하는지 알고 있는가를 물어보는 질문이었다.

- 음의 값을 갖는 Q22, Q23, Q32는 배우자가 아플 때 어떻게 보살펴줬으면 하는지, 배우자가 가장 좋아하는 음식이 무엇인지 아는지, 배우자와 논쟁 시 배우자를 다 아는 것처럼 표현하는지에 대한 내용이다.

- 양의 값을 갖는 변수들은 배우자의 내면을 잘 아는지에 대한 내용으로 분류가 가능하지만, 음의 값을 갖는 변수들은 서로 다른 의미로 분류될 수가 있다. 하지만 32과 같은 부정적인 내용도 있어 해석한다면, PC2의 값이 크다면 배우자의 내면을 잘 살피며 부정적인 표현을 대화 중에 사용하지 않는다고 말할 수 있으며 그 값이 작다면 배우자의 내면에 대해 잘 알지 못하고 있다고 말할 수 있다.

➢ 이를 통해 **PC2는 배우자의 내면을 얼마나 잘 살피는지에 관한 지표**라고 추정할 수 있다.

3) Component 3

- **선형식:** $PC3 = Q1 * 0.130086 + Q2 * 0.165 + \cdots + Q54 * 0.047022$

- Q8, Q12, Q18는 0.2보다 큰 값을 가졌고, Q16, Q23, Q31은 -0.2보다 작은 값을 가졌다. Q8, Q12, Q18은 배우자와 휴가시간을 많이 보내는지, 서로의 결혼관이 유사한지, 개인시간에 가지는 가치가 비슷한지에 관한 질문들이었다.

- Q16, Q23, Q31은 배우자와 사랑에 대해 다른 가치관을 가지는지, 배우자가 좋아하는 음식을 아는지, 배우자와 논쟁을 할 때 공격적이 되곤 하는지에 관한 질문이었다. 23번의 좋아하는 음식을 아는지에 대한 변수는 애매하지만 전반적으로 배우자에 대한 부정적인 내용이었다.

- 즉, PC3의 경우 애매한 변수가 존재하지만 '가치관'에 대한 뚜렷한 질문이 분류되었고 해당하는 변수만의 절대값이 0.3이상이기 때문에 값이 크다면 배우자와 결혼, 사랑, 개인 시간에 대해 비슷한 가치관을 가진다고 할 수 있다. 반대로 그 값이 작다면 서로 가치관이 다르며, 대화할 때도 공격적이 된다고 할 수 있다.

➢ 이를 통해 **PC3는 배우자와 가치관이 얼마나 일치하는지에 관한 지표**라고 추정할 수 있다.

4) Component 4

- **선형식:** $PC4 = Q1 * 0.134778 + Q2 * 0.116647 + \cdots + Q54 * 0.012741$

- Q10, Q31, Q32, Q33은 0.2보다 큰 값을 가졌고, Q19는 -0.2보다 작은 값을 가졌다. Q10, Q31, Q32, Q33는 대체적으로 대화 중 공격적이거나 부정적인 표현을 사용하는지, 상대를 비하하는지에 관한 질문이었다.

- Q19는 결혼에서 서로의 역할에 대한 가치관이 같는지를 묻는 질문이었다.

- 즉, PC4의 값이 크다면 의사소통 중에 쉽게 갈등을 빚을 수 있다는 것을 의미하며, 값이 작다면 서로를 존중하며 의사소통을 한다고 판단할 수 있다.

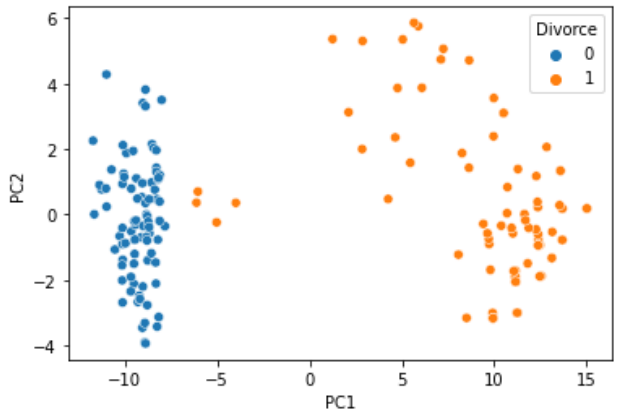
➢ 이를 통해 **PC4는 배우자와의 논쟁 중 공격성을 나타낼 수 있는 지에 대한 지표**라고 추정할 수 있다.

2.4 Score Plot

Scatterplot 함수를 통해 각 PC들을 서로 비교해보았다.

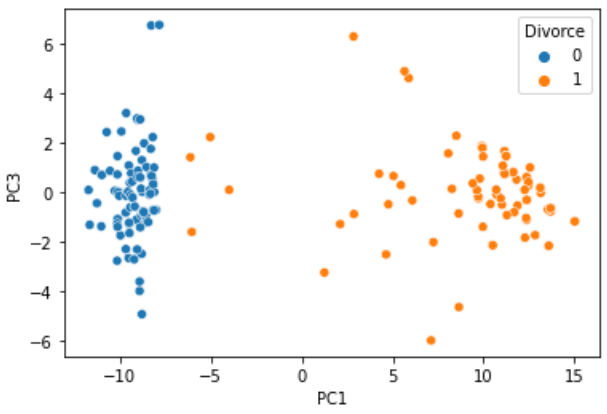
1) PC1과 PC2 비교

```
In [9]: # Principal Component Pattern Plot
        sns.scatterplot(data=X_pp, x='PC1',y='PC2',hue=X_pp.index)
Out[9]: <AxesSubplot:xlabel='PC1', ylabel='PC2'>
```



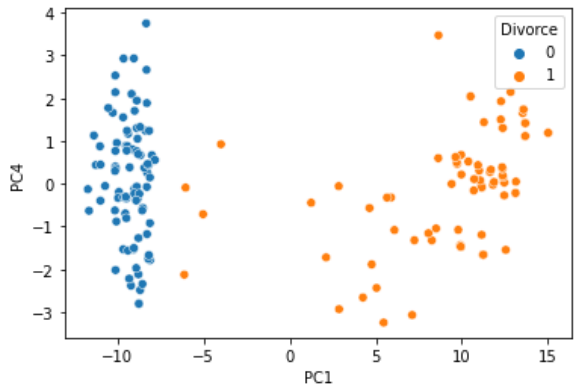
2) PC1과 PC3 비교

```
In [10]: # Principal Component Pattern Plot
         sns.scatterplot(data=X_pp, x='PC1',y='PC3',hue=X_pp.index)
Out[10]: <AxesSubplot:xlabel='PC1', ylabel='PC3'>
```



3) PC1과 PC4 비교

```
In [11]: # Principal Component Pattern Plot
         sns.scatterplot(data=X_pp, x='PC1',y='PC4',hue=X_pp.index)
Out[11]: <AxesSubplot:xlabel='PC1', ylabel='PC4'>
```




```
X_PC1.groupby('Divorce').describe()
```

	PC1							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-5.932357	0.596637	-7.711419	-6.239674	-5.823990	-5.537535	-5.079927
1	84.0	6.073603	2.850440	-4.114917	5.551229	6.997751	7.875186	9.597984

```
X_PC2.groupby('Divorce').describe()
```

	PC2							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.175424	1.236034	-2.892542	-0.855166	-0.172642	0.600760	2.738632
1	84.0	0.179601	1.675070	-2.017739	-1.017347	-0.333487	0.968475	4.822913

```
X_PC3.groupby('Divorce').describe()
```

	PC3							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.061159	1.101251	-3.187788	-0.826102	0.016449	0.428421	3.870448
1	84.0	0.062615	1.285097	-3.273100	-0.794973	0.129005	0.631267	3.810391

```
X_PC4.groupby('Divorce').describe()
```

	PC4							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.135289	0.955748	-2.440106	-0.670327	-0.162961	0.372947	2.82740
1	84.0	0.138510	1.213180	-1.450103	-0.717238	-0.064942	0.749446	3.46081

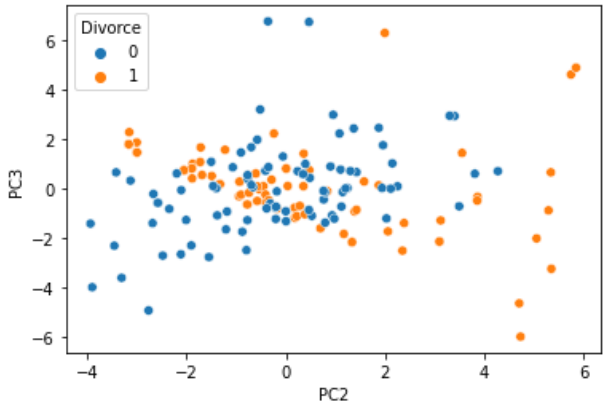
- 먼저, PC1이 총 분산에 대해 75%에 가까운 설명력을 보이고, Score Plot상에서 각 Divorce그룹을 명확히 나누기에 이것을 x축으로 갖고 나머지 성분을 y축으로 갖는 Score plot 세가지를 묶어 분석하였다. 이 경우, Plot의 개형이 비슷하여 두 성분간의 상관관계를 동일하게 분석할 수 있다. PC1은 음, 양 극단으로 관측치가 분포되어있는 것이 공통된 형태이다. 동시에 나머지 3개의 성분은 PC1의 값에 관계 없이 고르게 분포되어 있었다.
- 이를 통해서 PC1~PC2, PC1~PC3, PC1~PC4에는 명확한 양이나 음의 상관관계가 존재하지않는다는것을 볼수있다.

- 또한 PC1의 관점에서 데이터를 살펴보자면, Divorce를 하지않은 그룹(Divorce=0)은 PC1이 0보다 작게 나오는 경향이 있고, Divorce를 한 그룹(Divorce=1)은 PC1이 0보다 크게 나오는 경향이 있다.

이는 배우자와 보내는 시간, 의사소통 중 서로를 배려하는정도와 관련해서 표면적으로 드러나는 관계가 이혼과 상관이 있다는 것을 알수있다. 하지만 그 계수 부호와 관련되어서 의미적으로 반대로 설명되는 부분이 있기에 추후 FA를 통해 추가적인 분석이 필요할것으로 보인다.

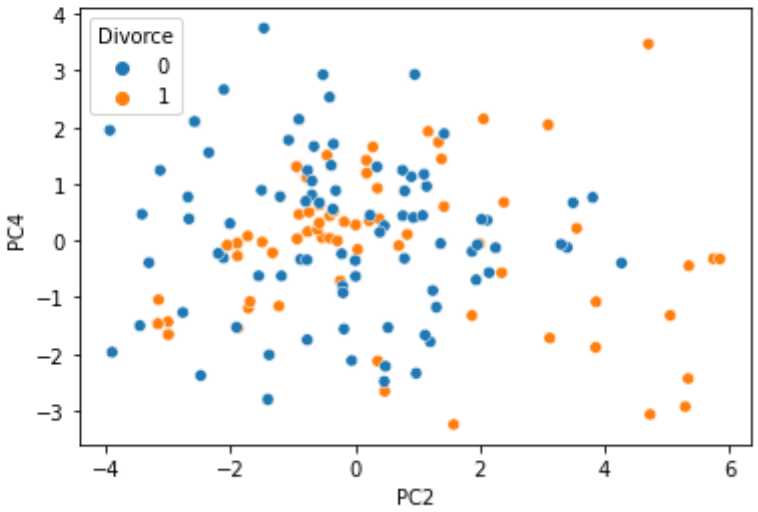
4) PC2와 PC3 비교

```
In [12]: # Principal Component Pattern Plot
         sns.scatterplot(data=X_pp, x='PC2',y='PC3',hue=X_pp.index)
Out[12]: <AxesSubplot:xlabel='PC2', ylabel='PC3'>
```



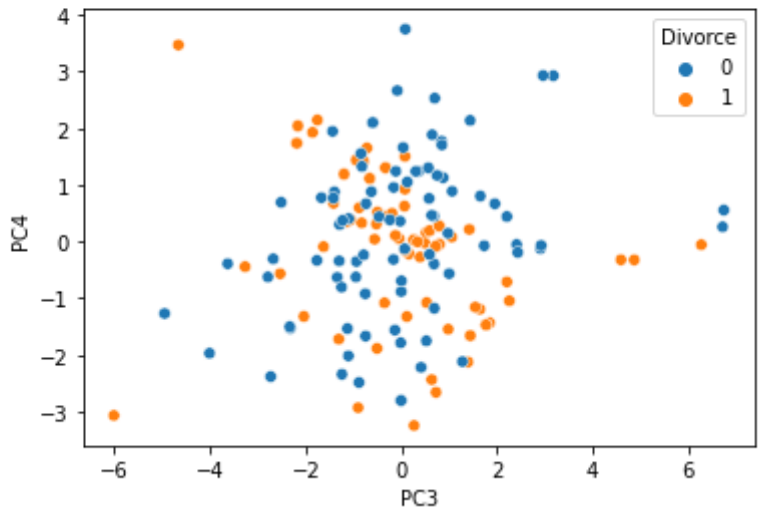
5) PC2와 PC4 비교

```
In [13]: # Principal Component Pattern Plot
         sns.scatterplot(data=X_pp, x='PC2',y='PC4',hue=X_pp.index)
Out[13]: <AxesSubplot:xlabel='PC2', ylabel='PC4'>
```



6) PC3과 PC4 비교

```
In [14]: # Principal Component Pattern Plot
         sns.scatterplot(data=X_pp, x='PC3',y='PC4',hue=X_pp.index)
Out[14]: <AxesSubplot:xlabel='PC3', ylabel='PC4'>
```




```
X_PC2.groupby('Divorce').describe()
```

	PC2							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.175424	1.236034	-2.892542	-0.855166	-0.172642	0.600760	2.738632
1	84.0	0.179601	1.675070	-2.017739	-1.017347	-0.333487	0.968475	4.822913

```
X_PC3.groupby('Divorce').describe()
```

	PC3							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.061159	1.101251	-3.187788	-0.826102	0.016449	0.428421	3.870448
1	84.0	0.062615	1.285097	-3.273100	-0.794973	0.129005	0.631267	3.810391

```
X_PC4.groupby('Divorce').describe()
```

	PC4							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.135289	0.955748	-2.440106	-0.670327	-0.162961	0.372947	2.82740
1	84.0	0.138510	1.213180	-1.450103	-0.717238	-0.064942	0.749446	3.46081

- 나머지 3경우의 Score plot을 살펴보자면, 모든 데이터들이 PC2, PC3, PC4의 관점에서 고르게 분포하거나, 약하게 음의 경향, 양의 경향을 보이고있기 때문에 PC2~PC3, PC2~PC4, PC3~PC4간에 상관관계는 0에 가까운것으로 추정된다.

- 또 PC2, PC3, PC4기준에서 각 Divorce그룹에 대해 약한 음의 상관관계, 또는 약한 양의 상관관계를 보이는데 이는 배우자의 내면을 살피는 정도, 배우자와 가치관이 일치하는 정도, 의사소통 중 상호존중에 관한 지표와 Divorce여부가 관련이 있다고 설명할수있다. 이 또한 계수 부호와 관련돼서 반대적으로 설명되는부분이 있기에 FA를 통해 보충해보려한다.

2.5 PCA 결론

해당 과정을 진행하기 전에, 표준화되지 않은 데이터를 통해 PCA를 한차례 진행했었다. 데이터의 공분산 행렬을 통해 위와 같은 방식으로 PCA를 도출한 결과, 데이터를 표준화하지 않았을 경우와 표준화를 했을 때의 총분산의 차이가 매우 적게 나타났다. 각 component의 eigen value와 누적 설명 가능 분산 비율 역시 매우 미미한 차이를 보였다. 그 이유는 아마도 설문조사의 특징상 0~4로 답변이 한정되기 때문에 평균과 분산의 차이는 있어도 심각한 크기의 차이가 존재하지 않기 때문이라고 추론할 수 있었다. 해당 과제에서는 그 차이가 무시할 수 있을 만큼 작다고 판단해 데이터를 표준화한 후 PCA를 진행해보았다.

이때 나타난 4가지로 나눈 주성분 값에 특성을 대략적으로 붙여보았으나, 확실하게 구분되는 공통점이 드러나지는 않았다. 그러한 이유로 Factor Analysis를 통해 한 번 더 데이터를 분석해보기로 결정했다.

3. Factor Analysis

요인 분석은 관찰된 변수 집합에서 영향력 있는 기본 요인 또는 잠재 변수를 검색하는데 사용되는 탐색적 데이터 분석 방법이다. 변수 수를 줄여 데이터 해석에 도움이 되게 하고, 모든 변수에서 최대 공분산을 추출하여 공통 점수에 넣는다. 요인 분석은 시장, 광고 등 분야에도 많이 사용되는데, 시장조사 시 요인 분석을 사용하여 가격에 민감한 고객을 식별하고 소비자 선택에 영향을 미치는 브랜드 기능을 식별하여 유통채널에 대한 채널 선택 기준을 이해하는데 도움을 준다.

이때 요인(Factor)란 관측된 변수 수 간의 연관성을 설명하는 잠재변수이다. 모든 요인은 관찰된 변수의 특정 분산을 설명한다.

3.1. Bartlett과 KMO 검정

- 요인 분석을 진행하기 전 두 검정을 통하여 해당 데이터가 요인 분석에 적합한지 판단하였다. 먼저 Bartlett 검정을 진행하였는데, 이 검정은 요인 분석 모형의 적합성 여부를 나타낸다. 적합 여부는 유의확률로서 파악을 하는데, 데이터의 상관계수 행렬의 행렬식 값을 계산하여 상관계수 행렬이 단위행렬인지 아닌지 카이제곱 분포를 이용해서 검정하는 방법이다. 이때 'H0: 상관계수 행렬은 단위행렬이다' 라는 귀무가설을 설정한 이후, 귀무가설이 기각되어야 요인분석 모형으로 적합하다. 파이썬 코드를 통해 Scaling한 데이터를 이용하여 검정을 진행하였다. 이때 통계량 값은 17654.27이 도출되었고, p-value는 0으로 0.05보다 작으므로 귀무가설을 기각할 수 있었으므로 요인 분석에 적합하다는 결론을 내렸다.

```
# Bartlett Test
## H0: 상관관계 행렬이 단위행렬이다
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericit
chi_square_value, p_value = calculate_bartlett_sphericity(data_scale)
chi_square_value, p_value # p-value < 0.05 --> 귀무가설 기각
```

```
(17654.270924632456, 0.0)
```

- Kaiser-Meyer-Olkin(KMO)는 변수들 간의 상관관계가 다른 변수에 의해 잘 설명되는 정도를 나타내는 값이다. 즉, 이 값이 적으면 요인분석을 위한 변수들의 선정이 적절치 않았다는 것을 의미한다. (0.90이상이면 상당히 좋은 편/0.80-0.89이면 꽤 좋은 편/0.7-0.79이면 적당한 편/0.6-0.69는 평범/0.50-0.59는 바람직하지 못한 편/0.5 미만이면 받아들일 수없는 수치) Python 코드를 통해 진행한 결과, KMO값이 0.9640이 나와 변수들의 선정이 적절했다는 결론을 도출하였다.

```
# KMO Test (Kaiser-Meyer-Olkin Test)

from factor_analyzer.factor_analyzer import calculate_kmo
kmo_all, kmo_model =calculate_kmo(data_scale)
kmo_model
```

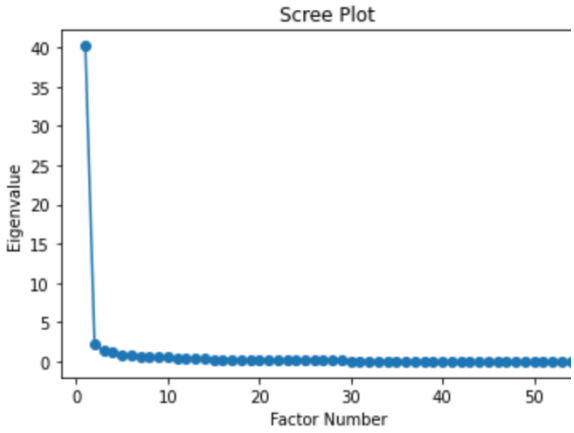
```
/Users/leejeonghyun/opt/anaconda3/lib/python3.8/site-packages/fact
alyzer/utils.py:248: UserWarning: The inverse of the variance-cova
e matrix was calculated using the Moore-Penrose generalized matri
rsion, due to its determinant being at or very close to zero.
  warnings.warn('The inverse of the variance-covariance matrix '
```

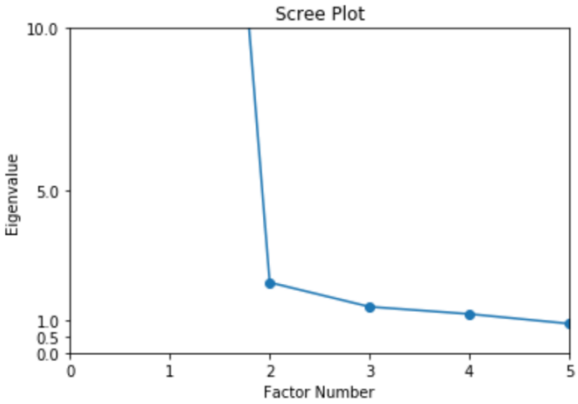
```
0.9639648508490816
```

-

3.2. 요인 수 선택

- 주성분 분석(PCA)에서 진행한 동일한 방법으로 Eigenvalue가 1 이상인 변수의 개수를 통해 진행하였다. 아래 그래프는 시각화에 용이하기 위해 일부만 표시했으며, 그래프에서 볼 수 있듯이 4개로 결론이 나와 요인 수도 4개로 하여 진행하였다.

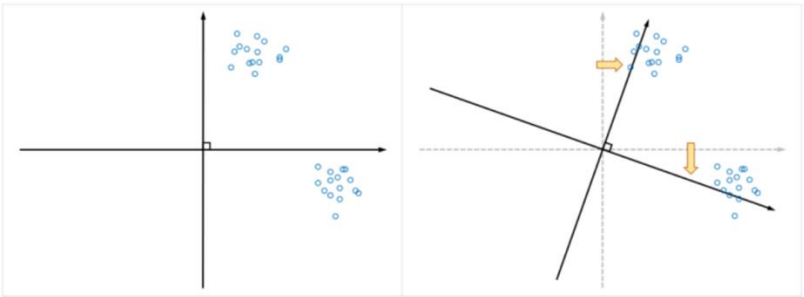




3.3. 요인 회전

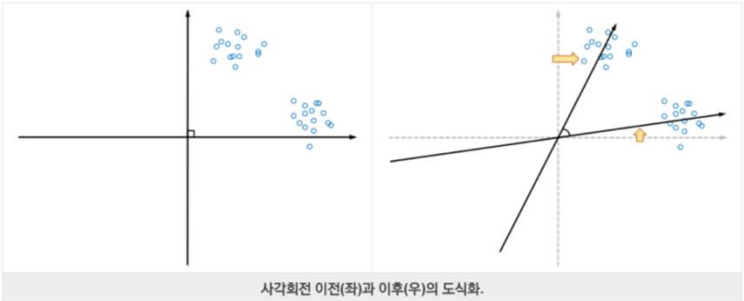
- 행렬의 각 성분들은 좌표축에 최대한 근접할수록 내용적 해석이 간편해지므로, 요인을 회전시켜 요인행렬을 좌표계 위에서 새롭게 생각한다. 이때, 내용적 해석이 간편한 것은, 성분의 값이 1.0에 가깝게 확실하게 크거나, 0에 가깝게 확실하게 작다는 것을 의미한다. 요인분석에는 직교 회전(Orthogonal Rotation)과 사각 회전(Oblique Rotation)을 이용하였고, 각각 2가지 방식을 통해 총 4가지 회전방법을 이용하여 요인회전을 했다. 이후 Factor들의 Cumulative Variance가 높은 회전방식이 가장 적합한 회전이라고 판단하고 이후 분석을 진행하였다.

- *(직교회전)* 직교회전은 각각의 좌표축 간의 각도를 직각으로 유지하여 회전한다. 요인 간의 상관계수 값을 cos90, 즉 0으로 가정한다. 이때 요인 간 상관이 없는 모형에 사용하는 것이 아닌, 요인 간 상관이 없다고 '가정하는' 모형을 원할 때 사용한다.



직각회전 이전(좌)과 이후(우)의 도식화.

- *(사각회전)* 사각회전은 각각의 좌표축 간의 직각을 인정하지 않으면서 회전시키는 방법이다. 요인이 서로 완벽한 독립이 이뤄질 수 없으며, 공분산이 존재할 수 있다. 사각회전은 먼저 직각회전을 한 번 실시한 이후 진행되며, 이때 각 요인별로 가장 대표성이 있는 지표변인들을 골라, 이 변인들끼리의 상관계수로 추가 초정을 한다. 하지만 대표적인 지표변인을 선정하는 과정이 주관성이 크고, 사각회전 자체의 통계적 처리와 해석에 있어서 까다롭다는 의견이 존재한다.



사각회전 이전(좌)과 이후(우)의 도식화.

1) Varimax Rotation – 직교회전

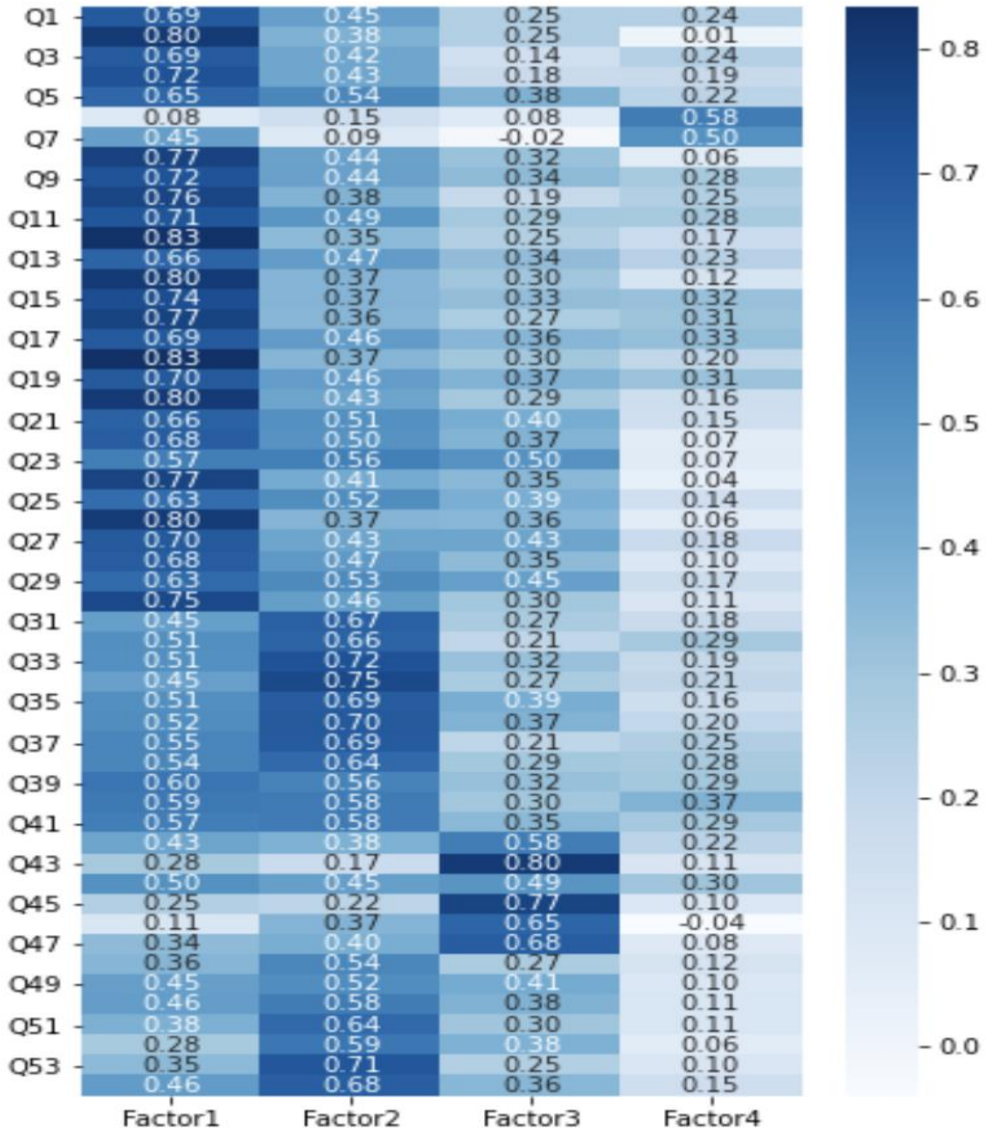
- 가장 대중적인 기준이다. 요인의 분산을 극대화하는 논리를 따르는데, 요인행렬을 변환할 때 행렬의 열인 요인을 기준으로 하여 큰 값은 더 크게, 작은 값은 더 작게 회전한다. 다요인 구조 속의 모든 요인들의 의미가 뚜렷하게 해석될 수 있다는 장점이 있다.

- 요인 수 선택을 통하여 4개의 요인 수를 설정하였고, varimax로 rotation을 설정하였다.

	Factor1	Factor2	Factor3	Factor4
Q1	0.693062	0.451776	0.253642	0.241222
Q2	0.798414	0.381013	0.249207	0.010322
Q3	0.687292	0.421764	0.141853	0.241013
Q4	0.721569	0.427477	0.181499	0.185077
Q5	0.649439	0.537233	0.378114	0.221224

(loading을 앞 5개만 불러옴.)

- Varimax Rotation을 이용하여 위 표와 같이 각 변수와 요인에 해당되는 Loading 값을 도출할 수 있었다. Loading 값의 절대값을 아래와 같이 히트맵을 이용하여 시각화를 진행하였다.



	Factor1	Factor2	Factor3	Factor4
SS Loadings	19.798426	13.708014	7.657385	2.687099
Proportion Var	0.366638	0.253852	0.141803	0.049761
Cumulative Var	0.366638	0.620490	0.762293	0.812054

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)*

- 다음은 Varimax Rotation을 이용한 요인 분석 결과이다. 총 4개의 요인으로 전체 분산의 약 81%정도의 설명력을 갖는다고 볼 수 있었다.

2) Quartimax Rotation – 직교회전

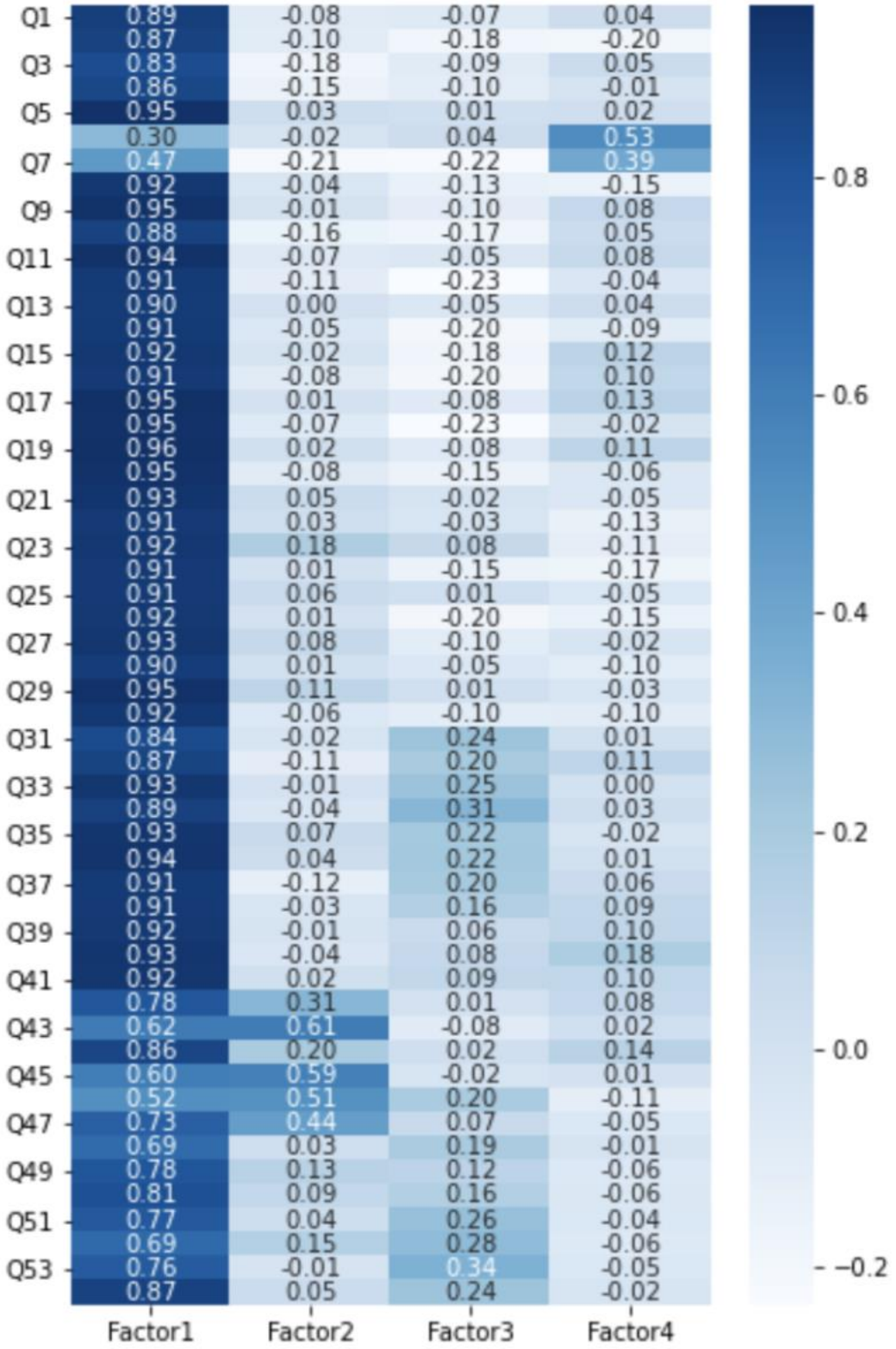
- 계산과정에서 4제곱이 활용되기 때문에 Quarti-가 붙는 방법이다. Varimax가 행렬의 열을 기준으로 한다면, Quartimax는 행렬의 행을 기준으로 분산을 극대화한다. 연구자들은 Quartimax가 제1요인(Factor1)만 과대해석하고 기타요인은 과소해석하는 문제가 있다고 비판하는 바가 있다. 따라서 Quartimax는 단일요인 구조가 존재한다는 확실한 상황이나 한정적으로 사용할 수 있어 범용성이 낮다. 본 데이터는 다요인 구조로 보이지만, 해당 rotation을 적용해보기로 하였다.

- 요인 수 선택을 통하여 4개의 요인 수를 설정하였고, Quartimax로 rotation을 설정하였다.

	Factor1	Factor2	Factor3	Factor4
Q1	0.890090	-0.084595	-0.074820	0.044075
Q2	0.874045	-0.095624	-0.183361	-0.195258
Q3	0.827784	-0.181084	-0.088879	0.050468
Q4	0.858001	-0.152859	-0.104438	-0.012432
Q5	0.949026	0.031229	0.013153	0.022371

(loading을 앞 5개만 불러옴.)

- Quartimax Rotation을 이용하여 위 표와 같이 각 변수와 요인에 해당되는 Loading 값을 도출할 수 있었다. Loading 값의 절대값을 아래와 같이 히트맵을 이용하여 시각화를 진행하였다. 특이한 점은 Factor1의 Loading 절대값이 상대적으로 높다는 것이었다.


Q1
Q3
Q5
Q7
Q9
Q11
Q13
Q15
Q17
Q19
Q21
Q23
Q25
Q27
Q29
Q31
Q33
Q35
Q37
Q39
Q41
Q43
Q45
Q47
Q49
Q51
Q53
0.89 -0.08 -0.07 0.04
0.87 -0.10 -0.18 -0.20
0.83 -0.18 -0.09 0.05
0.86 -0.15 -0.10 -0.01
0.95 0.03 0.01 0.02
0.30 -0.02 0.04 0.53
0.47 -0.21 -0.22 0.39
0.92 -0.04 -0.13 -0.15
0.95 -0.01 -0.10 0.08
0.88 -0.16 -0.17 0.05
0.94 -0.07 -0.05 0.08
0.91 -0.11 -0.23 -0.04
0.90 0.00 -0.05 0.04
0.91 -0.05 -0.20 -0.09
0.92 -0.02 -0.18 0.12
0.91 -0.08 -0.20 0.10
0.95 0.01 -0.08 0.13
0.95 -0.07 -0.23 -0.02
0.96 0.02 -0.08 0.11
0.95 -0.08 -0.15 -0.06
0.93 0.05 -0.02 -0.05
0.91 0.03 -0.03 -0.13
0.92 0.18 0.08 -0.11
0.91 0.01 -0.15 -0.17
0.91 0.06 0.01 -0.05
0.92 0.01 -0.20 -0.15
0.93 0.08 -0.10 -0.02
0.90 0.01 -0.05 -0.10
0.95 0.11 0.01 -0.03
0.92 -0.06 -0.10 -0.10
0.84 -0.02 0.24 0.01
0.87 -0.11 0.20 0.11
0.93 -0.01 0.25 0.00
0.89 -0.04 0.31 0.03
0.93 0.07 0.22 -0.02
0.94 0.04 0.22 0.01
0.91 -0.12 0.20 0.06
0.91 -0.03 0.16 0.09
0.92 -0.01 0.06 0.10
0.93 -0.04 0.08 0.18
0.92 0.02 0.09 0.10
0.78 0.31 0.01 0.08
0.62 0.61 -0.08 0.02
0.86 0.20 0.02 0.14
0.60 0.59 -0.02 0.01
0.52 0.51 0.20 -0.11
0.73 0.44 0.07 -0.05
0.69 0.03 0.19 -0.01
0.78 0.13 0.12 -0.06
0.81 0.09 0.16 -0.06
0.77 0.04 0.26 -0.04
0.69 0.15 0.28 -0.06
0.76 -0.01 0.34 -0.05
0.87 0.05 0.24 -0.02
Factor1
Factor2
Factor3
Factor4
0.8
0.6
0.4
0.2
0.0
−0.2

	Factor1	Factor2	Factor3	Factor4
SS Loadings	39.969521	1.650155	1.410461	0.820787
Proportion Var	0.740176	0.030558	0.026120	0.015200
Cumulative Var	0.740176	0.770735	0.796854	0.812054

- Varimax와 마찬가지로 총 분산의 약 81%정도의 설명력을 갖췄다고 볼 수 있다. 하지만, Factor1의 분산비율이 74%나 되며, 이는 앞서 언급한 Quartimax의 제1요인에 지나치게 해석이 된다는 단점과 부합하였다.

3) Promax Rotation – 사각회전

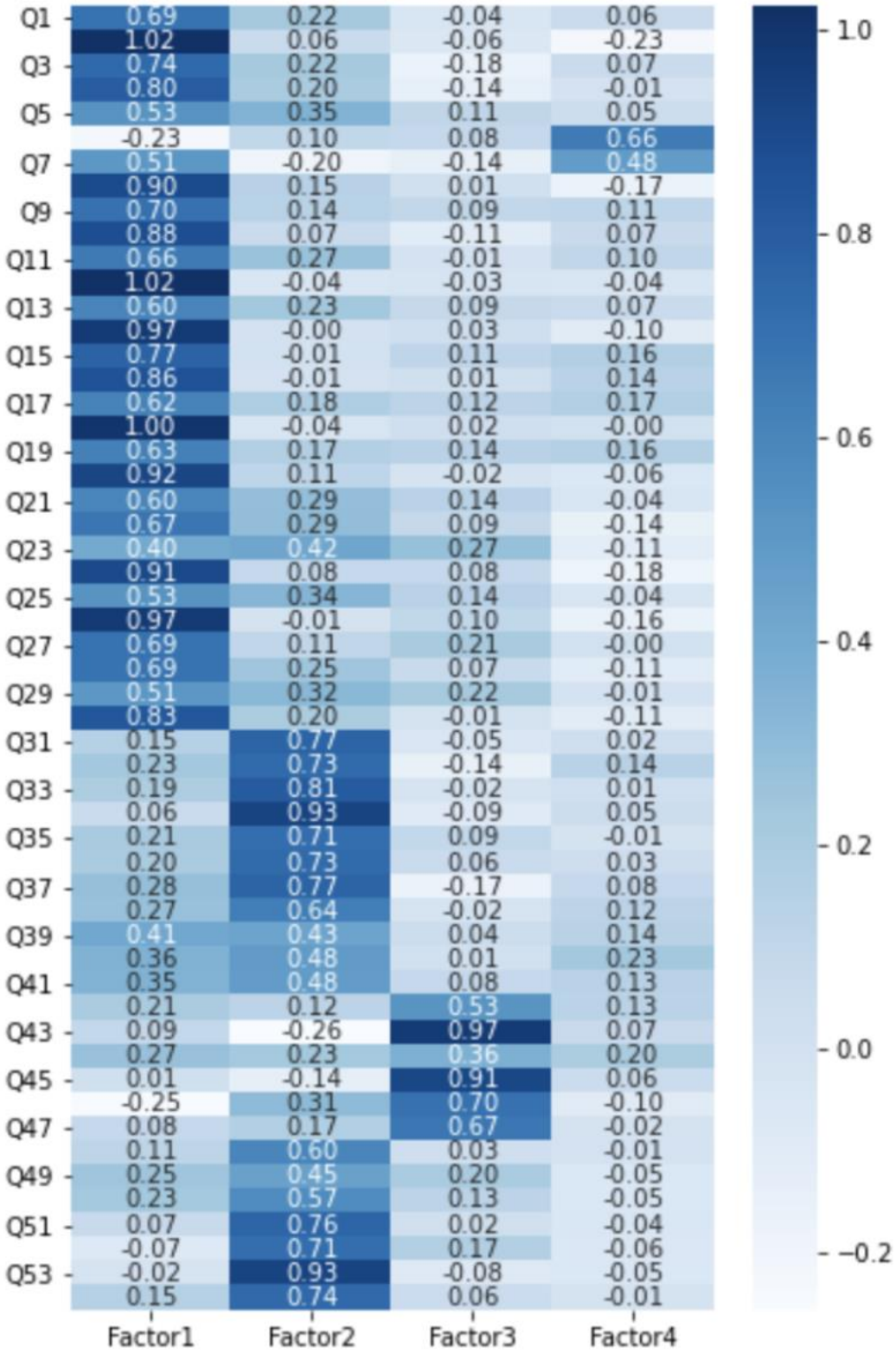
- Promax는 Varimax Rotation을 기반으로 축의 직교가 유지되지 않은 채 진행되는 사각회전의 한 방법이다. 따라서 Promax는 요인들 사이에 낮은 상관성을 갖도록 한다.

- 요인 수 선택을 통하여 4개의 요인 수를 설정하였고, Promax로 rotation을 설정하였다.

	Factor1	Factor2	Factor3	Factor4
Q1	0.691863	0.222440	-0.035146	0.064734
Q2	1.019329	0.063642	-0.064854	-0.228469
Q3	0.735160	0.223095	-0.175711	0.065371
Q4	0.801662	0.201815	-0.139443	-0.009277
Q5	0.529566	0.351665	0.108003	0.045121

(loading을 앞 5개만 불러옴.)

- 앞선 두 Rotation와 마찬가지로, 위 표와 같이 각 변수와 요인에 해당되는 Loading 값을 도출할 수 있었다. Loading 값의 절대값을 아래와 같이 히트맵을 이용하여 시각화를 진행하였다.


Q1 0.69 0.22 -0.04 0.06
1.02 0.06 -0.06 -0.23
Q3 0.74 0.22 -0.18 0.07
0.80 0.20 -0.14 -0.01
Q5 0.53 0.35 0.11 0.05
-0.23 0.10 0.08 0.66
Q7 0.51 -0.20 -0.14 0.48
0.90 0.15 0.01 -0.17
Q9 0.70 0.14 0.09 0.11
0.88 0.07 -0.11 0.07
Q11 0.66 0.27 -0.01 0.10
1.02 -0.04 -0.03 -0.04
Q13 0.60 0.23 0.09 0.07
0.97 -0.00 0.03 -0.10
Q15 0.77 -0.01 0.11 0.16
0.86 -0.01 0.01 0.14
Q17 0.62 0.18 0.12 0.17
1.00 -0.04 0.02 -0.00
Q19 0.63 0.17 0.14 0.16
0.92 0.11 -0.02 -0.06
Q21 0.60 0.29 0.14 -0.04
0.67 0.29 0.09 -0.14
Q23 0.40 0.42 0.27 -0.11
0.91 0.08 0.08 -0.18
Q25 0.53 0.34 0.14 -0.04
0.97 -0.01 0.10 -0.16
Q27 0.69 0.11 0.21 -0.00
0.69 0.25 0.07 -0.11
Q29 0.51 0.32 0.22 -0.01
0.83 0.20 -0.01 -0.11
Q31 0.15 0.77 -0.05 0.02
0.23 0.73 -0.14 0.14
Q33 0.19 0.81 -0.02 0.01
0.06 0.93 -0.09 0.05
Q35 0.21 0.71 0.09 -0.01
0.20 0.73 0.06 0.03
Q37 0.28 0.77 -0.17 0.08
0.27 0.64 -0.02 0.12
Q39 0.41 0.43 0.04 0.14
0.36 0.48 0.01 0.23
Q41 0.35 0.48 0.08 0.13
0.21 0.12 0.53 0.13
Q43 0.09 -0.26 0.97 0.07
0.27 0.23 0.36 0.20
Q45 0.01 -0.14 0.91 0.06
-0.25 0.31 0.70 -0.10
Q47 0.08 0.17 0.67 -0.02
0.11 0.60 0.03 -0.01
Q49 0.25 0.45 0.20 -0.05
0.23 0.57 0.13 -0.05
Q51 0.07 0.76 0.02 -0.04
-0.07 0.71 0.17 -0.06
Q53 -0.02 0.93 -0.08 -0.05
0.15 0.74 0.06 -0.01
Factor1
Factor2
Factor3
Factor4
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

	Factor1	Factor2	Factor3	Factor4
SS Loadings	18.125711	10.236781	3.706271	1.240803
Proportion Var	0.335661	0.189570	0.068635	0.022978
Cumulative Var	0.335661	0.525231	0.593866	0.616844

Factor #	각 요인의 Loading 절대값이 0.6이상인 변수
Factor 1	Q1, Q2, Q3, Q4, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20, Q21, Q22, Q24, Q26, Q27, Q28, Q30
Factor 2	Q31, Q32, Q33, Q34, Q35, Q36, Q37, Q38, Q48, Q51, Q53, Q54
Factor 3	Q43, Q45, Q46, Q47
Factor 4	Q6, Q7 *(Factor4의 경우 절대값 제일 높은 두가지를 선택함)*

- Promax Rotation을 통해 진행한 결과이다. Promax Rotation은 전체 분산의 약 62%의 설명력을 갖는 것으로 파악되었다. 흥미로웠던 점은 Factor별 Loading의 절대값이 높은 변수를 분류하였을 때, Varimax와 거의 같은 변수들이 분류되었다는 점이다.

4) Oblimin Rotation – 사각회전

- 또다른 사각회전의 방식 중 하나인 Oblimin rotation을 이용하여 진행하였다**.** Oblimin Rotation은 요인의 상관성을 인정하는 rotation의 일종이고, 상관성이 없을 경우에는 직교회전으로 진행된다. 이 회전은 높은 고유계수를 도출하지만, 요인의 해석력을 감소시킨다.

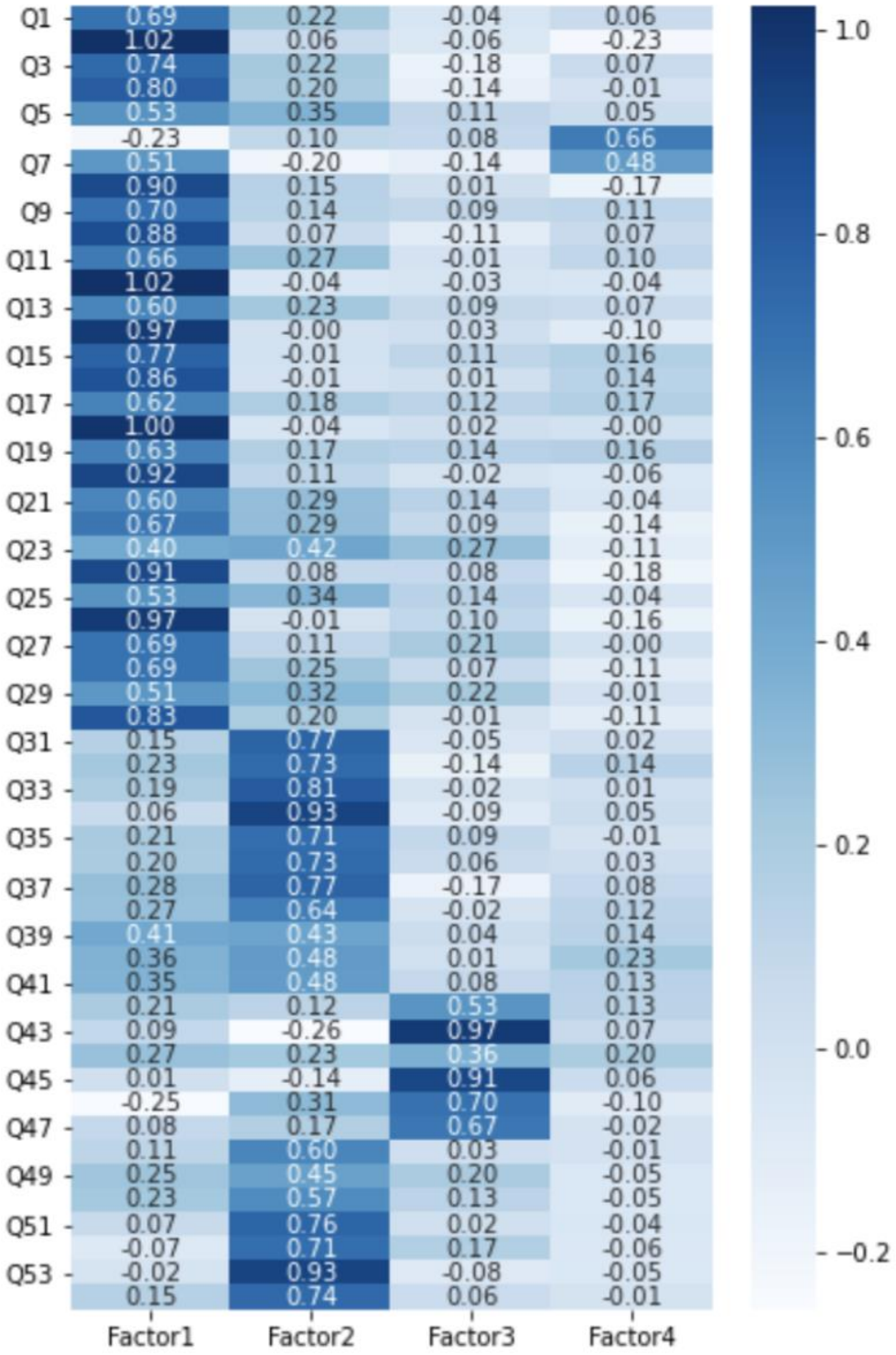
- 요인 수 선택을 통하여 4개의 요인 수를 설정하였고, Oblimin으로 rotation을 설정하였다.

	Factor1	Factor2	Factor3	Factor4
Q1	0.691863	0.222440	-0.035146	0.064734
Q2	1.019329	0.063642	-0.064854	-0.228469
Q3	0.735160	0.223095	-0.175711	0.065371
Q4	0.801662	0.201815	-0.139443	-0.009277
Q5	0.529566	0.351665	0.108003	0.045121

(loading을 앞 5개만 불러옴.)

- 앞선 Rotation와 마찬가지로, 위 표와 같이 각 변수와 요인에 해당되는 Loading 값을

도출할 수 있었다. 흥미로운 점은 Oblimin Rotation의 Loading값과 Promax의 Loading 값이 같았다. Loading 값의 절대값을 아래와 같이 히트맵을 이용하여 시각화를 진행하였다.



	Factor1	Factor2	Factor3	Factor4
SS Loadings	16.449169	10.438101	3.319364	1.269106
Proportion Var	0.304614	0.193298	0.061470	0.023502
Cumulative Var	0.304614	0.497912	0.559382	0.582884

Factor #	각 요인의 Loading 절대값이 0.6이상인 변수
Factor 1	Q1, Q2, Q3, Q4, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20, Q21, Q22, Q24, Q26, Q27, Q28, Q30
Factor 2	Q31, Q32, Q33, Q34, Q35, Q36, Q37, Q38, Q48, Q51, Q53, Q54
Factor 3	Q43, Q45, Q46, Q47
Factor 4	Q6, Q7 *(Factor4의 경우 절대값 제일 높은 두가지를 선택함)*

- Promax와 동일한 Loading 값을 가지므로, 동일한 표 구성을 갖는다. 하지만 누적 분산 비율이 61%였던 Promax와 달리 약 59%의 설명력을 갖추고 있다.

5) 요인회전 결과

Varimax

	Factor1	Factor2	Factor3	Factor4
SS Loadings	19.798426	13.708014	7.657385	2.687099
Proportion Var	0.366638	0.253852	0.141803	0.049761
Cumulative Var	0.366638	0.620490	0.762293	0.812054

Quartimax

	Factor1	Factor2	Factor3	Factor4
SS Loadings	39.969521	1.650155	1.410461	0.820787
Proportion Var	0.740176	0.030558	0.026120	0.015200
Cumulative Var	0.740176	0.770735	0.796854	0.812054

Promax

	Factor1	Factor2	Factor3	Factor4
SS Loadings	18.125711	10.236781	3.706271	1.240803
Proportion Var	0.335661	0.189570	0.068635	0.022978
Cumulative Var	0.335661	0.525231	0.593866	0.616844

Oblimin

	Factor1	Factor2	Factor3	Factor4
SS Loadings	16.449169	10.438101	3.319364	1.269106
Proportion Var	0.304614	0.193298	0.061470	0.023502
Cumulative Var	0.304614	0.497912	0.559382	0.582884

- Factor1의 경우 Varimax를 통해서는 부분 분산 비율이 약 37%, Quartimax를 통해서는 약 74%, Promax를 통해서는 약 34%, Oblimin은 약 30%가 나왔다. Varimax와 Promax는 유사한 수치가 나왔으며, Oblimin을 했을 경우에는 4%정도 낮아진 수치로 도출되었다. 특히 Quartimax는 74%가 나왔는데, 이는 제1요인에 과대해석되는 해당 회전 방식의 특성으로 이루어진 것으로 보인다.

- Factor2의 경우 Varimax를 통해서는 부분 분산 비율이 약 25%, Quartimax를 통해서는 약 3%, Promax를 통해서는 약 19%, Oblimin은 약 19%가 나왔다. Oblimin과 Promax는 유사한 수치가 나왔으며, Varimax을 했을 경우에는 6%정도 낮아진 수치로 도출되었다. 특히 Quartimax는 3%가 나왔는데, 이는 제1요인에 과대해석되어 다른 요인의 비율이 부족한 것으로 보인다. 또한 Varimax가 다른 두 회전 방식보다 높게 나온 이유는 Varimax의 방식이 분산을 최대화하는 방향으로 이루어지기 때문일 것으로 보인다.

- Factor3의 경우 Varimax, Quartimax, Promax, Oblimin 각각 14%, 3%, 7%, 6% 수치가 나

왔다. Oblimin과 Promax는 유사한 수치가 나왔으며, Varimax을 했을 경우에는 8%정도 높아진 수치로 도출되었다. 이 때 발생하는 특성 역시 위에서 언급한대로 각각 회전방식의 특성이 반영되어 보인다.

- Factor4의 경우 Varimax, Quartimax, Promax, Oblimin 각각 5%, 2%, 2%, 2% 수치가 나왔다. Oblimin과 Promax는 유사한 수치가 나왔으며, Varimax을 했을 경우에는 3%정도 높아진 수치로 도출되었다. 이때는 수치 차이가 커 보이지 않는다. 이유는 Factor4의 Loading이 전반적으로 낮고, 해당 요인에서 Loading 값이 비교적 높은 Q6, Q7 변수가 전체 변수와의 상관관계가 낮기 때문에 발생하는 낮은 분산 비율로 보인다.

- 총 4가지 요인회전을 진행해본 결과 제1요인에 과대해석되는 Quartimax를 제외하고는 모두 거의 동일한 변수에서 각 요인 당 Loading의 절대값이 높았다. 즉, 각 요인별 분류된 변수의 종류가 거의 유사하였다. (Rotation 별 표 참조)

- 직교회전의 두가지 방식에서는 누적 분산 비율이 81%가 나온 반면, 사각회전에서는 Promax가 61%, Oblimin 방법에서 59%가 도출되었다. 따라서, 이후 요인 분석에서는 직교회전 방식만을 이용하여 분석을 진행하였고, 두 가지 직교회전 분석에서는 본 과제의 다요인 구조를 갖는 데이터 특성으로 Varimax Rotation 방식을 선택하여 분석해보기로 했다.

3.4. 요인 해석

- Varimax Rotation을 이용하여 도출된 각 요인에서 Loading의 절대값이 높은 변수는 아래의 표와 같다. 이때 절대값의 높고 낮음의 기준은 0.6으로 설정하여 진행하였다.

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)*

(Varimax로 도출된 각 요인 당 변수)

F#	Question 내용
F1	1 | 토론이 악화될 때 우리 중 한 명이 사과하면 토론이 끝납니다.
	2 | 때때로 상황이 어려워지더라도 우리의 차이점을 무시할 수 있다는 것을 알고 있습니다.
	3 | 우리가 필요로 할 때 그것을 우리는 처음부터 내 배우자와 우리의 토론을 하고 수정할 수 있습니다.
	4 | 내가 배우자 와 연락하여 연락하면 결국 효과가 있을 것입니다.

	5	아내와 함께 보낸 시간은 우리에게 특별합니다.
	8	저는 아내와 함께 휴가를 즐깁니다.
	9	저는 아내와 함께 여행하는 것을 좋아합니다.
	10	대부분의 목표는 배우자에게 공통적입니다.
	11	언젠가 돌아 보면 배우자와 나는 서로 조화를 이루고있는 것을 보게 될 것 같습니다.
	12	내 배우자와 나는 개인적 자유 측면에서 비슷한 가치관을 가지고 있습니다.
	13	배우자와 나는 비슷한 오락 감각을 가지고 있습니다.
	14	사람 (어린이 친구 등)에 대한 대부분의 목표 는 동일합니다.
	15	배우자와의 꿈은 비슷하고 조화 롭습니다.
	16	우리는 사랑이 무엇인지에 대해 배우자와 양립 할 수 있습니다.
	17	우리는 배우자와 우리 삶에서 행복 해지는 것에 대해 같은 견해를 공유합니다.
	18	배우자와 나는 결혼이 어떻게되어야하는지에 대해 비슷한 생각을 가지고 있습니다.
	19	배우자와 나는 결혼 생활에서 역할이 어떻게되어야하는지 비슷한 생각을 가지고 있습니다.
	20	배우자와 나는 신뢰에 대해 비슷한 가치를 가지고 있습니다.
	21	아내가 뭘 좋아하는지 정확히 알고 있습니다.
	22	배우자가 아플 때 어떻게 보살핌을 받고 싶어하는지 알고 있습니다.
	24	내 배우자가 자신의 삶에서 어떤 스트레스를 받고 있는지 말해 줄 수 있습니다.
	25	배우자의 내면에 대해 알고 있습니다.
	26	배우자의 기본적인 불안을 알고 있습니다.
	27	제 배우자의 현재 스트레스 원인이 무엇인지 압니다.
	28	배우자의 희망과 소원을 알고 있습니다.
	29	저는 제 배우자를 아주 잘 압니다.
	30	배우자의 친구와 그들의 사회적 관계를 알고 있습니다.
F2	31	배우자와 논쟁 할 때 공격적으로 느껴집니다.
	32	배우자와 대화 할 때 보통 '당신은 항상'또는 '당신은 결코'와 같은 표현을 사용합니다.
	33	토론 중에 배우자의 성격에 대해 부정적인 말을 할 수 있습니다.
	34	토론 중에 불쾌한 표현을 사용할 수 있습니다.
	35	토론 중에 배우자를 모욕할 수 있습니다.
	36	저는 토론할 때 수치스러울 수 있습니다.
	37	배우자와의 대화가 차분하지 않습니다.
	38	배우자가 주제를 여는 방식이 싫습니다.
	51	나는 가정의 문제에 대해 틀린 사람이 아닙니다.
	53	토론할 때 배우자에게 부적절 함을 상기시킵니다.

	54	배우자에게 자신의 무능력에 대해 말하는 것이 두렵지 않습니다.
F3	43	저는 환경을 조금 진정시키기 위해 대부분 침묵합니다.39	우리의 토론은 종종 갑자기 발생합니다.
	45	배우자와상의하는 것보다 묵비권을 행사하고 싶습니다.
	46	내가 토론에서 옳다고해도 나는 내 배우자를 해치기 위해 침묵을 지킵니다.
	47	배우자 와상의 할 때 분노를 다 스릴 수 없을까 두려워 침묵을 지킵니다.
F4	6	우리는 파트너로서 집에서 시간이 없습니다.
	7	우리는 가족보다는 집에서 같은 환경을 공유하는 두 명의 낯선 사람과 같습니다.

(각 Factor당 변수 설명)

1) Factor 1: Affection

- Factor1로 분류된 변수들은 대체로 응답자의 배우자에 대한 관심도와 애정도를 의미한다. 예를 들어, Q5~Q9는 배우자와 함께하는 상황에서의 문항을 의미하고, 그 이하 10번대의 문항들은 응답자의 배우자에 대한 아는 점, 즉 관심을 의미한다. 또한 20번대의 질문들도 응답자가 배우자의 감정을 잘 아는가에 대한 질문이므로, 관심을 의미한다고 볼 수 있다. 앞 1~3번은 갈등 발생 시 쉽게 해결할 수 있는 여부를 의미한다. 전반적으로 배우자에 대한 긍정적인 부분 (애정, 관심 등) 이므로 Factor1을 Affection으로 명칭하였다.

2) Factor 2: Aggression

- Factor2로 분류된 변수들은 대체로 응답자가 배우자와 논쟁 발생 시 배우자에 대한 공격성, 부정적인 태도를 나타낼 수 있냐에 대한 질문이다. 예를 들어, Q37,38,53,54는 배우자에게 공격적인 언행을 하는 내용이다. 51번의 경우에는 응답자 본인의 특성을 나타내는 질문이다. 전반적으로 논쟁이 있는 상황을 가정하여, 응답자가 배우자에게 대할 수 있는 공격적인 태도와 부정적 발언들을 나타내는 질문들이므로, Factor2를 Aggression으로 명칭하였다

3) Factor 3: Silence

- Factor3로 분류된 변수들은 대체로 2번 요인과 마찬가지로 논쟁이 있는 상황을 가정한 질문들이다. 논쟁 발생 시 응답자가 배우자와의 의사소통에서 침묵을 유지할 수 있는 지에 대한 질문이다. 침묵을 하는 경향의 높음과 낮음이 논쟁 상황에 긍정적, 부정적 영향을 주는지에 대한 판단은 어려우나, 질문이 전반적으로 침묵과 관련되어있다. 따라서 높은 숫자로 응답할수록 침묵을 유지하는 경향이 커지는 것이므로, Factor3를 Silence로 명칭하였다.

4) Factor 4: Home-Distance

- Factor4는 Q6와 Q7이 분류되었다. 두 질문은 모두 응답자와 배우자가 집이라는 같은 공간 안에서 서로를 인식하는 정도이다. 질문의 내용은 집 내에서의 서로에 대한 친밀감을 부정적으

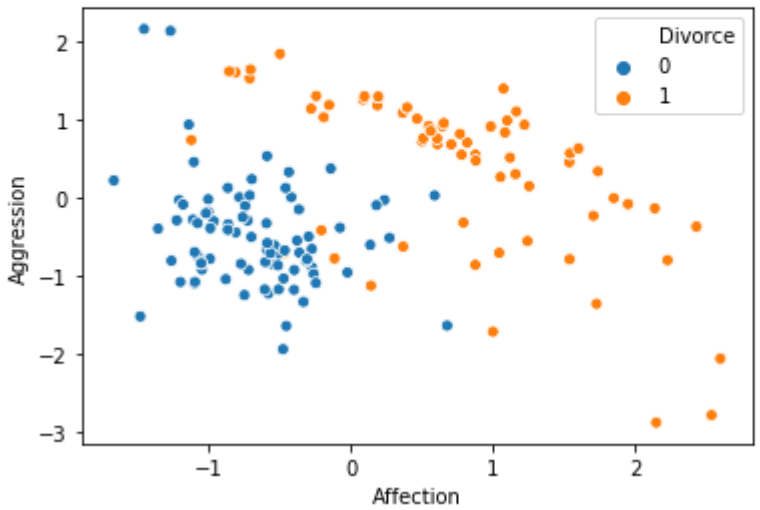
로 표현하였다. 높은 숫자로 응답할수록 집 내에서 서로를 무관심하게 인식하는 것으로 보인다. 따라서 Factor4를 집 내에서의 거리감을 의미하기 위해 Home-Distance로 명칭하였다

3.5. Score Plot

1) Affection (F1 factor) x Aggression (F2 factor)

```
# Score plot F1xF2
sns.scatterplot(data=X_ff, x='Affection',y='Aggression',hue=X_ff.index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21417294588>
```




```
X_aff.groupby('Divorce').describe()
```

	Affection							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.671048	0.451867	-1.670673	-1.042883	-0.653008	-0.406049	0.677936
1	84.0	0.687026	0.918651	-1.124784	0.085961	0.680695	1.228074	2.603220

```
X_agg.groupby('Divorce').describe()
```

	Aggression							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.484575	0.666662	-1.933641	-0.877503	-0.584601	-0.150299	2.170047
1	84.0	0.496112	0.992364	-2.873017	-0.072680	0.768889	1.171391	1.850772

- Affection과 Aggression에 대해 Divorce 그룹으로 나누어 데이터의 경향, 평균, 분산, Min, Max 값을 비교해 보았다. 전반적으로 Affection이 낮을수록 Aggression도 낮은 경향이 있어 보인다.

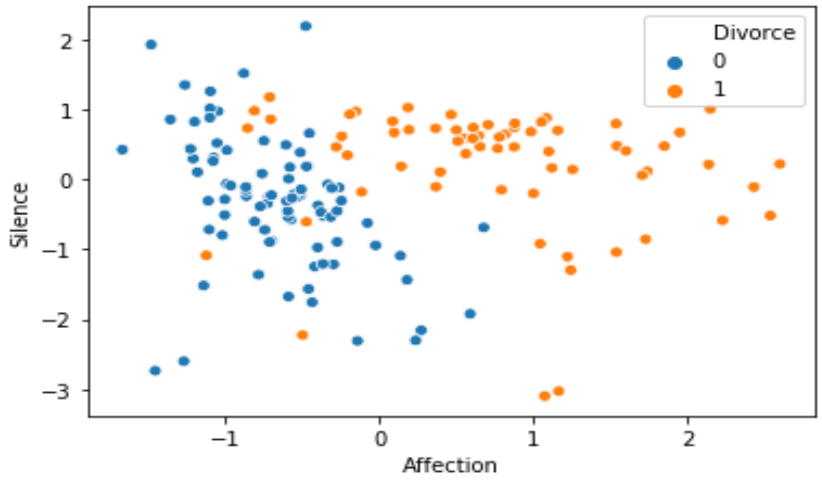
- Divorce를 하지 않은 그룹(Divorce = 0)은 Affection과 Aggression은 전반적으로 낮은 경향으로 보이고, 0에 가까이 분포를 하고 있는 것을 확인 할 수 있었다. Outlier는 좌측상단인 Affection이 낮지만, Aggression이 높은 데이터라 볼 수 있다.

- Divorce를 한 그룹(Divorce = 1)은 Affection은 높고, Aggression은 전반적으로 고르게 분포되어 있는 것을 알 수 있다. Outlier는 우측 하단인 Affection이 높지만, Aggression이 낮은 경우라 볼 수 있을 것이다.

2) Affection (F1 factor) x Silence (F3 factor)

```
# Score plot F1xF3
sns.scatterplot(data=X_ff, x='Affection',y='Silence',hue=X_ff.index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x16f8a4c84c8>
```




```
X_aff.groupby('Divorce').describe()
```

	Affection							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.671048	0.451867	-1.670673	-1.042883	-0.653008	-0.406049	0.677936
1	84.0	0.687026	0.918651	-1.124784	0.085961	0.680695	1.228074	2.603220

```
X_si.groupby('Divorce').describe()
```

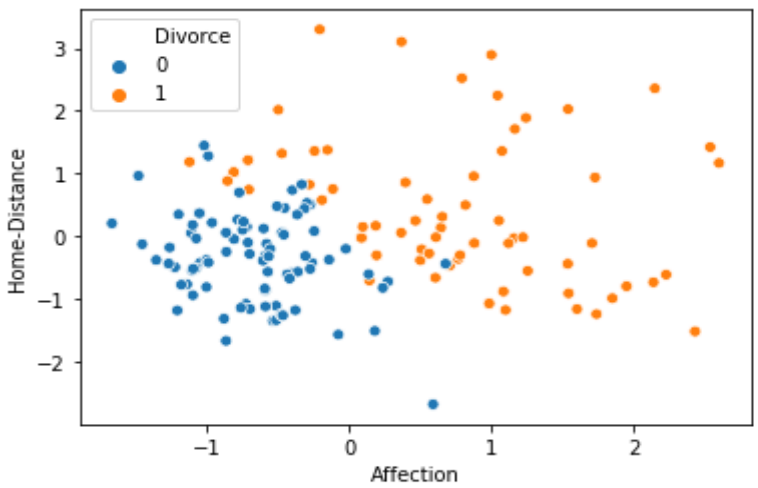
	Silence							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.307683	0.951504	-2.735682	-0.777659	-0.258489	0.288528	2.191510
1	84.0	0.315008	0.832585	-3.096517	0.114555	0.611581	0.807006	1.177418

- Affection과 Silence에 대해 Divorce 그룹으로 나누어 데이터의 경향, 평균, 분산, Min, Max 값을 비교해 보았다.
- Divorce를 하지 않은 그룹(Divorce = 0)은 Affection은 낮지만, Silence은 고르게 분포되어있다. 0에 가까이 분포를 하고 있는 것을 확인 할 수 있었다. Outlier는 좌측상단인 Affection이 낮지만, Aggression이 높은 데이터라 볼 수 있다.
- Divorce를 한 그룹(Divorce = 1)은 Affection은 높고, Silence는 -1에서 1값 사이에 분포 되어있는 것을 볼 수 있다. Outlier는 우측 하단인 Silence의 값이 3에 가까운 데이터 라고 볼 수 있다.

3) Affection (F1 factor) x Home-Distance(F4 factor)

```
# Score plot F1xF4
sns.scatterplot(data=X_ff, x='Affection',y='Home-Distance',hue=X_ff.index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x16fff5c5648>
```



```
X_home.groupby('Divorce').describe()
```

	Home-Distance							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.348145	0.699487	-2.683572	-0.77437	-0.379609	0.115701	1.444670
1	84.0	0.356434	1.093552	-1.520736	-0.44247	0.141061	1.056941	3.296804

```
X_aff.groupby('Divorce').describe()
```

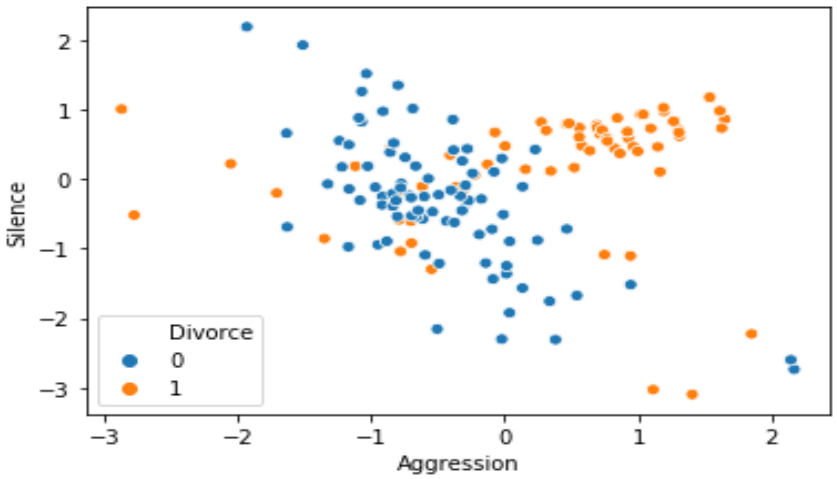
	Affection							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.671048	0.451867	-1.670673	-1.042883	-0.653008	-0.406049	0.677936
1	84.0	0.687026	0.918651	-1.124784	0.085961	0.680695	1.228074	2.603220

- Affection과 Silence에 대해 Divorce 그룹으로 나누어 데이터의 경향, 평균, 분산, Min, Max 값을 비교해 보았다.
- Divorce를 하지 않은 그룹(Divorce = 0)은 Affection은 낮고, Home-Distance도 낮은 경향을 보인다.
- Divorce를 한 그룹은(Divorce = 1)은 Affection은 높고, Home-Distance의 값도 전반적으로 높은 경향을 보인다.

4) Aggression (F2 factor) x Silence(F3 factor)

```
# Score plot F2xF3
sns.scatterplot(data=X_ff, x='Aggression',y='Silence',hue=X_ff.index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x16f8a8e8048>
```



```
X_agg.groupby('Divorce').describe()
```

	Aggression							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.484575	0.666662	-1.933641	-0.877503	-0.584601	-0.150299	2.170047
1	84.0	0.496112	0.992364	-2.873017	-0.072680	0.768889	1.171391	1.850772

```
X_si.groupby('Divorce').describe()
```

	Silence							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.307683	0.951504	-2.735682	-0.777659	-0.258489	0.288528	2.191510
1	84.0	0.315008	0.832585	-3.096517	0.114555	0.611581	0.807006	1.177418

- Aggression과 Silence에 대해 Divorce 그룹으로 나누어 데이터의 경향, 평균, 분산, Min, Max 값을 비교해 보았다.

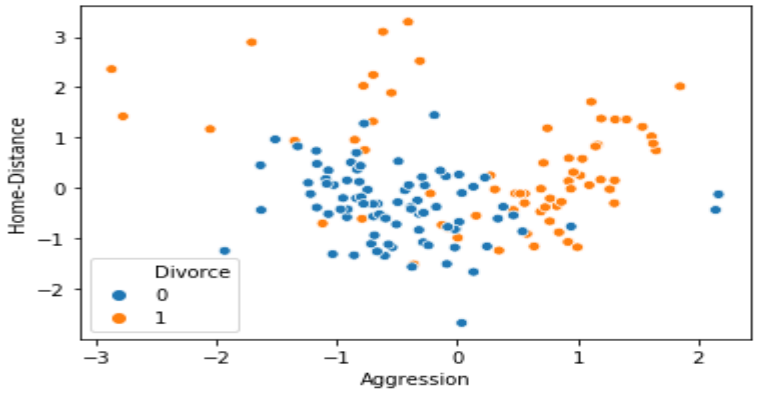
- Divorce를 하지 않은 그룹(Divorce = 0)은 Aggresion은 낮고, Silence는 전반적으로 고른 형태를 띄고 있다. Agreesion이 -1과 0 사이에서 데이터가 주로 분포하고 있는 것을 알 수 있다. Agression이 높고, Silence가 낮은 데이터에 대해서 Outlier라고 할 수 있다.

- Divorce를 한 그룹은(Divorce = 1)은 Affection은 낮고, Silence의 값은 -1에서 1사이에 주로 분포하고 있는 것을 볼 수 있다.

5) Aggression (F2 factor) x Home-Distance(F4 Factor)

```
# Score plot F2xF4
sns.scatterplot(data=X_ff, x='Aggression',y='Home-Distance',hue=X_ff.index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x16f8a5aa648>
```



```
X_agg.groupby('Divorce').describe()
```

	Aggression							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.484575	0.666662	-1.933641	-0.877503	-0.584601	-0.150299	2.170047
1	84.0	0.496112	0.992364	-2.873017	-0.072680	0.768889	1.171391	1.850772

```
X_home.groupby('Divorce').describe()
```

	Home-Distance							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.348145	0.699487	-2.683572	-0.77437	-0.379609	0.115701	1.444670
1	84.0	0.356434	1.093552	-1.520736	-0.44247	0.141061	1.056941	3.296804

- Aggression과 Home-Distance에 대해 Divorce 그룹으로 나누어 데이터의 경향, 평균, 분산, Min, Max 값을 비교해 보았다.

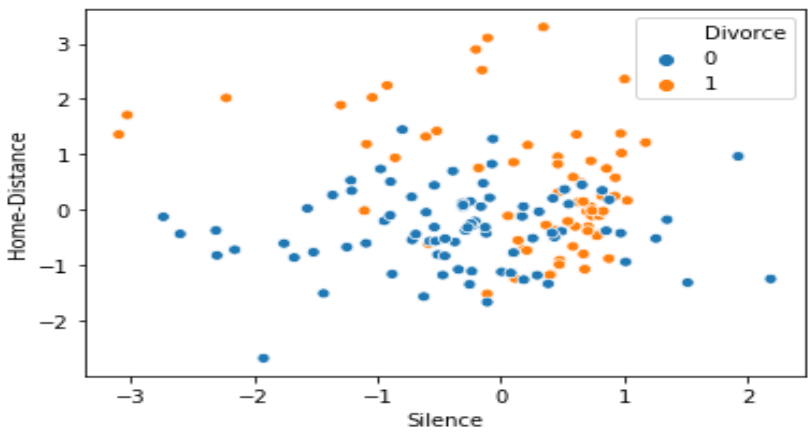
- Divorce를 하지 않은 그룹(Divorce = 0)은 Aggresion은 낮고, Home-Distance도 낮은 데이터의 경향을 보인다. Agreesion이 0 주변에서 데이터가 주로 분포하고 있는 것을 알 수 있다. Aggression이 높고, Home-Distance가 0에 가까운 데이터를 Outlier라고 볼 수 있다.

- Divorce를 한 그룹(Divorce = 1)은 Aggression은 0에 가까이 분포하고 있지만, 약간 높은 편으로 볼 수 있다. Home-Distance도 전반적으로 높은 경향을 보이고 있다. 그 중 Aggression이 낮지만, Home-Distance가 높은 경우 Outlier라고 볼 수 있다.

6) Silence (F3 factor) x Home-Distance(F4 factor)

```
# Score plot F3xF4
sns.scatterplot(data=X_ff, x='Silence',y='Home-Distance',hue=X_ff.index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x16f8a6996c8>
```



```
X_si.groupby('Divorce').describe()
```

	Silence							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.307683	0.951504	-2.735682	-0.777659	-0.258489	0.288528	2.191510
1	84.0	0.315008	0.832585	-3.096517	0.114555	0.611581	0.807006	1.177418

```
X_home.groupby('Divorce').describe()
```

	Home-Distance							
	count	mean	std	min	25%	50%	75%	max
Divorce								
0	86.0	-0.348145	0.699487	-2.683572	-0.77437	-0.379609	0.115701	1.444670
1	84.0	0.356434	1.093552	-1.520736	-0.44247	0.141061	1.056941	3.296804

- Aggression과 Home-Distance에 대해 Divorce 그룹으로 나누어 데이터의 경향, 평균, 분산, Min, Max 값을 비교해 보았다.
- Divorce를 하지 않은 그룹(Divorce = 0)은 Silence와 Home-Distance는 0 주위로 분퍼져있는 것을 확인할 수 있다.
- Divorce를 한 그룹(Divorce = 1)은 Silence에 대해선 0 주위로 퍼져있지만, Home-Distance에 대해서는 높은 경향을 보인다.

4. 결론 – 최종 결과분석

Principal Component	독립변수의 계수의 절댓값이 0.2 이상인 변수
PC1	Q8, Q15, Q19, Q25, Q29, Q30, Q34, Q36, Q38
PC2	Q22, Q23, Q24, Q25, Q26, Q32
PC3	Q8, Q12, Q16, Q18, Q23, Q31
PC4	Q10, Q19, Q31, Q32, Q33

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)*

- PCA를 진행하여 54개의 독립변수로 데이터를 설명할 수고를 들이지 않고, 4개의 Principal Component로 데이터를 84%에 가깝게 설명하면서도 차원을 크게 축소하여, 데이터 설명의 효율성을 높일 수 있었다. 하지만 이 data의 경우, principal component의 linear combination의 계수들을 보면, 그 component를 명확히 대표할 수 있을만한 원변수들이 존재하지 않았고, PCA에서 score plotting을 해본 결과, 데이터를 component관점에서 명확히 특징지을 수 없는 경우도 발생했다.

- 그러한 이유로 FA를 진행해본 결과, 4개의 factor로 정리할 수 있었다. 각각 Factor에 있는 변수들의 특징에 따라 Affection, aggression, silence, home-distance라는 4가지 키워드로 묶을 수 있었다. PC보다 Factor에서 각각의 특징이 훨씬 부각되어 나타났고, 설명력 또한 PCA의 경우 약 83%. FA의 경우 81%로 큰 차이가 나지 않았다. 그러므로 본 데이터의 해석과 차원축소를 하기 위해서는 PCA보다 FA가 더 나은 지표라는 결론을 지을 수 있었다.

- 설문조사의 특성상, 개인마다 주관적으로 해석할 수 있다. 예를 들면, 원 변수에서는 Q6의 Feature는 We don't have time at home as partners 그리고 Q7의 Feature는 We are like two strangers who share the same environment at home rather than family 이다. 둘은 다른 항목에 비해 상관계수가 낮게 나왔다. Q6의 경우 그 이유가 단지 바빠서 시간이 없다. 혹은 사이가 나빠서 서로 집에서 지내는 시간이 없다 등등 다양한 이유로 해석이 가능하기 때문이다. 따라서, 주관적으로 설문된 데이터가 분석에 영향도 미쳤다고 본다.