

국내 지역구 별 8values 정치 지형 평가

제21대 국회의원 선거를 중심으로

분산시스템 기말 프로젝트

박종현, 공과대학 컴퓨터정보통신공학과

jonghyeon@jnu.ac.kr

본 프로젝트에서는 대한민국의 21대 국회의원 선거 결과를 분석하여 우리나라의 각 지역구 별 정치 지형을 이해하는 것을 목표로 한다.

배경

우리나라는 주요 3개 정당의 후보가 대략 30% 전후의 득표율을 보였던 제 13대 대통령 선거 이후, 단일화와 정치적 사건을 거듭하여 거대 양당을 중심으로 한 정치 지형이 형성되어 있다.

일반적으로 이러한 거대 양당의 대결 구도는 정치적 대립을 명확하게 보여주어 외부에서 정치적 상황을 직관적으로 이해할 수 있게 한다. 2025년 대한민국에서는 더불어민주당(민주당)을 진보 성향으로, 국민의힘을 보수 성향으로 분류하여 정치 상황을 이해하려고 한다.

하지만 실제로 정치에서 다루는 분야와 사상, 이념은 매우 폭넓어서 단순히 이와 같이 이분법적으로 나누는 것에는 한계가 있다.

다양한 정치학과 이념들이 존재하기 때문에, 해외에서는 다양한 사상적 근거를 바탕으로 다양한 군소 정당이 실질적으로 정치에 참여하여, 조금 더 세분화된 정치 스펙트럼을 대변하는 사례도 존재한다.

우리나라의 경우에도 거대 양당을 제외한 3지대 정당, 군소 정당이 존재하는 것도 이러한 정치 스펙트럼을 대변하려는 시도로 볼 수 있다. 그럼에도 불구하고 제 3지대는 정치적 영향력이 미비하여 실질적으로 거대 양당의 대결 구도만이 정치적 상황을 대변하는 것으로 여겨진다.

하지만 복잡하고 고차원적인 정치 스펙트럼을 민주당과 국민의힘으로 구분하는 것은 유권자들의 성향과 이념을 지나치게 단순화하여 표현하여 유권자들의 실제 정치 성향을 왜곡하곤 한다. 이러한 왜곡은 유권자 뿐 아니라 정치 지형의 당사자인 각 정당에 있어서도 발생한다.

최근의 사례로는 민주당은 이념적으로 중도 보수에 가깝지만 진보 성향으로 표현되고 있다는 이재명 대표의 발언[1]이 있었다. 이러한 발언이 민주당이 진보 성향으로 표현되는 것에 대한 반대의견인지, 정치공학적인 행동인지 다양한 평가와 비판, 논의가 있었으나, 정당의 이념과 성향을 단순히 1차원적으로 구분하는 것의 한계로서도 이해할 수 있다.

그럼에도 국내의 정치 지형은 진보-보수 축으로만 논의되고 있다. 지역 간 성향, 정치인과 정당별 성향 등 정치적 담론 모두 한개의 대립으로 근사하여 표현된다.

본 프로젝트에서는 이 대립 구도의 기성 담론을 해체하여 조금 더 구체적으로 정치 지형을 분석하려고 한다.

8values 정치 성향 평가 ———

이 평가에서 사용하는 8values 지표는 현대의 정치 성향을 경제 축, 외교 축, 시민 축, 사회 축, 4개의 대립 축을 사용하여 성향을 요약하는 간이 평가 도구이다. 이 축선은 각각 평등/시장 지향(경제), 국가/국제 지향(외교), 자유/권위 지향(시민), 전통/진보 지향(사회)을 표현한다.

이 평가는 개인에게 수행되도록 설계되었다. 평가 대상은 사상과 이념, 가치관에 따라 의견이 변화할 수 있는 70개 진술에 대해 “매우 동의”부터 “매우 비동의”까지 5단계로 응답할 수 있다.

응답이 종료되면 각 문항의 응답에 문항이 갖는 가치 값을 가중 합산하여, 8개 가치에 대해 대상의 성향을 Figure 7 와 같이 수치화하고 요약한다.

8values 역시 개인의 가치를 4개 차원으로 근사하여 평가한다는 점에서 현실상을 온전히 표현하기는 어렵다. 하지만 신뢰성과 포괄성을 갖추고 있다[2]고 평가되거나 [3], [4] 과 같이 학술적으로 이용되기도 하는 등 정치 성향을 조금 더 세밀하게 이해하는 데 유용함이 인정되었다.

읍면동명	투표구명	선거인수	투표수	정당별 득표수						무효 투표 수	기권수
				민생당	미래한국당	더불어시민당	정의당	우리공화당	...		
합계		460,663	316,462	5,708	146,035	68,620	21,128	1,631	...	8,763	144,201
거소·선상 투표		564	531	15	223	104	43	1	...	47	33
관외사전투표		33,829	33,800	785	12,619	9,223	2,665	125	...	1,083	29
재외투표		5,378	1,036	12	434	255	67	3	...	17	4,342
재외투표 (공관)		0	0	0	0	0	0	0	...	0	0
신사동	소계	13,991	9,442	149	5,252	1,593	456	38	...	260	4,549
	관내사전투표	2,716	2,715	48	1,281	655	147	7	...	59	1
	신사동제1투	2,370	1,177	19	629	192	58	5	...	28	1,193

Table 1: 서울 강남구의 비례대표 투표 개표 결과

데이터의 특징 및 수량

분석하고자 하는 데이터에 대한 설명

데이터가 표현하는 사회상 ——

이 프로젝트에서는 2020년 이루어진 제 21대 국회의원 선거 개표 결과를 사용한다.[5]

2020년 이후 2024년의 윤석열 정부 비상 계엄과 같이 국내 정치 상황을 크게 변동시킬 수 있는 주요한 정치적 사건이 발생하였고, 그 사이 집권여당이 변경되는 두 번의 대선, 한 번의 탄핵이 있었다.

따라서 이 평가를 수행하는 데 사용되는 데이터는 2025년의 정치 상황을 이해하는 데 보조적으로 사용되거나, 최근 수 년 간의 지역별 이념 경향을 이해하는 데 활용되어야 한다.

대표적인 예시로는 보수정당 지지세가 강한 대구·경북 지역, 진보정당 지지세가 강한 호남 지역의 지역구별 지지 이념 경향성 파악에 활용이 가능할 것으로 보인다.

데이터의 특징 및 확인 사항 ——

제 21대 국회의원 선거부터 연동형 비례대표제가 적용되어, 지역구 선거와 비례대표 선거로 분리된 병립형으로 진행되었다.

이에 따라 개표 결과 역시 비례대표와 지역구로 구분하여 주어진다. 구체적으로는 Listing 1 과 같이 구분되었다. 각 투표구별로 구분된 개표 결과는 선거인수, 투표수, 정당/후보별 득표수, 무효 투표수, 기권수를 읍면동, 투표구로 분류하여 xlsx 파일로 제공되었다.

- 비례대표 (디렉토리)
 - 시도별
 - 투표구별
 - 1서울
 - 강남구
 - 강동구
 - ...
 - 2부산
 - ...
 - 지역구
 - 시도별
 - 투표구별

Listing 1: 제 21대 국회의원 선거 개표 결과 데이터 파일 구조

예시로 비례대표/1서울/강남구의 개표결과는 Table 1 와 같이 표현되었다.

방법 및 데이터 분석 목표

데이터 분석의 목표 및 도전적 과제

이 프로젝트에서는 대한민국의 21대 국회의원 선거에 참여한 정당들의 가치와 이념을 8values로 평가하여, 이를 각 지역구 별 득표율에 투영하여 지역구 별 정치 지형을 8개 가치로 분석하였다. 최종적으로 지역구 별로 드러나는 8values 분포를 획득하는 것이 목표이다.

구체적으로 다음과 같은 방법으로 프로젝트를 진행한다:

1. 정당별 8values 평가

Table 8 와 같은 프롬프트를 사용하여 인터넷 검색이 가능한 LLM 모델 여러개에 각 정당의 8values의 4개 축을 수치화하도록 한다. 평가

정당명	경제 평등(-1) - 시장(1)		외교 국가(-1) - 국제(1)		국가 자유(-1) - 권위(1)		사회 전통(-1) - 진보(1)		8values 이념분류 [6]
	값	근거	값	근거	값	근거	값	근거	
더불어민주당 더불어시민당	-0.6	[7], [8]	+0.6	[8], [9]	-0.5	[9], [10], [11]	+0.5	[9], [12]	자유지상주의적 사회주의
미래통합당 미래한국당	+0.6	[13], [14]	-0.2	[13], [14]	+0.2	[13], [14]	-0.4	[13], [14]	보수주의
민생당	0.0	[8], [15]	-0.1	[8], [15]	-0.1	[8], [15]	+0.1	[8], [15]	중도주의
정의당	-0.9	[16]	+0.7	[16]	-0.7	[16]	+0.8	[16]	자유지상주의적 공산주의
우리공화당	+0.8		+0.8		+0.9		+0.8		국제주의적 전체주의
민중당	-0.9	[9]	+0.7	[9]	-0.7	[9]	+0.8	[9]	자유지상주의적 공산주의
한국경제당	+0.8		0.0		+0.3		-0.2		국가자본주의
국민의당	+0.1	[9]	+0.2	[9]	-0.1	[9]	+0.1	[9]	자유주의
친박신당	+0.8		-0.8		+0.8		+0.8		전체주의적 자본주의
열린민주당	-0.7		+0.6		-0.6		+0.6		자유지상주의적 사회주의

Table 3: 정당별 8values 가치 가중치 [17]¹

정당	선거구 수	비례대표 출마
더불어민주당	253	-
미래통합당	237	-
민생당	58	0
미래한국당	-	0
더불어시민당	-	0
정의당	76	0
우리공화당	42	0
민중당	59	0
한국경제당	-	0
국민의당	-	0
친박신당	4	0
열린민주당	-	0

Table 2: 제 21대 국회의원 선거 참여 원내 정당

근거가 타당한지 직접 평가 후, 각 축에 대해 평가 결과의 평균을 계산하여 정당별 8values 값 Table 3 을 산출한다.

평가 정당은 총선 당일 선거에 참여한 원내정당 Table 2 을 대상으로, 평가 가능할만큼 근거 자료를 확보할 수 있는 정당에 대해서 수행하였다.

2. 지역구 별 득표율 처리

중앙선거관리위원회가 발표한 제 21대 국회의원 선거 결과를 재처리하여 지역구별 정당 득표율을 산출한다. xlsx 파일로 제공되는 투표소별 선거 결과 데이터를 가공하여 분산 시스템 데이터베이스에 저장하고 처리하는 SW를 작성하고 사용한다.

3. 지역구 별 8values 분포 산출

각 지역구 별로 정당의 8values v_i 를 득표율 r_i 에 따라 가중 평균하여 지역구 별 8values 분포를 산출한다.

$$\text{지역구 8values} = \sum v_i \cdot r_i$$

Figure 1: 지역구별 8values 계산식

예를 들어, A 지역구에서 더불어민주당(민주당)이 50%, 미래통합당(통합당)이 30%, 정의당이 20% 득표율을 보였다면, A 지역구의 8values는 Figure 2 와 같이 계산한다.

$$\begin{aligned} \text{A지역구 8values} = & .5 \times (v_{\text{민주당}}) + \\ & .3 \times (v_{\text{통합당}}) + \\ & .2 \times (v_{\text{정의당}}) \end{aligned}$$

Figure 2: 지역구별 8values 계산식 예시

분석 수행

시스템의 구조

데이터분석시스템 소프트웨어 구조

이 평가에서는 Apache Hadoop, Apache Spark를 사용하여 데이터 분석 시스템을 구축하였다. 선관위에서 제공하고 있는 데이터[5]는 총 635개이므로 Hadoop의 병렬 처리 기능을 이용하여 데이터를 맵리듀스 작업으로 처리하고자 하였다.

Apache Hadoop (하둡)

하지만 실제로는 하둡의 파일 시스템인 HDFS만을 사용하고, 시스템은 HDFS로부터 선거 데이터를 가져와 사용하는 것 이상으로 구현되지 않았다.

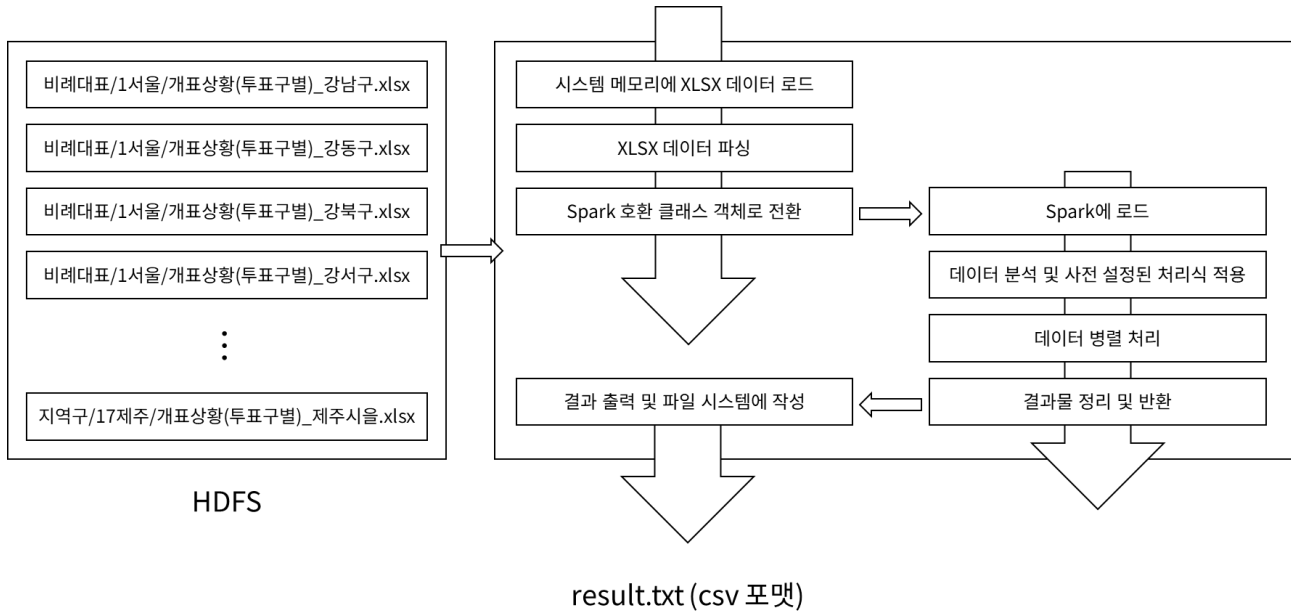


Figure 3: 데이터 분석 시스템의 개요

VoteType	RegionArea	RegionCity	RegionWard	Candidate	Count
비례대표국회의원선거	경상남도	창원시마산합포구	창원시마산합포구	민생당	35
비례대표국회의원선거	경상남도	창원시마산합포구	창원시마산합포구	미래한국당	908
비례대표국회의원선거	경상남도	창원시마산합포구	창원시마산합포구	더불어시민당	286
비례대표국회의원선거	경상남도	창원시마산합포구	창원시마산합포구	정의당	101
비례대표국회의원선거	경상남도	창원시마산합포구	창원시마산합포구	우리공화당	9
비례대표국회의원선거	경상남도	창원시마산합포구	창원시마산합포구	민중당	18
비례대표국회의원선거	경상남도	창원시마산합포구	창원시마산합포구	한국경제당	1

Table 4: 스파크 호환 각 셀 객체의 실제 데이터 예시

Apache Spark (스파크)

이 데이터 분석 시스템은 스파크를 이용하여 데이터를 분석하고 처리한다.

시스템이 분석할 대상 파일을 불러오고 파싱을 마치면 결과물로서 선거 데이터를 표현하는 클래스의 객체를 생성한다. 이 클래스는 스파크에 호환성을 갖도록 작성되어, 스파크에 파싱한 데이터를 로드할 수 있다.

전체 데이터의 스파크 호환 객체를 모두 생성하면, 스파크 세션을 시작하고 스파크에 데이터를 삽입한다.

*이후에는 전체 데이터를 모두 스파크 호환 객체로 생성하는 것이 아니라, 부분적으로 데이터를 스파크 호환 객체로 생성한 후 스파크에 데이터를 로드하면 객체를 제거하는 등의 방식으로 개선할 수 있을 것으로 보인다.

스파크에는 Table 1과 같은 엑셀 데이터의 각 셀 값이 파일에 별도로 기록되는 투표구와 결합된 데

이터를 로드한다. 구체적으로는 Table 4와 같은 구조를 갖는다.

데이터가 스파크에 모두 로드되면, 스파크로부터 데이터를 로드하며 Figure 1 수식을 적용하여 8values 가치평가를 수행한다.

수행 결과

각 선거 단위에 대하여 8values 평가를 수행한 결과, 자유주의가 56.4%로 가장 높은 비율을 보였고, 중도주의 23.4%, 자유지상주의적 사회주의 13.1%, 사회적 자유지상주의 3.4%가 줄을 이었다.

자유주의의 과반 등장 ——

다만 Table 5의 집계에서 자유주의 항목이 대한민국 국회 의석인 300석을 초과한다는 점에서 확인할 수 있듯, 8values 평가는 각 선거구에 대해서만 수행되지 않았다. Table 11에서 확인할 수 있듯, 모든 유형의 선거 단위에 대하여 평가하고, 각 지역의 행정구 단위와 선거구 단위가 구분 없이 모두 합계되었다.

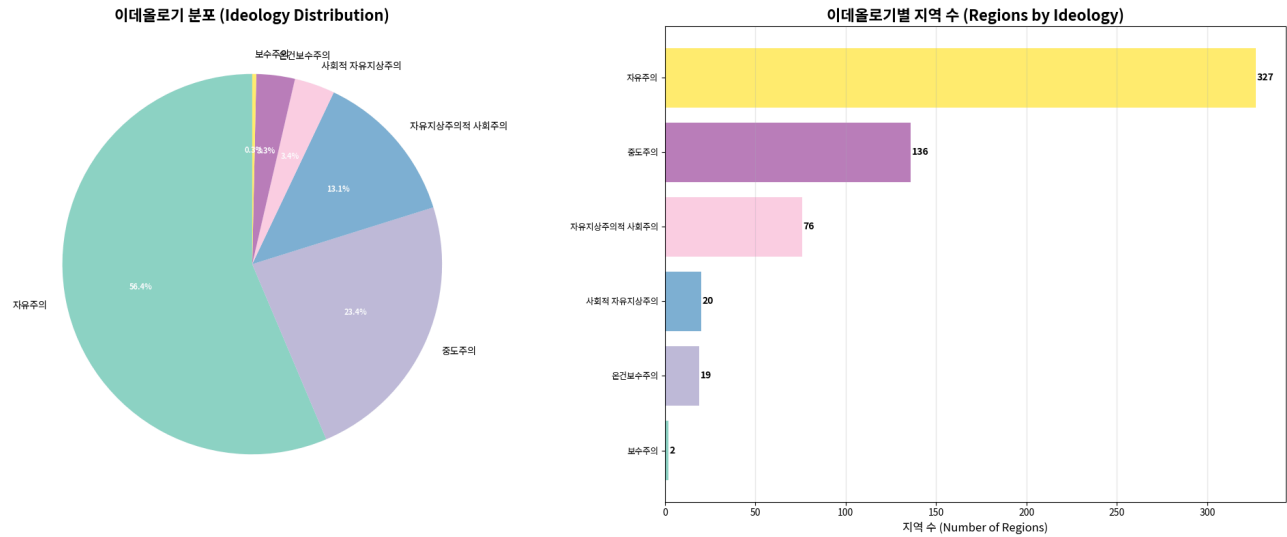


Figure 4: 8values 가치 평가 요약

#	분류	계	%
1	자유주의	327	56.4%
2	중도주의	136	23.4%
3	자유지상주의적 사회주의	76	13.1%
4	사회적 자유지상주의	20	3.4%
5	온건보수주의	19	3.3%
6	보수주의	2	0.3%

Table 5: 전체 지역 단위에 대하여 8values 가치 평가의 수행 결과²

때문에 점유율도 선거구를 많이 가지고 있는 행정구의 성향이 조금 더 두드러져 보이게 하는 다소의 왜곡이 존재한다. 예를 들어, 서울특별시 송파구는 송파구 갑, 을, 병 세 개의 선거구를 갖지만, 충청남도 계룡시의 선거구는 논산시와 금산군에 묶여 세 개의 행정구가 한 개의 선거구를 갖는다.

그럼에도 불구하고 자유주의가 두드러지게 나타난 배경에는 제21대 국회의원 선거에서 민주당계 정당이 큰 승리를 거두고, 보수정당은 패스트 트랙 저지선인 120석에 못 미치는 의석을 확보했기 때문이다. (Figure 9, Table 9)

특히 당시 여당이었던 더불어민주당은 단일 정당 사상 가장 많은 의석을 차지하는 기록을 세움으로써, 문재인 정부가 87년 체제에서 유례 없는 가장 강력한 권한을 획득했다는 평가를 받았다.

시도별 8values 사상 평가 ———

부록의 Figure 10와 같이 전통적으로 범보수진영에서 중시하는 가치를 사각형의 내부에 가깝게, 범진보진영에서 중시하는 가치를 사각형의 외부

축	최대 보수	최대 진보	차이
경제 축	경상북도 (0.272)	광주광역시 (-0.527)	0.799
외교 축	경상북도 (0.031)	광주광역시 (0.514)	0.483
정부 축	경상북도 (0.014)	광주광역시 (-0.455)	0.469
사회 축	경상북도 (-0.128)	광주광역시 (0.460)	0.588

Table 6: 최대 보수/최대 진보 지역 간 수치 차이

에 가깝게 나타나도록 표현한 그래프에서, 경상북도와 대구광역시가 최내각에, 광주광역시와 전라남북도가 최외각에 위치한 것을 확인할 수 있다.

가치관의 차이가 가장 큰 지역인 경상북도와 광주광역시의 8values 가치 평가 값 차이는 경제 0.799, 외교 0.483, 정부 0.469, 사회 0.588로, 경제, 사회, 외교, 정부 순으로 가치관의 차이를 보였다.

$$\frac{\sum \text{ideology}_{\text{each}} * w_{\text{each}}}{4}$$

$$w_{\text{each}} = \begin{cases} -1 & \text{economic} \\ 1 & \text{diplomatic} \\ -1 & \text{civil} \\ 1 & \text{society} \end{cases}$$

Figure 5: 진보 지수 산출 식³

그 외에도 진보 지수 산출 식 Figure 5을 적용하여 산출한 가장 진보적인 개별 지역으로는 경기도 성남시 수정구(진보 지수: 0.614), 가장 보수적인

²투표구, 선거구 등의 선거 단위를 모두 합계한 것으로 실제 선거구 수보다 많음

³4개 축의 각 값을 보수는 감소하는 방향, 진보는 증가하는 방향으로 조정하여 산출 평균한 값

지역	진보지수 Figure 5	이데올로기	경제 성향	외교 성향	정부 성향	사회 성향
가장 진보 성향이 두드러지는 광역시도						
광주광역시	0.489	자유지상주의적 사회주의	0.527 =평등	0.514 =국제	0.455 =자유	0.460 =진보
전라남도	0.468	자유지상주의적 사회주의	0.498 =평등	0.494 =국제	0.436 =자유	0.445 =진보
전라북도	0.455	자유지상주의적 사회주의	0.480 =평등	0.482 =국제	0.424 =자유	0.434 =진보
가장 보수 성향이 두드러지는 광역시도						
경상북도	-0.096	중도주의	-0.272 =시장	0.031 =국가	-0.014 =권위	-0.128 =전통
대구광역시	-0.045	중도주의	-0.204 =시장	0.077 =국가	0.026 =권위	-0.078 =전통
경상남도	0.015	중도주의	-0.122 =시장	0.124 =국가	0.076 =권위	-0.018 =전통

Table 7: 가장 진보/보수 성향이 두드러지는 광역시도

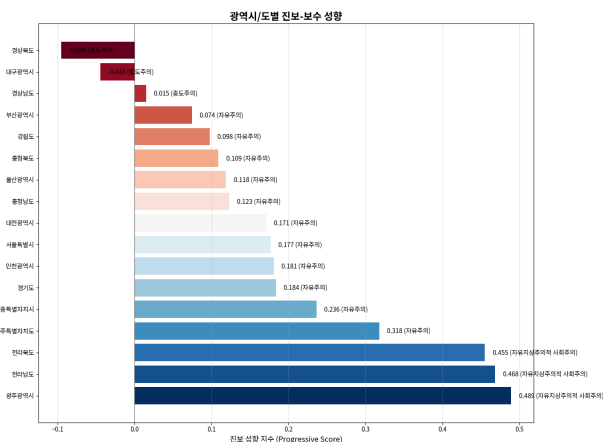


Figure 6: 광역시도별 진보-보수 성향 표

개별 지역으로는 의령군(진보 지수: -0.350)으로 나타났다.

주목할만한 점은 상위 보수 성향 3개 광역시도가 중도주의 범위로 평가되고 있다는 것이다.

21대 총선에서 더불어민주당은 대구-경북 지역의 모든 지역구에 후보자를 내어 보수성향 광역시도에서는 지역구 투표에서도 범진보계 표를 획득할 수 있었으나, 미래통합당을 포함한 범보수계는 호남 지역에 많은 후보자를 내지 않았다. 심지어 더불어민주당 지역구 후보는 정의당 후보와 표가 같려 14.7% 표를 획득한 경주시를 제외하고, 대구-경북 지역의 모든 선거구에서 15% 이상의 표를 얻어 선거비 보전을 받았다.

이로 인하여 호남지역의 진보 성향은 두드러지게 표현되었고, 대구-경북 지역의 보수 성향은 다소 희석되었다.

결론

최초에는 지금까지 수행한 지역 단위 별 8values 가치 평가가 국내의 정치 지형을 구체적으로 표현하는데 유용한 도구가 될 것으로 기대하였다. 하지만 이 가치 평가에 투입한 데이터가 다소 왜곡된 상태에 놓여있을 수 있음을 고려하지 않았다.

단순히 큰 수의 법칙을 따라 데이터가 현실을 잘 반영하고 있을 것으로 생각하였으나, 범보수계의 호남지역 미진출과 같은 변수를 고려하지 않았다는 점에서 이 평가분석은 큰 약점을 가지고 있었다.

하지만 평가 결과에 왜곡이 있더라도 이 평가는 왜곡이 발생된 원인을 조명하거나 고려해볼 수 있게 하였다. 구체적으로, 수행 결과에서 왜곡이 발생한 이유를 “범진보계의 보수 성향 지역 진출과 범보수계의 진보 성향 미진출”에서 발생하였음을 찾을 수 있었다.

다시 말해 현실 상황을 이해하는 데 직접적으로 쓰일 수는 없을지라도 이해에 보조적으로 사용될 수 있다.

이 평가에서는 지역구 투표와 비례대표 후보 투표, 심지어는 보궐선거 투표까지 모두 한데 병합하여 평가를 수행하였다. 이는 꽤 부적절한 방법

이었는데, 선거구를 많이 가지고 있는 행정구역과
궐위가 발생한 선거구의 성향을 더욱 두드러지게
하였다.

선거구의 설정은 보통 인구수를 비례하므로 적절
하다고 판단할 수도 있으나, 인구수를 고려하여
평가 값을 보정하고자 한다면 다른 방식을 고려하
는 것이 더 적절할 것이다.

이후에 이와 관련하여 분석 평가를 더 발전시키고
자 한다면, 위와 같은 문제점을 충분히 보완하여
야 할 것으로 보인다.

부록

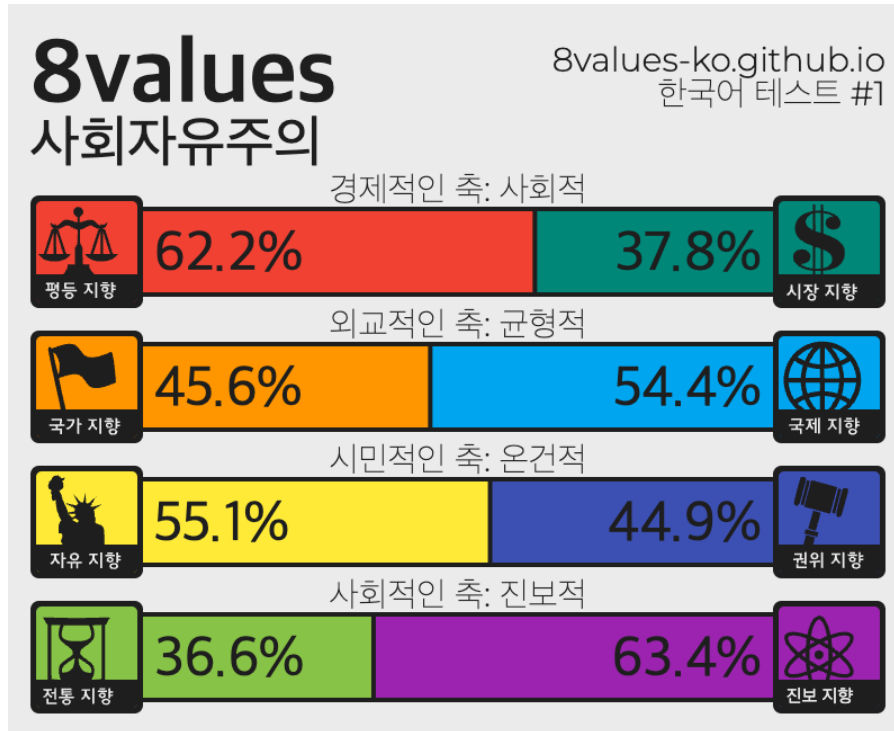


Figure 7: 8values 평가 결과 예시

프롬프트에 작성	번역
<p>8values(https://8values.github.io/) denotes 4 axes representing 8 values like the below:</p> <p>Economic: Equality(-1) - Markets(+1)</p> <p>Diplomatic: Nation(-1) - Globe(+1)</p> <p>Civil: Liberty(-1) - Authority(+1)</p> <p>Society: Tradition(-1) - Progress(+1)</p> <p>Conduct an 8values evaluation on Korea's political parties(at the 21st National Assembly elections) and cite references. If you cannot find enough reference, drop the party from the evaluation.</p> <p>더불어민주당 (+더불어시민당)</p> <p>미래통합당 (+미래한국당)</p> <p>민생당</p> <p>정의당</p> <p>우리공화당</p> <p>민중당</p> <p>한국경제당</p> <p>국민의당</p> <p>친박신당</p> <p>열린민주당</p>	<p>8values(https://8values.github.io/)는 아래와 같이 4개의 축으로 8개 가치를 나타낸다:</p> <p>경제: 평등(-1) - 시장(+1)</p> <p>외교: 국가(-1) - 국제(+1)</p> <p>시민: 자유(-1) - 권위(+1)</p> <p>사회: 전통(-1) - 진보(+1)</p> <p>제 21대 국회의원 선거에 참여한 대한민국의 정당들에 대해 8values 평가를 수행하고, 평가 근거를 제시하라.</p> <p>더불어민주당 (+더불어시민당)</p> <p>미래통합당 (+미래한국당)</p> <p>민생당</p> <p>정의당</p> <p>우리공화당</p> <p>민중당</p> <p>한국경제당</p> <p>국민의당</p> <p>친박신당</p> <p>열린민주당</p>

Table 8: LLM 모델에 입력한 평가 요구 프롬프트

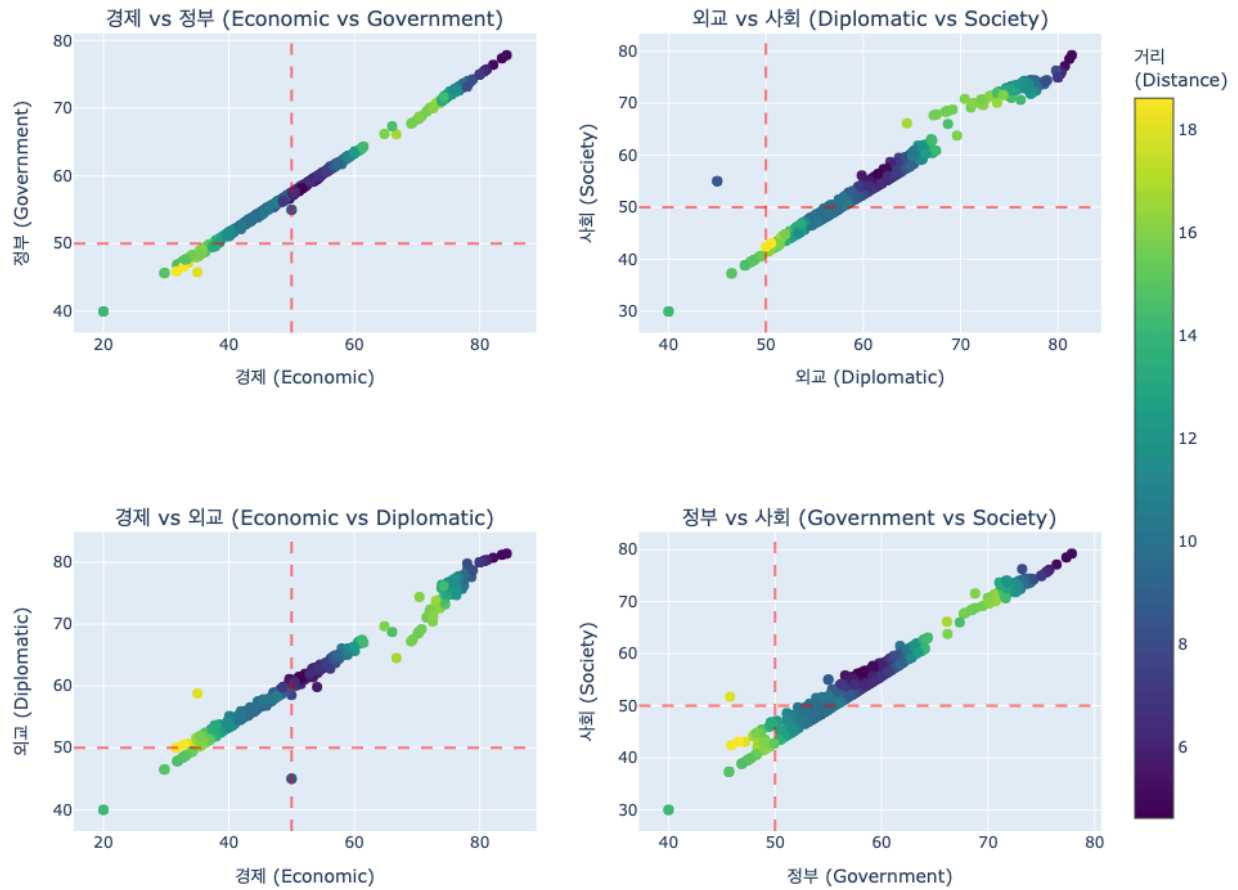


Figure 8: 각 지역 단위 별 두 가치관 수치 차이의 분포

정당	지역구	비례대표	총합	비율
더불어민주당	163석	-	180석	60.00%
더불어시민당	-	17석		
미래통합당	84석	-	103석	34.33%
미래한국당	-	19석		
정의당	1석	5석	6석	2.00%
국민의당	-	3석	3석	1.00%
열린민주당	-	3석	3석	1.00%
무소속	5석	-	5석	1.67%

Table 9: 제 21대 총선 결과 표

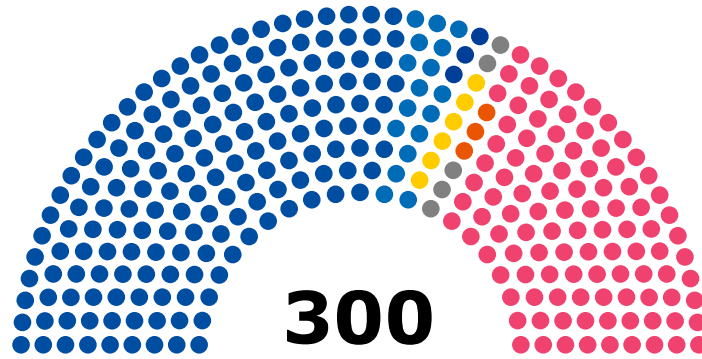


Figure 9: 제21대 총선 결과 인포그래픽

선거구	후보자	공천 유형
광주 서구 갑	주동식	단수공천
광주 북구 갑	범기철	단수공천
전주 을	이수진	단수 공천
익산 갑	김경안	단수 공천
완주·진안·무주·장수	김창도	단수 공천
군산	이근열	단수 공천
목포	황규원	단수 공천
여수 을	임동하	단수 공천
순천·광양·곡성·구례 갑	천하람	단수 공천
순천·광양·곡성·구례 을	김창남	단수 공천
나주·화순	최공재	단수 공천
무안·신안·영암	이인	단수공천

Table 10: 미래통합당의 호남지역 지역구 후보 목록

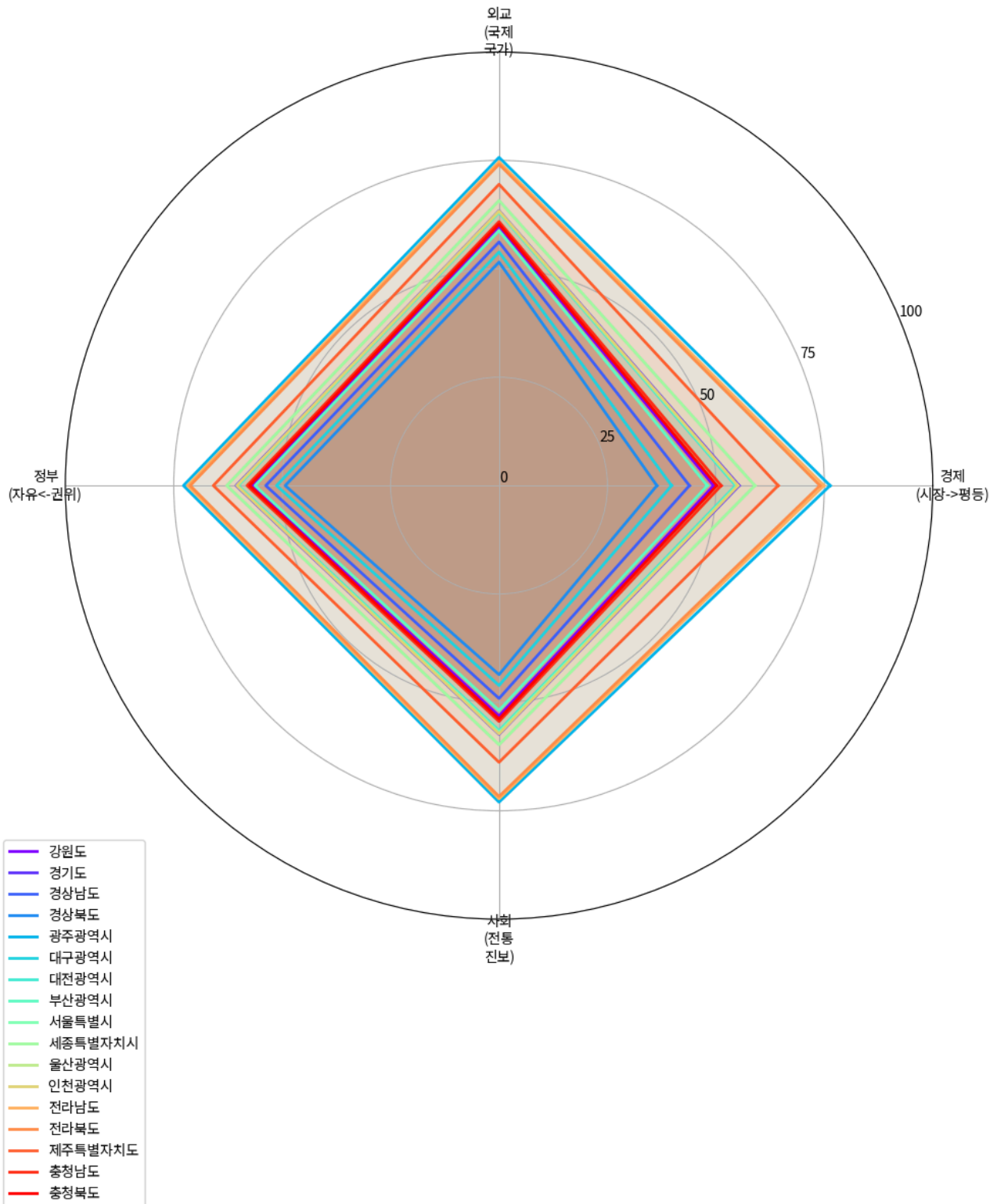


Figure 10: 시도별 8values 가치 분포 사각형 그래프

소스코드 —

부록에는 코드 흐름과 프로젝트를 이해하는 데 활용 가능한 소스코드만을 첨부하였다. 첨부한 소스코드 만으로도 동일한 구현체를 재구현하는데 무리가 없으나, 각종 도구의 설정값과 폴더 구조를 포함하여 전체 구성을 다음의 리포지토리에서도 확인할 수 있다:

<https://github.com/ShapeLayer/election-result-8values-spark>

```
src/main/java/me.jonghyeon.election_result_8values/Main.java

package me.jonghyeon.election_result_8values;

import io.github.cdimascio.dotenv.Dotenv;
import io.github.cdimascio.dotenv.DotenvException;
import me.jonghyeon.election_result_8values.commons.Defaults;
import me.jonghyeon.election_result_8values.commons.Env;
import me.jonghyeon.election_result_8values.commons.PartySupports;
import me.jonghyeon.election_result_8values.controllers.*;
import me.jonghyeon.election_result_8values.models.VoteCounted;
import me.jonghyeon.election_result_8values.models.Struct8Values;
import org.apache.hadoop.fs.FileSystem;

import java.io.IOException;
import java.io.PrintWriter;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;

class MainArgs {
    public static final String USAGE = "ElectionResult8Values --input=<input-path> [--hadoop] [--spark] [--output=<hdfs-output-path>] [--namenode=<hdfs-namenode-url>]";

    public boolean useHadoop = false;
    public boolean useSpark = false;
    public String[] inputPaths;
    public String outputPath;
    public String nameNode;

    public static MainArgs from(String[] args) {
        MainArgs mainArgs = new MainArgs();
        for (String arg : args) {
            if (arg.startsWith("--input=")) {
                mainArgs.inputPaths = parsePathArr(
                    arg.split("=", 2)[1]
                );
            } else if (arg.equals("--hadoop")) {
                mainArgs.useHadoop = true;
            } else if (arg.equals("--spark")) {
                mainArgs.useSpark = true;
            } else if (arg.startsWith("--output=")) {
                mainArgs.outputPath = arg.split("=", 2)[1];
            } else if (arg.startsWith("--namenode=")) {
                mainArgs.nameNode = arg.split("=", 2)[1];
            }
        }

        if (!mainArgs.validate()) {
            System.err.println(USAGE);
            System.exit(1);
        }
    }
}
```

```

    mainArgs.fillDefaults();
    mainArgs.validate();
    return mainArgs;
}

public void fillDefaults() {
    if (this.outputPath == null) {
        this.outputPath = this.useHadoop ? Defaults.DEFAULT_OUTPUT_PATH : "." +
Defaults.DEFAULT_LOCAL_OUTPUT_PATH;
        this.outputPath += "/" + Defaults.DEFAULT_RESULT_REPORT_NAME;
    }
    if (this.nameNode == null) {
        this.nameNode = Defaults.DEFAULT_HDFS_NAMENODE;
    }
}

public static String[] parsePathArr(String path) {
    return path.split(",");
}

public boolean validate() {
    boolean result = true;
    result &= inputPaths != null;

    return result;
}

}

public class Main {
    public static Env env = new Env();

    public static void main(String[] args) throws IOException {
        // Parse args
        MainArgs mainArgs = MainArgs.from(args);

        List<List<VoteCounted>> parsed = null;

        // Input
        FileSystem fs = null; // used when mainArgs.useHadoop is true only

        System.out.println("Input paths: " + String.join(", ", mainArgs.inputPaths));
        if (!mainArgs.useHadoop) {
            System.out.println("Running in local mode. Use --hadoop to run on HDFS.");
            parsed = LocalExcelHandler.parseAll(mainArgs.inputPaths);
        } else {
            System.out.println("Using HDFS NameNode: " + mainArgs.nameNode);
            fs = HdfsConfig.getFileSystem(mainArgs.nameNode);
            parsed = HDFSExcelHandler.parseAll(mainArgs.inputPaths, fs);
        }

        // Output
        System.out.println("Output path: " + mainArgs.outputPath);

        if (mainArgs.useSpark) {
            System.out.println("Using Apache Spark for data processing...");
            SparkHandler sparkHandler = new SparkHandler();
            try {
                Map<String, Map<String, Map<String, Struct8Values>>> results =
sparkHandler.proc(parsed);

```

```

        System.out.println("=== Processing Complete ===");
        System.out.println("Final calculated Struct8Values results by region:");
        for (Entry<String, Map<String, Map<String, Struct8Values>>> entry :
results.entrySet()) {
            for (Entry<String, Map<String, Struct8Values>> subEntry :
entry.getValue().entrySet()) {
                System.out.println("Region: " + entry.getKey() + ", Area: " +
subEntry.getKey());
                for (Entry<String, Struct8Values> valuesEntry :
subEntry.getValue().entrySet()) {
                    System.out.println("  Candidate: " + valuesEntry.getKey() + ",
Struct8Values: " + valuesEntry.getValue());
                }
            }
            System.out.println("Region " + entry.getKey() + ": " +
entry.getValue().toString());
        }

        // Write results to file
        PrintWriter writer = null;
        try {
            if (!mainArgs.useHadoop) {
                java.nio.file.Path outputPath = Paths.get(mainArgs.outputPath);
                if (!Files.exists(outputPath.getParent())) {
                    Files.createDirectories(outputPath.getParent());
                }
                writer = new PrintWriter(Files.newBufferedWriter(outputPath));
            } else {
                org.apache.hadoop.fs.Path outputPath = new
org.apache.hadoop.fs.Path(mainArgs.outputPath);
                if (!fs.exists(outputPath.getParent())) {
                    fs.mkdirs(outputPath.getParent());
                }
                writer = new PrintWriter(fs.create(outputPath, true));
            }

            writer.println("Region,Economic,Diplomatic,Civil,Society");
            // Flatten and write all entries to CSV
            for (Entry<String, Map<String, Map<String, Struct8Values>>> regionEntry :
results.entrySet()) {
                String region = regionEntry.getKey();

                for (Entry<String, Map<String, Struct8Values>> areaEntry :
regionEntry.getValue().entrySet()) {
                    String area = areaEntry.getKey();

                    for (Entry<String, Struct8Values> candidateEntry :
areaEntry.getValue().entrySet()) {
                        String candidate = candidateEntry.getKey();
                        Struct8Values values = candidateEntry.getValue();

                        writer.println(String.format("%s,%.4f,%.4f,%.4f,%.4f",
"\\" + region + "\",\" + area + "\",\" + candidate + "\",
values.economic,
values.diplomatic,
values.civil,
values.society));
                    }
                }
            }
        }
    }
}

```

```

    } finally {
        if (writer != null) {
            writer.close();
        }
    }
} finally {
    sparkHandler.close();
}
} else {
    // Original CSV output logic
    PrintWriter writer = null;
    if (!mainArgs.useHadoop) {
        java.nio.file.Path outputPath = Paths.get(mainArgs.outputPath);

        if (!Files.exists(outputPath.getParent())) {
            Files.createDirectories(outputPath.getParent());
        }
        writer = new PrintWriter(Files.newBufferedWriter(outputPath));
    } else {
        org.apache.hadoop.fs.Path outputPath = new
org.apache.hadoop.fs.Path(mainArgs.outputPath);

        if (!fs.exists(outputPath.getParent())) {
            fs.mkdirs(outputPath.getParent());
        }
        writer = new PrintWriter(fs.create(outputPath, true));
    }

    writer.println("VoteType,RegionArea,RegionCity,RegionWard,Candidate,Count");

    for (List<VoteCounted> votesCountedList : parsed) {
        for (VoteCounted voteCounted : votesCountedList) {
            writer.println(String.format("%s,%s,%s,%s,%s,%d",
                voteCounted.voteType,
                voteCounted.regionArea,
                voteCounted.regionCity,
                voteCounted.regionWard,
                voteCounted.candidate,
                voteCounted.count));
        }
    }

    writer.close();
}

if (mainArgs.useHadoop && fs != null) {
    fs.close();
}
System.out.println("Processing completed. Results written to: " +
mainArgs.outputPath);
}
}

```

```
src/main/java/me.jonghyeon.election_result_8values/controllers/SparkHandler.java
```

```

package me.jonghyeon.election_result_8values.controllers;

import me.jonghyeon.election_result_8values.Main;
import me.jonghyeon.election_result_8values.commons.Defaults;
import me.jonghyeon.election_result_8values.commons.PartySupports;

```

```

import me.jonghyeon.election_result_8values.models.Party;
import me.jonghyeon.election_result_8values.models.VoteCounted;
import me.jonghyeon.election_result_8values.models.Struct8Values;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.api.java.function.Function;
import org.apache.spark.sql.Dataset;
import org.apache.spark.sql.Row;
import org.apache.spark.sql.Session;
import org.apache.spark.sql.types.DataTypes;
import org.apache.spark.sql.types.StructField;
import org.apache.spark.sql.types.StructType;

public class SparkHandler {
    Session spark;

    public SparkHandler() {
        this.spark = Session
            .builder()
            .appName(Defaults.DEFAULT_SPARK_APPNAME)
            .master(Defaults.DEFAULT_SPARK_MASTER)
            .getOrCreate();
    }

    public Map<String, Map<String, Map<String, Struct8Values>>> proc(
        List<List<VoteCounted>> parsed
    ) {
        Map<Party, Struct8Values> partyValues = PartySupports.partyEnumTo8Values;

        List<VoteCounted> flattenedData = new ArrayList<>();
        for (List<VoteCounted> votesCountedList : parsed) {
            flattenedData.addAll(votesCountedList);
        }

        JavaSparkContext sc = new JavaSparkContext(spark.sparkContext());
        JavaRDD<VoteCounted> voteRDD = sc.parallelize(flattenedData);

        Dataset<Row> voteDF = createVoteRDDDataFrame(voteRDD);

        System.out.println("=== Processing Struct8Values Calculation by Region (Area/City/
Ward) ===");

        Dataset<Row> regionTotals = voteDF
            .groupBy("regionArea", "regionCity", "regionWard")
            .agg(org.apache.spark.sql.functions.sum("count").alias("totalVotes"));

        Dataset<Row> partyRegionVotes = voteDF
            .groupBy("regionArea", "regionCity", "regionWard", "candidate")
            .agg(org.apache.spark.sql.functions.sum("count").alias("partyVotes"));

        Dataset<Row> partyRegionPercentages = partyRegionVotes
            .join(regionTotals, new String[]{"regionArea", "regionCity", "regionWard"})
            .withColumn("percentage",
                org.apache.spark.sql.functions.col("partyVotes")

```



```

        .divide(org.apache.spark.sql.functions.col("totalVotes"))));

System.out.println("Party votes by region with percentages:");
partyRegionPercentages.show();

List<Row> results = partyRegionPercentages.collectAsList();

Map<String, Map<String, Map<String, Struct8Values>>> regionResults = new
java.util.HashMap<>();

for (Row row : results) {
    String regionArea = row.getString(0);
    String regionCity = row.getString(1);
    String regionWard = row.getString(2);
    String candidate = row.getString(3);
    long partyVotes = row.getLong(4);
    long totalVotes = row.getLong(5);
    double percentage = row.getDouble(6);

    if (!regionResults.containsKey(regionArea)) {
        regionResults.put(regionArea, new java.util.HashMap<>());
    }
    if (!regionResults.get(regionArea).containsKey(regionCity)) {
        regionResults.get(regionArea).put(regionCity, new java.util.HashMap<>());
    }
    if (!regionResults.get(regionArea).get(regionCity).containsKey(regionWard)) {
        regionResults.get(regionArea).get(regionCity).put(regionWard, new
Struct8Values(0, 0, 0, 0));
    }

    Party matchingParty = null;
    for (Party party : partyValues.keySet()) {
        if (PartySupports.toString(party).equals(candidate)) {
            matchingParty = party;
            break;
        }
    }

    if (matchingParty != null) {
        Struct8Values partyValue = partyValues.get(matchingParty);

        Struct8Values weightedValue = partyValue.mul((float) percentage);
        Struct8Values currentWardSum =
regionResults.get(regionArea).get(regionCity).get(regionWard);
        regionResults.get(regionArea).get(regionCity).put(regionWard,
currentWardSum.add(weightedValue));

        System.out.printf("Area: %s, City: %s, Ward: %s, Party: %s, Percentage: %.4f,
Struct8Values: %s, Weighted: %s\n",
            regionArea, regionCity, regionWard, candidate, percentage,
partyValue.toString(), weightedValue.toString());
    }
}

sc.close();

System.out.println("=== Final Weighted Sum (sum(v_i * r_i)) by Region Hierarchy
===");
for (Map.Entry<String, Map<String, Map<String, Struct8Values>>> areaEntry :
regionResults.entrySet()) {

```

```

        System.out.println("Area: " + areaEntry.getKey());
        for (Map.Entry<String, Map<String, Struct8Values>> cityEntry :
areaEntry.getValue().entrySet()) {
            System.out.println("  City: " + cityEntry.getKey());
            for (Map.Entry<String, Struct8Values> wardEntry :
cityEntry.getValue().entrySet()) {
                System.out.println("    Ward: " + wardEntry.getKey() + " -> " +
wardEntry.getValue().toString());
            }
        }
    }
}

return regionResults;
}

public void processVoteCountedData(List<List<VoteCounted>> parsed, String outputPath,
boolean useHadoop) {
    List<VoteCounted> flattenedData = new ArrayList<>();
    for (List<VoteCounted> votesCountedList : parsed) {
        flattenedData.addAll(votesCountedList);
    }

    JavaSparkContext sc = new JavaSparkContext(spark.sparkContext());
    JavaRDD<VoteCounted> voteRDD = sc.parallelize(flattenedData);

    Dataset<Row> voteDF = createVoteRDDDataFrame(voteRDD);

    System.out.println("=== Spark Processing Statistics ===");
    System.out.println("Total records: " + voteDF.count());

    Dataset<Row> candidateSummary = voteDF
        .groupBy("candidate")
        .sum("count")
        .orderBy(org.apache.spark.sql.functions.desc("sum(count)"));

    System.out.println("Top candidates by total votes:");
    candidateSummary.show(10);

    Dataset<Row> regionSummary = voteDF
        .groupBy("regionArea")
        .sum("count")
        .orderBy(org.apache.spark.sql.functions.desc("sum(count)"));

    System.out.println("Top regions by total votes:");
    regionSummary.show(10);

    String finalOutputPath = outputPath;
    if (useHadoop) {
        voteDF.coalesce(1)
            .write()
            .mode("overwrite")
            .option("header", "true")
            .csv(finalOutputPath + "_spark_processed");
    } else {
        voteDF.coalesce(1)
            .write()
            .mode("overwrite")
            .option("header", "true")
            .csv(finalOutputPath + "_spark_processed");
    }
}

```

```

        sc.close();
        System.out.println("Spark processed data saved to: " + finalOutputPath +
            "_spark_processed");
    }

    public void processAndSaveData(Object data, String outputPath, boolean useHadoop,
        String dataType) {
        if ("VoteCounted".equals(dataType)) {
            processVoteCountedData((List<List<VoteCounted>>) data, outputPath, useHadoop);
            return;
        }

        JavaSparkContext sc = new JavaSparkContext(spark.sparkContext());

        if ("Struct8Values".equals(dataType)) {
            createStruct8ValuesTable(data, outputPath, useHadoop, sc);
        }

        sc.close();
    }

    private void createStruct8ValuesTable(Object data, String outputPath, boolean
        useHadoop, JavaSparkContext sc) {
        System.out.println("=== Creating Struct8Values Table ===");

        StructType Struct8ValuesSchema = new StructType(new StructField[]{
            DataTypes.createStructField("economic", DataTypes.FloatType, false),
            DataTypes.createStructField("diplomatic", DataTypes.FloatType, false),
            DataTypes.createStructField("civil", DataTypes.FloatType, false),
            DataTypes.createStructField("society", DataTypes.FloatType, false)
        });

        List<Struct8Values> Struct8ValuesList;
        if (data instanceof List) {
            Struct8ValuesList = (List<Struct8Values>) data;
        } else {
            Struct8ValuesList = new ArrayList<>();
            Struct8ValuesList.add(new Struct8Values(75.5f, -25.3f, 60.2f, -40.1f));
            Struct8ValuesList.add(new Struct8Values(-30.7f, 45.8f, -20.5f, 35.9f));
            Struct8ValuesList.add(new Struct8Values(15.2f, -60.4f, 80.1f, -15.7f));
            Struct8ValuesList.add(new Struct8Values(-50.3f, 20.6f, -35.8f, 70.4f));
        }

        JavaRDD<Struct8Values> Struct8ValuesRDD = sc.parallelize(Struct8ValuesList);
        JavaRDD<Row> rowRDD = Struct8ValuesRDD.map(v8 ->
            org.apache.spark.sql.RowFactory.create(
                v8.economic,
                v8.diplomatic,
                v8.civil,
                v8.society
            )
        );

        Dataset<Row> Struct8ValuesDF = spark.createDataFrame(rowRDD, Struct8ValuesSchema);

        System.out.println("Struct8Values table created with " + Struct8ValuesDF.count() + "
            records");

        System.out.println("Struct8Values Statistics:");
    }

```

```

Struct8ValuesDF.describe().show();

Dataset<Row> avgValues = Struct8ValuesDF.select(
    org.apache.spark.sql.functions.avg("economic").alias("avg_economic"),
    org.apache.spark.sql.functions.avg("diplomatic").alias("avg_diplomatic"),
    org.apache.spark.sql.functions.avg("civil").alias("avg_civil"),
    org.apache.spark.sql.functions.avg("society").alias("avg_society")
);

System.out.println("Average Struct8Values:");
avgValues.show();

Struct8ValuesDF.show();

String Struct8ValuesOutputPath = outputPath + "_Struct8Values";
if (useHadoop) {
    Struct8ValuesDF.coalesce(1)
        .write()
        .mode("overwrite")
        .option("header", "true")
        .csv(Struct8ValuesOutputPath);
} else {
    Struct8ValuesDF.coalesce(1)
        .write()
        .mode("overwrite")
        .option("header", "true")
        .csv(Struct8ValuesOutputPath);
}

System.out.println("Struct8Values table saved to: " + Struct8ValuesOutputPath);
}

private Dataset<Row> createStruct8ValuesRDDDataFrame(JavaRDD<Struct8Values>
Struct8ValuesRDD) {
    StructType schema = new StructType(new StructField[]{
        DataTypes.createStructField("economic", DataTypes.FloatType, false),
        DataTypes.createStructField("diplomatic", DataTypes.FloatType, false),
        DataTypes.createStructField("civil", DataTypes.FloatType, false),
        DataTypes.createStructField("society", DataTypes.FloatType, false)
    });

    JavaRDD<Row> rowRDD = Struct8ValuesRDD.map(v8 ->
        org.apache.spark.sql.RowFactory.create(
            v8.economic,
            v8.diplomatic,
            v8.civil,
            v8.society
        )
    );

    return spark.createDataFrame(rowRDD, schema);
}

private Dataset<Row> createVoteRDDDataFrame(JavaRDD<VoteCounted> voteRDD) {
    StructType schema = new StructType(new StructField[]{
        DataTypes.createStructField("voteType", DataTypes.StringType, false),
        DataTypes.createStructField("regionArea", DataTypes.StringType, false),
        DataTypes.createStructField("regionCity", DataTypes.StringType, false),
        DataTypes.createStructField("regionWard", DataTypes.StringType, false),
        DataTypes.createStructField("candidate", DataTypes.StringType, false),
    });

```

```

        DataTypes.createStructField("count", DataTypes.LongType, false)
    });

    JavaRDD<Row> rowRDD = voteRDD.map(vote ->
        org.apache.spark.sql.RowFactory.create(
            vote.voteType,
            vote.regionArea,
            vote.regionCity,
            vote.regionWard,
            vote.candidate,
            vote.count
        )
    );

    return spark.createDataFrame(rowRDD, schema);
}

public void close() {
    if (spark != null) {
        spark.stop();
    }
}
}

```

src/main/java/me.jonghyeon.election_result_8values/controllers/HDFSExcelHandler.java

```

package me.jonghyeon.election_result_8values.controllers;

import me.jonghyeon.election_result_8values.models.VoteCounted;

import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.RemoteIterator;
import org.apache.hadoop.fs.LocatedFileStatus;

import java.io.IOException;
import java.io.InputStream;
import java.util.*;

public class HDFSExcelHandler extends ExcelHandler {
    public static List<List<VoteCounted>> parseAll(String[] targets, FileSystem fs) throws
    IOException {
        List<Path> targetPaths = new ArrayList<>();

        for (String target : targets) {
            Path currPath = new Path(target);

            if (!fs.exists(currPath)) {
                System.err.println("Target path does not exist in HDFS. Passed: " + target);
                continue;
            }

            FileStatus fileStatus = fs.getFileStatus(currPath);
            if (!fileStatus.isDirectory()) {
                targetPaths.add(currPath);
                continue;
            }

            RemoteIterator<LocatedFileStatus> fileIterator = fs.listFiles(currPath, true);

```

```

        while (fileIterator.hasNext()) {
            LocatedFileStatus eachFileStatus = fileIterator.next();
            if (!eachFileStatus.isDirectory() &&
eachFileStatus.getPath().getName().toLowerCase().endsWith(".xlsx")) {
                targetPaths.add(eachFileStatus.getPath());
            }
        }
    }

    List<List<VoteCounted>> results = new ArrayList<>();
    for (Path path : targetPaths) {
        System.out.println("Processing " + path);
        try (InputStream is = fs.open(path)) {
            results.add(parse(is));
        }
    }

    fs.close();
    return results;
}
}

```

src/main/java/me.jonghyeon.election_result_8values/controllers/LocalExcelHandler.java

```

package me.jonghyeon.election_result_8values.controllers;

import me.jonghyeon.election_result_8values.models.VoteCounted;

import java.io.IOException;
import java.io.InputStream;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.*;
import java.util.stream.Stream;

public class LocalExcelHandler extends ExcelHandler {
    public static List<List<VoteCounted>> parseAll(String[] targets) throws IOException {
        List<Path> targetPaths = new ArrayList<>();

        for (String target : targets) {
            Path currPath = Paths.get(target);

            if (!Files.exists(currPath)) {
                System.err.println("Target path does not exist. Passed: " + target);
            }

            if (!Files.isDirectory(currPath)) {
                targetPaths.add(currPath);
                continue;
            }

            try (Stream<Path> ps = Files.walk(currPath)) {
                ps
                    .filter(path -> !Files.isDirectory(path) &&
path.toString().toLowerCase().endsWith(".xlsx"))
                    .forEach(xlsxPath -> {
                        targetPaths.add(xlsxPath);
                    });
            }
        }
    }
}

```

```

    }

    List<List<VoteCounted>> results = new ArrayList<>();
    for (Path path : targetPaths) {
        System.out.println("Processing " + path);
        try (InputStream is = Files.newInputStream(path)) {
            results.add(parse(is));
        }
    }

    return results;
}
}

```

src/main/java/me.jonghyeon.election_result_8values/controllers/ExcelHandler.java

```

package me.jonghyeon.election_result_8values.controllers;

import me.jonghyeon.election_result_8values.commons.PartySupports;
import me.jonghyeon.election_result_8values.models.Party;
import me.jonghyeon.election_result_8values.models.PartyFounds;
import me.jonghyeon.election_result_8values.models.VoteCounted;
import org.apache.poi.ss.usermodel.*;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public abstract class ExcelHandler {
    public static List<VoteCounted> parse(InputStream is) throws IOException {
        List<VoteCounted> votesCounted = new ArrayList<>();
        Party[] parties = PartySupports.parties;
        try (Workbook workbook = new XSSFWorkbook(is)) {
            // A single sheet is expected.
            Sheet sheet = workbook.getSheetAt(0);
            StringBuilder sb = new StringBuilder();

            int currRow = 0;
            int currCol = 0;

            boolean regionFound = false;
            String[] region = {};

            PartyFounds partyFound = PartyFounds.NotFound;
            Map<Integer, Party> indexPartyMatched = new HashMap<>();

            for (Row row : sheet) {
                for (Cell cell : row) {
                    String now = cell.toString();
                    if (!regionFound) {
                        // has region data
                        if (now.contains("[") {
                            region = now
                                .replace("[", "")
                                .split(",");
                            regionFound = true;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
}

if (partyFound != PartyFounds.Done) {
    boolean partyContains = false;
    for (Party party : parties) {
        String name = PartySupports.toString(party);
        if (now.contains(name)) {
            partyContains = true;
            indexPartyMatched.put(currCol, party);
            break;
        }
    }
    if (partyContains) {
        partyFound = PartyFounds.Finding;
    } else {
        if (partyFound == PartyFounds.Finding) {
            partyFound = PartyFounds.Done;
        }
    }
}

if (partyFound == PartyFounds.Done) {
    if (indexPartyMatched.containsKey(currCol)) {
        Party party = indexPartyMatched.get(currCol);
        if (cell.getCellType() != CellType.NUMERIC) { continue; }

        VoteCounted voteCounted = new VoteCounted(
            region[0],
            region[1],
            region.length >= 3 ? (region.length >= 4 ? region[3] : region[2]) :
region[1],
            region.length >= 3 ? region[2] : region[1],
            PartySupports.toString(party),
            (long) cell.getNumericCellValue()
        );

        votesCounted.add(voteCounted);
    }
}

// finalize
currCol++;
}

// finalize
currRow++;
currCol = 0;
}
}
return votesCounted;
}
}

```

```
src/main/java/me.jonghyeon.election_result_8values/controllers/HdfsConfig.java
```

```

package me.jonghyeon.election_result_8values.controllers;

import me.jonghyeon.election_result_8values.commons.Defaults;
import org.apache.hadoop.conf.Configuration;

```



```

import org.apache.hadoop.fs.FileSystem;

import java.io.IOException;

public class HdfsConfig {
    public static Configuration createConfiguration(String nameNode) {
        Configuration conf = new Configuration();

        if (nameNode != null) {
            conf.set("fs.defaultFS", nameNode);
        } else {
            conf.set("fs.defaultFS", Defaults.DEFAULT_HDFS_NAMENODE);
        }

        conf.set("dfs.client.use.datanode.hostname", "true");
        conf.set("dfs.datanode.use.datanode.hostname", "true");

        return conf;
    }

    public static FileSystem getFileSystem(String nameNode) throws IOException {
        Configuration conf = createConfiguration(nameNode);
        return FileSystem.get(conf);
    }
}

```

src/main/java/me.jonghyeon.election_result_8values/models/Party.java

```

package me.jonghyeon.election_result_8values.models;

public enum Party {
    DEMOCRATIC_PARTY_OF_KOREA,
    DEMOCRATIC_CITIZENS_PARTY,
    UNITED_FUTURE_PARTY,
    FUTURE_KOREA_PARTY,
    MINSAENGDANG,
    JUSTICE_PARTY,
    OUR_REPUBLICAN_PARTY,
    THE_PEOPLES_PARTY_OF_SOUTH_KOREA,
    KOREAN_ECONOMIC_PARTY,
    THE_PEOPLES_PARTY,
    PRO_PARK_NEW_PARTY,
    OPEN_DEMOCRATIC_PARTY,
}

```

src/main/java/me.jonghyeon.election_result_8values/models/PartyFounds.java

```

package me.jonghyeon.election_result_8values.models;

public enum PartyFounds {
    NotFound,
    Finding,
    Done
}

```

src/main/java/me.jonghyeon.election_result_8values/models/Struct8Values.java

```

package me.jonghyeon.election_result_8values.models;

import java.io.Serializable;

```

```
public class Struct8Values implements Serializable {
    public float economic;
    public float diplomatic;
    public float civil;
    public float society;

    public Struct8Values(float economic, float diplomatic, float civil, float society) {
        this.economic = economic;
        this.diplomatic = diplomatic;
        this.civil = civil;
        this.society = society;
    }

    public Struct8Values() {
        this(0, 0, 0, 0);
    }

    @Override
    public String toString() {
        return String.format("Economic: %.2f, Diplomatic: %.2f, Civil: %.2f, Society: %.2f",
            economic, diplomatic, civil, society);
    }

    public Struct8Values add(Struct8Values other) {
        return new Struct8Values(
            this.economic + other.economic,
            this.diplomatic + other.diplomatic,
            this.civil + other.civil,
            this.society + other.society
        );
    }

    public Struct8Values sub(Struct8Values other) {
        return new Struct8Values(
            this.economic - other.economic,
            this.diplomatic - other.diplomatic,
            this.civil - other.civil,
            this.society - other.society
        );
    }

    public Struct8Values mul(float factor) {
        return new Struct8Values(
            this.economic * factor,
            this.diplomatic * factor,
            this.civil * factor,
            this.society * factor
        );
    }

    public Struct8Values divide(float divisor) {
        if (divisor == 0) {
            throw new IllegalArgumentException("Divisor cannot be zero");
        }
        return new Struct8Values(
            this.economic / divisor,
            this.diplomatic / divisor,
            this.civil / divisor,
            this.society / divisor
        );
    }
}
```

```
}  
}
```

src/main/java/me.jonghyeon.election_result_8values/models/VoteCounted.java

```
package me.jonghyeon.election_result_8values.models;  
  
import java.io.Serializable;  
  
public class VoteCounted implements Serializable {  
    public String voteType;  
    public String regionArea;  
    public String regionCity;  
    public String regionWard;  
    public String candidate;  
    public long count;  
  
    public VoteCounted(  
        String voteType,  
        String regionArea,  
        String regionCity,  
        String regionWard,  
        String candidate,  
        long count  
    ) {  
        this.voteType = voteType;  
        this.regionArea = regionArea;  
        this.regionCity = regionCity;  
        this.regionWard = regionWard;  
        this.candidate = candidate;  
        this.count = count;  
    }  
  
    public String getCandidate() {  
        return candidate;  
    }  
  
    public long getCount() {  
        return count;  
    }  
  
    @Override  
    public String toString() {  
        return "VoteCounted{" +  
            "voteType='" + voteType + '\'' +  
            ", regionArea='" + regionArea + '\'' +  
            ", regionCity='" + regionCity + '\'' +  
            ", regionWard='" + regionWard + '\'' +  
            ", candidate='" + candidate + '\'' +  
            ", count=" + count +  
            '}';  
    }  
  
    public String getRegionName() {  
        return regionWard;  
    }  
}
```

src/main/java/me.jonghyeon.election_result_8values/commons/Defaults.java

```
package me.jonghyeon.election_result_8values.commons;
```

```

public class Defaults {
    public static final String DEFAULT_HDFS_NAMENODE = "hdfs://localhost:9000";
    public static final String DEFAULT_OUTPUT_PATH = "/election_result/output";
    public static final String DEFAULT_RESULT_REPORT_NAME = "result.txt";
    public static final String DEFAULT_INPUT_PATH = "/election_result/input";
    public static final String DEFAULT_HDFS_INPUT_PATH = "/election_result/input/hdfs";
    public static final String DEFAULT_LOCAL_INPUT_PATH = "/election_result/input/local";
    public static final String DEFAULT_LOCAL_OUTPUT_PATH = "/election_result/output/local";

    public static final String DEFAULT_SPARK_MASTER = "local";
    public static final String DEFAULT_SPARK_APPNAME = "ElectionAnalyzer";
    public static final String DEFAULT_SPARK_HOME = "/usr/lib/spark/";
}

```

src/main/java/me.jonghyeon.election_result_8values/commons/Env.java

```

package me.jonghyeon.election_result_8values.commons;

import io.github.cdimascio.dotenv.Dotenv;
import io.github.cdimascio.dotenv.DotenvException;

public class Env {
    boolean dotenvEnabled = false;
    Dotenv dotenv;
    public Env() {
        try {
            this.dotenv = Dotenv.load();
            dotenvEnabled = true;
        } catch (DotenvException e) {
            System.err.println(".env not found. use default system env");
        }
    }
    public String get(String name) {
        if (dotenvEnabled) {
            return dotenv.get(name);
        } else {
            return System.getenv(name);
        }
    }
}

```

src/main/java/me.jonghyeon.election_result_8values/commons/PartySupports.java

```

package me.jonghyeon.election_result_8values.commons;

import me.jonghyeon.election_result_8values.models.Party;
import me.jonghyeon.election_result_8values.models.Struct8Values;

import java.util.HashMap;
import java.util.Map;

public class PartySupports {
    public static Map<Party, String> partyEnumToString = new HashMap<Party, String>() {{
        put(Party.DEMOCRATIC_PARTY_OF_KOREA, Presets.NAME_DEMOCRATIC_PARTY_OF_KOREA);
        put(Party.DEMOCRATIC_CITIZENS_PARTY, Presets.NAME_DEMOCRATIC_CITIZENS_PARTY);
        put(Party.UNITED_FUTURE_PARTY, Presets.NAME_UNITED_FUTURE_PARTY);
        put(Party.FUTURE_KOREA_PARTY, Presets.NAME_FUTURE_KOREA_PARTY);
        put(Party.MINSAENGDANG, Presets.NAME_MINSAENGDANG);
        put(Party.JUSTICE_PARTY, Presets.NAME_JUSTICE_PARTY);
        put(Party.OUR_REPUBLICAN_PARTY, Presets.NAME_OUR_REPUBLICAN_PARTY);
        put(Party.THE_PEOPLES_PARTY_OF_SOUTH_KOREA,

```

```

Presets.NAME_THE_PEOPLES_PARTY_OF_SOUTH_KOREA);
    put(Party.KOREAN_ECONOMIC_PARTY, Presets.NAME_KOREAN_ECONOMIC_PARTY);
    put(Party.THE_PEOPLES_PARTY, Presets.NAME_THE_PEOPLES_PARTY);
    put(Party.PRO_PARK_NEW_PARTY, Presets.NAME_PRO_PARK_NEW_PARTY);
    put(Party.OPEN_DEMOCRATIC_PARTY, Presets.NAME_OPEN_DEMOCRATIC_PARTY);
  });
  public static String toString(Party party) {
    return partyEnumToString.get(party);
  }
  public static Party[] parties = {
    Party.DEMOCRATIC_PARTY_OF_KOREA,
    Party.DEMOCRATIC_CITIZENS_PARTY,
    Party.UNITED_FUTURE_PARTY,
    Party.FUTURE_KOREA_PARTY,
    Party.MINSAENGDANG,
    Party.JUSTICE_PARTY,
    Party.OUR_REPUBLICAN_PARTY,
    Party.THE_PEOPLES_PARTY_OF_SOUTH_KOREA,
    Party.KOREAN_ECONOMIC_PARTY,
    Party.THE_PEOPLES_PARTY,
    Party.PRO_PARK_NEW_PARTY,
    Party.OPEN_DEMOCRATIC_PARTY,
  };
  public static String[] getTotalString() {
    return partyEnumToString.values().toArray(new String[0]);
  }

  public static Map<Party, Struct8Values> partyEnumTo8Values = new HashMap<Party,
Struct8Values>() {{
    put(Party.DEMOCRATIC_PARTY_OF_KOREA, Presets.VALUES_DEMOCRATIC_PARTY_OF_KOREA);
    put(Party.UNITED_FUTURE_PARTY, Presets.VALUES_UNITED_FUTURE_PARTY);
    put(Party.MINSAENGDANG, Presets.VALUES_MINSAENGDANG);
    put(Party.JUSTICE_PARTY, Presets.VALUES_JUSTICE_PARTY);
    put(Party.OUR_REPUBLICAN_PARTY, Presets.VALUES_OUR_REPUBLICAN_PARTY);
    put(Party.THE_PEOPLES_PARTY_OF_SOUTH_KOREA,
Presets.VALUES_THE_PEOPLES_PARTY_OF_SOUTH_KOREA);
    put(Party.KOREAN_ECONOMIC_PARTY, Presets.VALUES_KOREAN_ECONOMIC_PARTY);
    put(Party.THE_PEOPLES_PARTY, Presets.VALUES_THE_PEOPLES_PARTY);
    put(Party.PRO_PARK_NEW_PARTY, Presets.VALUES_PRO_PARK_NEW_PARTY);
    put(Party.OPEN_DEMOCRATIC_PARTY, Presets.VALUES_OPEN_DEMOCRATIC_PARTY);
  }};

  public static Struct8Values to8Values(Party party) {
    return partyEnumTo8Values.get(party);
  }
}

```

```
src/main/java/me.jonghyeon.election_result_8values/commons/Presets.java
```

```

package me.jonghyeon.election_result_8values.commons;

import me.jonghyeon.election_result_8values.models.Struct8Values;

public class Presets {
  public static String NAME_DEMOCRATIC_PARTY_OF_KOREA = "더불어민주당";
  public static String NAME_DEMOCRATIC_CITIZENS_PARTY = "더불어시민당";
  public static String NAME_UNITED_FUTURE_PARTY = "미래통합당";
  public static String NAME_FUTURE_KOREA_PARTY = "미래한국당";
  public static String NAME_MINSAENGDANG = "민생당";
  public static String NAME_JUSTICE_PARTY = "정의당";
}

```

```

public static String NAME_OUR_REPUBLICAN_PARTY = "우리공화당";
public static String NAME_THE_PEOPLES_PARTY_OF_SOUTH_KOREA = "민중당";
public static String NAME_KOREAN_ECONOMIC_PARTY = "한국경제당";
public static String NAME_THE_PEOPLES_PARTY = "국민의당";
public static String NAME_PRO_PARK_NEW_PARTY = "친박신당";
public static String NAME_OPEN_DEMOCRATIC_PARTY = "열린민주당";

public static Struct8Values VALUES_DEMOCRATIC_PARTY_OF_KOREA = new
Struct8Values(-0.6f, +0.6f, -0.5f, +0.5f);
public static Struct8Values VALUES_UNITED_FUTURE_PARTY = new
Struct8Values(+0.6f, -0.2f, +0.2f, -0.4f);
public static Struct8Values VALUES_MINSAENGDANG = new
Struct8Values(0.0f, -0.1f, -0.1f, +0.1f);
public static Struct8Values VALUES_JUSTICE_PARTY = new
Struct8Values(-0.9f, +0.7f, -0.7f, +0.8f);
public static Struct8Values VALUES_OUR_REPUBLICAN_PARTY = new
Struct8Values(+0.8f, +0.8f, +0.7f, +0.8f);
public static Struct8Values VALUES_THE_PEOPLES_PARTY_OF_SOUTH_KOREA = new
Struct8Values(-0.9f, +0.7f, -0.7f, +0.8f);
public static Struct8Values VALUES_KOREAN_ECONOMIC_PARTY = new
Struct8Values(+0.8f, 0.0f, +0.3f, -0.2f);
public static Struct8Values VALUES_THE_PEOPLES_PARTY = new
Struct8Values(+0.1f, +0.2f, -0.1f, +0.1f);
public static Struct8Values VALUES_PRO_PARK_NEW_PARTY = new
Struct8Values(+0.8f, +0.8f, +0.8f, +0.8f);
public static Struct8Values VALUES_OPEN_DEMOCRATIC_PARTY = new
Struct8Values(-0.7f, +0.6f, -0.6f, +0.6f);
}

```

```
build.gradle.kts
```

```

import com.github.jengelman.gradle.plugins.shadow.tasks.ShadowJar

plugins {
    id("java")
    id("com.github.johnrengelman.shadow") version "8.1.1"
}

group = "me.jonghyeon"
version = "1.0-SNAPSHOT"

repositories {
    mavenCentral()
}

dependencies {
    implementation("org.apache.hadoop:hadoop-common:3.3.6")
    implementation("org.apache.hadoop:hadoop-mapreduce-client-core:3.3.6")

    implementation("org.apache.poi:poi:5.2.5")
    implementation("org.apache.poi:poi-ooxml:5.2.5")

    implementation("org.apache.spark:spark-core_2.12:3.5.0")
    implementation("org.apache.spark:spark-sql_2.12:3.5.0")

    implementation("io.github.cdimascio:dotenv-java:2.3.2")

    testImplementation(platform("org.junit:junit-bom:5.10.0"))
    testImplementation("org.junit.jupiter:junit-jupiter")
}

```

```
tasks.test {
    useJUnitPlatform()
}

java {
    toolchain {
        languageVersion.set(JavaLanguageVersion.of(8))
    }
}

task<Exec>("cloneRequiredResources") {
    commandLine("git", "clone", "https://github.com/southkorea/southkorea-maps")
}

tasks.withType<ShadowJar> {
    manifest {
        attributes["Main-Class"] = "me.jonghyeon.election_result_8values.Main"
    }
    mergeServiceFiles()
}

transform(com.github.jengelman.gradle.plugins.shadow.transformers.ServiceFileTransformer::class.java) {
    setProperty("zip64", true)
}
```

지역별 8values 분석 결과 표

Region	Economic	Diplomatic	Civil	Society
강원도,강릉시,강릉시	-0.1906	0.2409	-0.1894	0.2023
강원도,고성군,고성군	-0.054	0.1692	-0.1224	0.0991
강원도,고성군,속초시인제군고성군양양군	0.084	0.144	-0.101	-0.013
강원도,동해시,동해시	-0.1029	0.1281	-0.0939	0.1365
강원도,동해시,동해시태백시삼척시정선군	0.0822	0.1452	-0.1021	-0.0116
강원도,삼척시,동해시태백시삼척시정선군	0.0565	0.1623	-0.117	0.0076
강원도,삼척시,삼척시	-0.0889	0.1184	-0.0806	0.1276
강원도,속초시,속초시	-0.1122	0.1299	-0.1021	0.1385
강원도,속초시,속초시인제군고성군양양군	0.0487	0.1675	-0.1216	0.0135
강원도,양구군,양구군	-0.0975	0.1205	-0.0896	0.1319
강원도,양구군,춘천시철원군화천군양구군을	0.0106	0.1929	-0.1438	0.042
강원도,양양군,속초시인제군고성군양양군	0.0938	0.1375	-0.0953	-0.0203
강원도,양양군,양양군	-0.0749	0.108	-0.0699	0.1152
강원도,영월군,영월군	-0.0868	0.1215	-0.0795	0.1314
강원도,영월군,홍천군횡성군영월군평창군	0.1192	0.1206	-0.0805	-0.0394
강원도,원주시,원주시	-0.1258	0.1419	-0.1145	0.1515
강원도,원주시,원주시갑	-0.0404	0.2334	-0.1714	0.0879
강원도,원주시,원주시을	-0.0636	0.2448	-0.1849	0.1057
강원도,인제군,속초시인제군고성군양양군	0.0093	0.1938	-0.1445	0.043
강원도,인제군,인제군	-0.092	0.1173	-0.0868	0.1275
강원도,정선군,동해시태백시삼척시정선군	0.048	0.168	-0.122	0.014
강원도,정선군,정선군	-0.0849	0.1179	-0.0768	0.1279
강원도,철원군,철원군	-0.0914	0.1194	-0.0852	0.1284
강원도,철원군,춘천시철원군화천군양구군을	0.1004	0.1331	-0.0915	-0.0253
강원도,춘천시,춘천시	-0.1462	0.1519	-0.1299	0.1634
강원도,춘천시,춘천시철원군화천군양구군갑	-0.0773	0.2475	-0.1961	0.1111
강원도,춘천시,춘천시철원군화천군양구군을	0.0012	0.1992	-0.1493	0.0491
강원도,춘천시사선거구,춘천시	-0.001	0.2007	-0.1506	0.0508
강원도,태백시,동해시태백시삼척시정선군	0.0836	0.1442	-0.1012	-0.0127
강원도,태백시,태백시	-0.096	0.1184	-0.0857	0.1279
강원도,평창군,평창군	-0.0915	0.1199	-0.0825	0.1301
강원도,평창군,홍천군횡성군영월군평창군	0.005	0.1967	-0.1471	0.0463
강원도,홍천군,홍천군	-0.118	0.1421	-0.1038	0.1552
강원도,홍천군,홍천군횡성군영월군평창군	0.0928	0.1381	-0.0959	-0.0196
강원도,홍천군가선거구,홍천군	0.0156	0.1896	-0.1409	0.0383
강원도,화천군,춘천시철원군화천군양구군을	0.0493	0.1671	-0.1213	0.013
강원도,화천군,화천군	-0.0919	0.1189	-0.0864	0.1288
강원도,횡성군,홍천군횡성군영월군평창군	0.0805	0.1464	-0.1031	-0.0103
강원도,횡성군,횡성군	-0.0573	0.1733	-0.1271	0.1001
경기도,가평군,가평군	-0.0817	0.1137	-0.076	0.1211
경기도,가평군,포천시가평군	0.0659	0.1544	-0.112	0.0018
경기도,고양시덕양구,고양시갑	-0.3205	0.3741	-0.3468	0.32
경기도,고양시덕양구,고양시덕양구	-0.206	0.1966	-0.1778	0.2124
경기도,고양시덕양구,고양시을	-0.1853	0.314	-0.2605	0.1962
경기도,고양시일산동구,고양시병	-0.0499	0.2389	-0.1772	0.0941
경기도,고양시일산동구,고양시일산동구	-0.1519	0.1578	-0.1381	0.1662
경기도,고양시일산서구,고양시일산서구	-0.1502	0.1545	-0.1378	0.1622
경기도,고양시일산서구,고양시정	-0.049	0.2327	-0.1786	0.0867
경기도,과천시,과천시	-0.141	0.1488	-0.1348	0.1585
경기도,과천시,의왕시과천시	0.0003	0.1705	-0.1468	0.0572
경기도,광명시,광명시	-0.1316	0.1408	-0.1229	0.1514

Region	Economic	Diplomatic	Civil	Society
경기도,광명시,광명시갑	-0.0684	0.2281	-0.187	0.1042
경기도,광명시,광명시을	-0.2116	0.3411	-0.2734	0.2087
경기도,광주시,광주시	-0.118	0.1298	-0.1081	0.1385
경기도,광주시,광주시갑	-0.0787	0.2525	-0.1959	0.109
경기도,광주시,광주시을	-0.0836	0.2558	-0.1988	0.1127
경기도,구리시,구리시	-0.1144	0.2139	-0.1631	0.1456
경기도,군포시,군포시	-0.1289	0.2154	-0.1764	0.1483
경기도,김포시,김포시	-0.1297	0.1383	-0.1202	0.1463
경기도,김포시,김포시갑	-0.0923	0.2615	-0.2039	0.1192
경기도,김포시,김포시을	-0.0541	0.2332	-0.1811	0.091
경기도,남양주시,남양주시	-0.1288	0.1393	-0.1179	0.1488
경기도,남양주시,남양주시갑	-0.1129	0.2719	-0.2127	0.1423
경기도,남양주시,남양주시병	-0.0347	0.2209	-0.1708	0.0776
경기도,남양주시,남양주시을	-0.1811	0.3207	-0.2556	0.1858
경기도,동두천시,동두천시	-0.093	0.1196	-0.0863	0.1282
경기도,동두천시,동두천시연천군	0.0289	0.1807	-0.1331	0.0283
경기도,부천시,부천시	-0.1365	0.1439	-0.1239	0.1538
경기도,부천시,부천시갑	-0.1686	0.3087	-0.2493	0.1792
경기도,부천시,부천시병	-0.2219	0.3418	-0.281	0.221
경기도,부천시,부천시을	-0.1349	0.2849	-0.2299	0.1549
경기도,부천시,부천시정	-0.169	0.307	-0.25	0.1811
경기도,성남시라선거구,성남시수정구	-0.6858	0.6286	-0.5572	0.5858
경기도,성남시분당구,성남시분당구	-0.1272	0.1411	-0.1223	0.1449
경기도,성남시분당구,성남시분당구갑	0.0075	0.195	-0.1456	0.0444
경기도,성남시분당구,성남시분당구을	-0.0348	0.2243	-0.1697	0.0824
경기도,성남시수정구,성남시수정구	-0.1504	0.2298	-0.187	0.1665
경기도,성남시제7선거구,성남시분당구	-0.0136	0.2011	-0.1599	0.0662
경기도,성남시중원구,성남시중원구	-0.1122	0.2024	-0.1637	0.1363
경기도,수원시권선구,수원시권선구	-0.1428	0.1505	-0.1313	0.1597
경기도,수원시권선구,수원시무	-0.1525	0.2958	-0.2404	0.1688
경기도,수원시권선구,수원시을	-0.1329	0.2886	-0.2275	0.1497
경기도,수원시영통구,수원시영통구	-0.1456	0.1542	-0.1371	0.1605
경기도,수원시영통구,수원시정	-0.1275	0.284	-0.2246	0.1464
경기도,수원시장안구,수원시갑	-0.1157	0.2771	-0.2175	0.1368
경기도,수원시장안구,수원시장안구	-0.1264	0.1393	-0.1172	0.1477
경기도,수원시팔달구,수원시병	-0.0909	0.2618	-0.2023	0.1271
경기도,수원시팔달구,수원시팔달구	-0.1229	0.1387	-0.1136	0.1475
경기도,시흥시,시흥시	-0.1396	0.1452	-0.1266	0.1554
경기도,시흥시,시흥시갑	-0.0641	0.24	-0.1881	0.1002
경기도,시흥시,시흥시을	-0.2144	0.343	-0.2751	0.2108
경기도,안산시단원구,안산시단원구	-0.1228	0.1339	-0.1135	0.1435
경기도,안산시단원구,안산시단원구갑	-0.1033	0.267	-0.2107	0.1289
경기도,안산시단원구,안산시단원구을	-0.0148	0.2219	-0.1546	0.0755
경기도,안산시상록구,안산시상록구	-0.1247	0.1359	-0.1144	0.1458
경기도,안산시상록구,안산시상록구갑	-0.1342	0.2949	-0.2263	0.161
경기도,안산시상록구,안산시상록구을	-0.1009	0.2673	-0.2089	0.1257
경기도,안성시,안성시	-0.0525	0.1895	-0.1445	0.0926
경기도,안양시동안구,안양시동안구	-0.1495	0.1534	-0.1366	0.1628
경기도,안양시동안구,안양시동안구갑	-0.1293	0.2819	-0.2265	0.1502
경기도,안양시동안구,안양시동안구을	-0.1019	0.2629	-0.2099	0.1291
경기도,안양시만안구,안양시만안구	-0.1114	0.2019	-0.1637	0.1357
경기도,양주시,양주시	-0.1365	0.2235	-0.1758	0.1562
경기도,양평군,양평군	-0.1118	0.1314	-0.0996	0.141
경기도,양평군,여주시양평군	0.0847	0.1458	-0.0996	-0.0042
경기도,여주시,여주시	-0.1057	0.1297	-0.0937	0.1404
경기도,여주시,여주시양평군	0.0438	0.175	-0.1227	0.0293
경기도,연천군,동두천시연천군	0.1085	0.1276	-0.0867	-0.0314
경기도,연천군,연천군	-0.0753	0.1126	-0.0717	0.121
경기도,오산시,오산시	-0.109	0.1982	-0.1624	0.1329
경기도,용인시기흥구,용인시기흥구	-0.1341	0.1415	-0.1251	0.148
경기도,용인시기흥구,용인시을	-0.1271	0.2826	-0.2238	0.1457
경기도,용인시기흥구,용인시정	-0.0737	0.249	-0.1927	0.1094
경기도,용인시수지구,용인시병	-0.0187	0.2125	-0.1609	0.064

Region	Economic	Diplomatic	Civil	Society
경기도,용인시수지구,용인시수지구	-0.1295	0.1411	-0.1239	0.1447
경기도,용인시처인구,용인시갑	0.047	0.1687	-0.1226	0.0148
경기도,용인시처인구,용인시처인구	-0.1109	0.1266	-0.1027	0.1348
경기도,의왕시,의왕시	-0.1294	0.1332	-0.1236	0.15
경기도,의왕시,의왕시과천시	-0.0812	0.1969	-0.1891	0.1222
경기도,의정부시,의정부시	-0.1183	0.1553	-0.1067	0.1647
경기도,의정부시,의정부시갑	-0.0997	0.2665	-0.2082	0.1248
경기도,의정부시,의정부시을	-0.1404	0.2893	-0.233	0.1586
경기도,이천시,이천시	-0.0284	0.1482	-0.1117	0.072
경기도,파주시,파주시	-0.1353	0.1442	-0.1225	0.1538
경기도,파주시,파주시갑	-0.142	0.2947	-0.2328	0.1565
경기도,파주시,파주시을	-0.0576	0.2384	-0.1836	0.0932
경기도,평택시,평택시	-0.121	0.1334	-0.1114	0.1421
경기도,평택시,평택시갑	-0.0149	0.2099	-0.1587	0.0612
경기도,평택시,평택시을	-0.0019	0.1997	-0.1515	0.0526
경기도,평택시나선거구,평택시	-0.0742	0.2454	-0.1943	0.1087
경기도,포천시,포천시	-0.0922	0.1215	-0.0846	0.1303
경기도,포천시,포천시가평군	-0.019	0.2106	-0.1616	0.0658
경기도,하남시,하남시	-0.1256	0.2066	-0.1683	0.1436
경기도,화성시,화성시	-0.1364	0.1481	-0.1268	0.1559
경기도,화성시,화성시갑	-0.0479	0.2302	-0.1783	0.0872
경기도,화성시,화성시병	-0.1811	0.3207	-0.2556	0.1858
경기도,화성시,화성시을	-0.1807	0.3205	-0.2554	0.1855
경상남도,거제시,거제시	-0.0233	0.1416	-0.1087	0.0691
경상남도,거창군,거창군	-0.0772	0.1211	-0.0745	0.1267
경상남도,거창군,산청군함양군거창군합천군	0.2065	0.0724	-0.0248	-0.0873
경상남도,고성군,고성군	-0.0761	0.1046	-0.0688	0.1138
경상남도,고성군,통영시고성군	0.2014	0.0794	-0.0279	-0.0847
경상남도,고성군다선거구,고성군	0.26	0.0267	0.0017	-0.145
경상남도,김해시,김해시	-0.1303	0.1345	-0.1173	0.1448
경상남도,김해시,김해시갑	-0.059	0.2365	-0.1851	0.0964
경상남도,김해시,김해시을	-0.0954	0.2583	-0.2069	0.1255
경상남도,남해군,남해군	-0.0714	0.1065	-0.0643	0.1152
경상남도,남해군,사천시남해군하동군	0.2403	0.0398	-0.0098	-0.1303
경상남도,밀양시,밀양시	-0.0746	0.1074	-0.0688	0.1149
경상남도,밀양시,밀양시의령군함안군창녕군	0.2257	0.0495	-0.0184	-0.1193
경상남도,사천시,사천시	-0.0906	0.1125	-0.0824	0.1209
경상남도,사천시,사천시남해군하동군	0.0968	0.1355	-0.0936	-0.0226
경상남도,산청군,산청군	-0.0746	0.114	-0.0716	0.1208
경상남도,산청군,산청군함양군거창군합천군	0.1719	0.0907	-0.0461	-0.0657
경상남도,양산시,양산시	-0.1104	0.1219	-0.1009	0.1301
경상남도,양산시,양산시갑	0.0942	0.1372	-0.095	-0.0207
경상남도,양산시,양산시을	-0.0341	0.2198	-0.1706	0.0778
경상남도,의령군,밀양시의령군함안군창녕군	0.2631	0.0246	0.0035	-0.1473
경상남도,의령군,의령군	-0.0657	0.1148	-0.0587	0.125
경상남도,의령군나선거구,의령군	0.6	-0.2	0.2	-0.4
경상남도,진주시,진주시	-0.1042	0.1204	-0.0956	0.1283
경상남도,진주시,진주시갑	0.0944	0.1357	-0.0953	-0.0197
경상남도,진주시,진주시을	0.1736	0.0938	-0.0455	-0.0688
경상남도,진주시제3선거구,진주시	0.0801	0.1361	-0.1059	-0.0022
경상남도,창녕군,밀양시의령군함안군창녕군	0.2934	0.0044	0.0212	-0.1701
경상남도,창녕군,창녕군	-0.0571	0.0926	-0.0538	0.0996
경상남도,창원시마산합포구,창원시마산합포구	0.0529	0.0893	-0.0599	0.0077
경상남도,창원시마산회원구,창원시마산회원구	0.0071	0.1265	-0.0867	0.0487
경상남도,창원시성산구,창원시성산구	-0.1686	0.222	-0.2068	0.1924
경상남도,창원시의창구,창원시의창구	0.0016	0.1222	-0.0956	0.0485
경상남도,창원시진해구,창원시진해구	-0.0506	0.1604	-0.1251	0.0867
경상남도,통영시,통영시	-0.0791	0.1032	-0.0733	0.1105
경상남도,통영시,통영시고성군	0.1066	0.1498	-0.0807	-0.0051

Region	Economic	Diplomatic	Civil	Society
경상남도,하동군,사천시남해군하동군	0.1334	0.111	-0.0722	-0.0501
경상남도,하동군,하동군	-0.0845	0.1133	-0.0759	0.1242
경상남도,함안군,밀양시의령군함안군창녕군	0.1807	0.0796	-0.0446	-0.0855
경상남도,함안군,함안군	-0.0859	0.1121	-0.0783	0.1199
경상남도,함양군,산청군함양군거창군합천군	0.0862	0.1498	-0.0956	0.0024
경상남도,함양군,함양군	-0.0727	0.1218	-0.071	0.1283
경상남도,합천군,산청군함양군거창군합천군	0.3059	0.0057	0.0332	-0.1622
경상남도,합천군,합천군	-0.0462	0.1056	-0.0483	0.1107
경상북도,경산시,경산시	0.0816	0.0773	-0.0423	-0.0051
경상북도,경주시,경주시	0.0226	0.1031	-0.0739	0.0448
경상북도,고령군,고령군	-0.0394	0.0996	-0.0385	0.1045
경상북도,고령군,고령군성주군칠곡군	0.3651	-0.0434	0.063	-0.2239
경상북도,구미시,구미시	-0.0637	0.1038	-0.064	0.1071
경상북도,구미시,구미시갑	0.2247	0.0714	-0.0117	-0.0932
경상북도,구미시,구미시을	0.138	0.108	-0.0695	-0.0535
경상북도,구미시제6선거구,구미시	0.3362	-0.0241	0.0461	-0.2021
경상북도,군위군,군위군	-0.0114	0.0873	-0.0147	0.0904
경상북도,군위군,군위군의성군청송군영덕군	0.4063	-0.0708	0.087	-0.2547
경상북도,김천시,김천시	0.1472	0.0344	-0.002	-0.0549
경상북도,문경시,문경시	-0.0433	0.0965	-0.0417	0.1009
경상북도,문경시,상주시문경시	0.3466	-0.031	0.0522	-0.2099
경상북도,봉화군,봉화군	-0.05	0.1107	-0.0461	0.1172
경상북도,봉화군,영주시영양군봉화군울진군	0.3149	-0.0099	0.0337	-0.1862
경상북도,상주시,상주시	0.1498	0.0392	0.0024	-0.0493
경상북도,상주시,상주시문경시	0.3438	-0.0292	0.0506	-0.2079
경상북도,성주군,고령군성주군칠곡군	0.3346	-0.0231	0.0452	-0.201
경상북도,성주군,성주군	-0.0465	0.1042	-0.0437	0.1099
경상북도,안동시,안동시	-0.0534	0.1032	-0.0547	0.1065
경상북도,안동시,안동시예천군	0.1468	0.1141	-0.0603	-0.0459
경상북도,안동시마선거구,안동시	0.1477	0.1015	-0.0639	-0.0608
경상북도,안동시제2선거구,안동시	0.2888	0.0074	0.0185	-0.1666
경상북도,영덕군,군위군의성군청송군영덕군	0.3212	-0.0141	0.0374	-0.1909
경상북도,영덕군,영덕군	-0.0236	0.0801	-0.0245	0.0847
경상북도,영양군,영양군	-0.0481	0.1109	-0.0448	0.1169
경상북도,영양군,영주시영양군봉화군울진군	0.3388	-0.0259	0.0476	-0.2041
경상북도,영주시,영주시	-0.0715	0.1113	-0.0665	0.1171
경상북도,영주시,영주시영양군봉화군울진군	0.1972	0.0685	-0.035	-0.0979
경상북도,영천시,영천시	-0.0353	0.0944	-0.0385	0.0972
경상북도,영천시,영천시청도군	0.2626	0.0249	0.0032	-0.1469
경상북도,예천군,안동시예천군	0.2819	0.028	0.0198	-0.1424
경상북도,예천군,예천군	-0.0424	0.0997	-0.0441	0.1032
경상북도,울릉군,울릉군	-0.057	0.095	-0.0554	0.0998
경상북도,울릉군,포항시남구울릉군	0.1983	0.0671	-0.0345	-0.0982
경상북도,울진군,영주시영양군봉화군울진군	0.364	-0.0427	0.0624	-0.223
경상북도,울진군,울진군	-0.0528	0.1057	-0.0523	0.1098
경상북도,울진군다선거구,울진군	0.6	-0.2	0.2	-0.4
경상북도,의성군,군위군의성군청송군영덕군	0.4046	-0.0698	0.086	-0.2535
경상북도,의성군,의성군	-0.0221	0.0905	-0.0221	0.0959
경상북도,청도군,영천시청도군	0.3619	-0.0412	0.0611	-0.2214
경상북도,청도군,청도군	-0.023	0.0865	-0.0279	0.0889
경상북도,청송군,군위군의성군청송군영덕군	0.348	-0.032	0.053	-0.211
경상북도,청송군,청송군	-0.0357	0.0942	-0.0349	0.0993
경상북도,칠곡군,고령군성주군칠곡군	0.216	0.056	-0.024	-0.112
경상북도,칠곡군,칠곡군	-0.0593	0.1044	-0.059	0.1081
경상북도,포항시남구,포항시남구	-0.0765	0.1052	-0.0733	0.1102
경상북도,포항시남구,포항시남구울릉군	0.1313	0.1114	-0.0737	-0.0476

Region	Economic	Diplomatic	Civil	Society
경상북도,포항시마선거구,포항시북구	0.3181	-0.0121	0.0356	-0.1886
경상북도,포항시북구,포항시북구	0.0537	0.0915	-0.0616	0.0106
경상북도,포항시제6선거구,포항시남구	0.1942	0.0705	-0.0367	-0.0957
광주광역시,광산구,광산구	-0.1885	0.1611	-0.1656	0.1842
광주광역시,광산구,광산구갑	-0.5431	0.5135	-0.4621	0.4702
광주광역시,광산구,광산구를	-0.5729	0.5526	-0.4819	0.4882
광주광역시,남구,남구	-0.1513	0.1334	-0.1395	0.1541
광주광역시,남구,동구남구갑	-0.4729	0.4517	-0.4152	0.4152
광주광역시,동구,동구	-0.1507	0.1359	-0.1386	0.1551
광주광역시,동구,동구남구를	-0.5345	0.5178	-0.4563	0.4586
광주광역시,북구,북구	-0.1689	0.1477	-0.1507	0.1674
광주광역시,북구,북구갑	-0.5625	0.5722	-0.4788	0.474
광주광역시,북구,북구를	-0.5497	0.525	-0.4664	0.473
광주광역시,서구,서구	-0.1686	0.1462	-0.1522	0.1681
광주광역시,서구,서구갑	-0.4975	0.4983	-0.4359	0.4304
광주광역시,서구,서구를	-0.4896	0.4636	-0.4264	0.4295
대구광역시,남구,남구	-0.042	0.1	-0.0472	0.1008
대구광역시,남구,중구남구	0.2378	0.0414	-0.0113	-0.1284
대구광역시,달서구,달서구	-0.044	0.1157	-0.0519	0.116
대구광역시,달서구,달서구갑	0.2287	0.0623	-0.01	-0.0996
대구광역시,달서구,달서구병	0.3	0.176	0.0845	0.0347
대구광역시,달서구,달서구를	0.2093	0.067	-0.0254	-0.0926
대구광역시,달성군,달성군	0.0928	0.0721	-0.0381	-0.0133
대구광역시,동구,동구	-0.0579	0.1062	-0.0622	0.108
대구광역시,동구,동구갑	0.2418	0.0361	-0.0096	-0.1294
대구광역시,동구,동구를	0.1883	0.0716	-0.0401	-0.0902
대구광역시,동구라선거구,동구	0.2723	0.0185	0.0088	-0.1542
대구광역시,동구마선거구,동구	0.1986	0.0676	-0.0341	-0.099
대구광역시,동구바선거구,동구	0.1388	0.1075	-0.069	-0.0541
대구광역시,동구제3선거구,동구	0.2599	0.0267	0.0016	-0.1449
대구광역시,동구제4선거구,동구	0.1488	0.1008	-0.0632	-0.0616
대구광역시,북구,북구	-0.0647	0.1104	-0.0689	0.1117
대구광역시,북구,북구갑	0.1607	0.1002	-0.0535	-0.0545
대구광역시,북구,북구를	0.1384	0.1038	-0.0703	-0.0508
대구광역시,북구바선거구,북구	0.2052	0.0632	-0.0303	-0.1039
대구광역시,북구아선거구,북구	0.2775	0.015	0.0119	-0.1581
대구광역시,서구,서구	0.1235	0.05	-0.0131	-0.0311
대구광역시,수성구,수성구	-0.0607	0.1046	-0.0676	0.1032
대구광역시,수성구,수성구갑	0.131	0.1177	-0.0715	-0.0423
대구광역시,수성구,수성구를	0.1094	0.1271	-0.0862	-0.032
대구광역시,중구,중구	-0.0543	0.1034	-0.061	0.103
대구광역시,중구,중구남구	0.2068	0.0622	-0.0294	-0.1051
대전광역시,대덕구,대덕구	-0.0625	0.1709	-0.1327	0.0982
대전광역시,동구,동구	-0.0574	0.1696	-0.1296	0.0956
대전광역시,서구,서구	-0.1254	0.1405	-0.1181	0.1474
대전광역시,서구,서구갑	-0.0727	0.2485	-0.1924	0.1045
대전광역시,서구,서구를	-0.0911	0.2698	-0.2001	0.1291
대전광역시,서구나선거구,서구	-0.0502	0.2334	-0.1793	0.0876
대전광역시,서구바선거구,서구	-0.0723	0.2482	-0.1922	0.1042
대전광역시,서구제6선거구,서구	-0.0583	0.2388	-0.184	0.0937
대전광역시,유성구,유성구	-0.1509	0.1588	-0.1405	0.1661
대전광역시,유성구,유성구갑	-0.1101	0.2755	-0.2133	0.1393
대전광역시,유성구,유성구를	-0.1717	0.3078	-0.2518	0.1838
대전광역시,유성구다선거구,유성구	-0.0766	0.2511	-0.1947	0.1075
대전광역시,중구,중구	-0.0533	0.1673	-0.1273	0.0926
부산광역시,강서구,강서구	-0.1097	0.1217	-0.1026	0.1283
부산광역시,강서구,북구강서구를	0.0398	0.1716	-0.1273	0.0216
부산광역시,금정구,금정구	-0.0087	0.1283	-0.0988	0.0574
부산광역시,기장군,기장군	-0.0326	0.149	-0.1129	0.0744
부산광역시,남구,남구	-0.097	0.1134	-0.0913	0.1196
부산광역시,남구,남구갑	0.0416	0.169	-0.1266	0.0213
부산광역시,남구,남구를	-0.0061	0.204	-0.1535	0.0545
부산광역시,남구제2선거구,남구	0.0752	0.1499	-0.1061	-0.0064

Region	Economic	Diplomatic	Civil	Society
부산광역시,동구,동구	-0.0742	0.106	-0.072	0.1117
부산광역시,동구,서구동구	0.0975	0.1476	-0.0889	-0.0081
부산광역시,동래구,동래구	-0.0269	0.1427	-0.1099	0.069
부산광역시,부산진구,부산진구	-0.0874	0.1059	-0.0829	0.1123
부산광역시,부산진구,부산진구갑	0.0272	0.1806	-0.1339	0.0298
부산광역시,부산진구,부산진구를	0.0737	0.1509	-0.107	-0.0053
부산광역시,북구,북구	-0.0852	0.1017	-0.0795	0.1093
부산광역시,북구,북구강서구갑	-0.0088	0.2058	-0.1551	0.0566
부산광역시,사상구,사상구	-0.0179	0.1424	-0.1015	0.0664
부산광역시,사하구,사하구	-0.0907	0.1087	-0.0854	0.1152
부산광역시,사하구,사하구갑	-0.0003	0.2002	-0.1502	0.0502
부산광역시,사하구,사하구를	0.1178	0.1263	-0.0795	-0.0304
부산광역시,사하구가선거구,사하구	0.0818	0.1455	-0.1023	-0.0113
부산광역시,서구,서구	-0.0765	0.1036	-0.0735	0.1089
부산광역시,서구,서구동구	0.0964	0.1467	-0.09	-0.0093
부산광역시,수영구,수영구	0.0087	0.1175	-0.0877	0.0424
부산광역시,연제구,연제구	-0.0327	0.1464	-0.1136	0.0729
부산광역시,영도구,영도구	-0.0835	0.1039	-0.0772	0.1113
부산광역시,영도구,중구영도구	0.0365	0.1757	-0.1287	0.0226
부산광역시,중구,중구	-0.0081	0.1329	-0.0924	0.06
부산광역시,중구,중구영도구	0.0773	0.1485	-0.1049	-0.008
부산광역시,해운대구,해운대구	-0.0874	0.1123	-0.085	0.1161
부산광역시,해운대구,해운대구갑	0.1475	0.1149	-0.0594	-0.0448
부산광역시,해운대구,해운대구를	0.0464	0.169	-0.1229	0.0152
서울특별시,강남구,강남구	-0.088	0.1109	-0.0904	0.1121
서울특별시,강남구,강남구갑	0.1193	0.1165	-0.0798	-0.0388
서울특별시,강남구,강남구병	0.2004	0.0718	-0.031	-0.0939
서울특별시,강남구,강남구를	0.0321	0.1746	-0.1306	0.0266
서울특별시,강동구,강동구	-0.1199	0.1344	-0.1137	0.1406
서울특별시,강동구,강동구갑	-0.0189	0.2126	-0.161	0.0641
서울특별시,강동구,강동구를	-0.0966	0.2617	-0.2071	0.1245
서울특별시,강북구,강북구	-0.1208	0.1348	-0.1119	0.1438
서울특별시,강북구,강북구갑	-0.1218	0.2795	-0.2215	0.1427
서울특별시,강북구,강북구를	-0.1763	0.3175	-0.2528	0.1822
서울특별시,강북구나선거구,강북구	-0.1098	0.2732	-0.214	0.1323
서울특별시,강서구,강서구	-0.127	0.1412	-0.1191	0.1486
서울특별시,강서구,강서구갑	-0.108	0.272	-0.213	0.131
서울특별시,강서구,강서구병	-0.153	0.3005	-0.2396	0.1659
서울특별시,강서구,강서구를	-0.0725	0.2588	-0.1887	0.1169
서울특별시,관악구,관악구	-0.1326	0.1467	-0.126	0.1566
서울특별시,관악구,관악구갑	-0.56	0.5962	-0.4639	0.5254
서울특별시,관악구,관악구를	-0.0761	0.2498	-0.1921	0.1147
서울특별시,광진구,광진구	-0.1264	0.1352	-0.1184	0.1436
서울특별시,광진구,광진구갑	-0.1068	0.2624	-0.2123	0.1336
서울특별시,광진구,광진구를	-0.0116	0.2077	-0.1568	0.0587
서울특별시,구로구,구로구	-0.131	0.1416	-0.1207	0.1504
서울특별시,구로구,구로구갑	-0.1341	0.2882	-0.2281	0.1605
서울특별시,구로구,구로구를	-0.1195	0.2797	-0.2197	0.1396
서울특별시,금천구,금천구	-0.106	0.1953	-0.1556	0.1313
서울특별시,노원구,노원구	-0.1376	0.1502	-0.1286	0.158
서울특별시,노원구,노원구갑	-0.1409	0.2899	-0.2332	0.1586
서울특별시,노원구,노원구병	-0.0692	0.2441	-0.1909	0.1035
서울특별시,노원구,노원구를	-0.156	0.304	-0.241	0.167
서울특별시,도봉구,도봉구	-0.1271	0.1393	-0.1172	0.1481
서울특별시,도봉구,도봉구갑	-0.1216	0.2761	-0.2221	0.1448
서울특별시,도봉구,도봉구를	-0.0413	0.2275	-0.1741	0.081
서울특별시,도봉구가선거구,도봉구	-0.0738	0.2492	-0.1931	0.1054
서울특별시,동대문구,동대문구	-0.1186	0.1331	-0.1129	0.1418
서울특별시,동대문구,동대문구갑	-0.0732	0.2422	-0.1922	0.1068
서울특별시,동대문구,동대문구를	-0.0719	0.2467	-0.1922	0.1048
서울특별시,동대문구가선거구,동대문구	-0.103	0.2686	-0.2101	0.1272
서울특별시,동작구,동작구	-0.1296	0.143	-0.1218	0.15
서울특별시,동작구,동작구갑	-0.0814	0.2531	-0.1978	0.1119

Region	Economic	Diplomatic	Civil	Society
서울특별시, 동작구, 동작구	-0.0606	0.2379	-0.186	0.0973
서울특별시, 마포구, 마포구	-0.1522	0.1597	-0.1406	0.1675
서울특별시, 마포구, 마포구갑	-0.0755	0.2503	-0.194	0.1066
서울특별시, 마포구, 마포구를	-0.1796	0.3108	-0.257	0.1914
서울특별시, 서대문구, 서대문구	-0.1369	0.1488	-0.1276	0.1568
서울특별시, 서대문구, 서대문구갑	-0.0737	0.2523	-0.1918	0.1114
서울특별시, 서대문구, 서대문구를	-0.1393	0.2929	-0.2313	0.1545
서울특별시, 서대문구마선거구, 서대문구	-0.1412	0.2912	-0.2331	0.1581
서울특별시, 서초구, 서초구	-0.0975	0.118	-0.0984	0.1193
서울특별시, 서초구, 서초구갑	0.1603	0.0931	-0.0565	-0.0702
서울특별시, 서초구, 서초구를	0.0573	0.1594	-0.1162	0.0075
서울특별시, 성동구, 성동구	-0.1283	0.141	-0.1207	0.1482
서울특별시, 성동구, 중구성동구갑	-0.1154	0.2727	-0.2184	0.1397
서울특별시, 성북구, 성북구	-0.1299	0.142	-0.1217	0.1504
서울특별시, 성북구, 성북구갑	-0.1437	0.2899	-0.2328	0.1587
서울특별시, 성북구, 성북구를	-0.1397	0.2914	-0.2319	0.156
서울특별시, 송파구, 송파구	-0.1117	0.1268	-0.1089	0.1318
서울특별시, 송파구, 송파구갑	0.0224	0.185	-0.1369	0.0332
서울특별시, 송파구, 송파구병	-0.0538	0.2291	-0.1802	0.0915
서울특별시, 송파구, 송파구를	0.0051	0.196	-0.147	0.0513
서울특별시, 양천구, 양천구	-0.1219	0.1359	-0.1162	0.1417
서울특별시, 양천구, 양천구갑	-0.0391	0.2261	-0.1728	0.0793
서울특별시, 양천구, 양천구를	-0.0907	0.2661	-0.201	0.1248
서울특별시, 영등포구, 영등포구	-0.1263	0.1393	-0.1198	0.1468
서울특별시, 영등포구, 영등포구갑	-0.1487	0.2942	-0.238	0.1652
서울특별시, 영등포구, 영등포구를	-0.0323	0.2173	-0.1681	0.0749
서울특별시, 용산구, 용산구	-0.0716	0.1735	-0.1411	0.1053
서울특별시, 은평구, 은평구	-0.1383	0.1467	-0.1268	0.1561
서울특별시, 은평구, 은평구갑	-0.1792	0.316	-0.2539	0.185
서울특별시, 은평구, 은평구를	-0.1665	0.3065	-0.2483	0.1783
서울특별시, 종로구, 종로구	-0.1168	0.2103	-0.1664	0.1418
서울특별시, 중구, 중구	-0.1143	0.1326	-0.1099	0.1381
서울특별시, 중구, 중구성동구를	-0.025	0.2167	-0.1646	0.0688
서울특별시, 중랑구, 중랑구	-0.1192	0.1316	-0.1103	0.1412
서울특별시, 중랑구, 중랑구갑	-0.159	0.3001	-0.2433	0.1723
서울특별시, 중랑구, 중랑구를	-0.1393	0.2913	-0.2317	0.1557
세종특별자치시, 세종특별자치시, 세종특별자치시	-0.1587	0.1654	-0.1468	0.1739
세종특별자치시, 세종특별자치시, 세종특별자치시갑	-0.2006	0.3278	-0.2685	0.2049
세종특별자치시, 세종특별자치시, 세종특별자치시을	-0.1079	0.2674	-0.2122	0.1317
세종특별자치시, 세종특별자치시제9선거구, 세종특별자치시제9선거구	-0.1827	0.3218	-0.2566	0.187
울산광역시, 남구, 남구	-0.1105	0.1252	-0.1029	0.1332
울산광역시, 남구, 남구갑	0.0651	0.1499	-0.1109	0.0023
울산광역시, 남구, 남구를	0.1142	0.1239	-0.0834	-0.0356
울산광역시, 남구바선거구, 남구	0.0208	0.1861	-0.1379	0.0344
울산광역시, 동구, 동구	-0.2181	0.2612	-0.235	0.2347
울산광역시, 북구, 북구	-0.158	0.2296	-0.1969	0.1769
울산광역시, 북구가선거구, 북구	-0.1407	0.2771	-0.2363	0.1681
울산광역시, 울주군, 울주군	-0.0216	0.1431	-0.1075	0.0671
울산광역시, 중구, 중구	0.0126	0.1178	-0.085	0.042
인천광역시, 강화군, 강화군	-0.0688	0.1063	-0.0634	0.1138
인천광역시, 강화군, 중구강화군옹진군	0.16	0.1078	-0.0518	-0.0527
인천광역시, 계양구, 계양구	-0.1324	0.1404	-0.1204	0.1508
인천광역시, 계양구, 계양구갑	-0.1442	0.2962	-0.2341	0.1582
인천광역시, 계양구, 계양구를	-0.133	0.2869	-0.228	0.1511
인천광역시, 남동구, 남동구	-0.1422	0.1484	-0.1277	0.1594
인천광역시, 남동구, 남동구갑	-0.0575	0.2383	-0.1835	0.0931
인천광역시, 남동구, 남동구를	-0.1485	0.2939	-0.2379	0.1652
인천광역시, 동구, 동구	-0.1395	0.1483	-0.1215	0.1622
인천광역시, 동구, 동구미추홀구갑	-0.0867	0.2503	-0.2025	0.1207
인천광역시, 미추홀구, 동구미추홀구갑	-0.1167	0.2697	-0.2201	0.1436
인천광역시, 미추홀구, 동구미추홀구를	-0.296	0.3926	-0.3239	0.2756

Region	Economic	Diplomatic	Civil	Society
인천광역시, 미추홀구, 미추홀구	-0.1357	0.1461	-0.1208	0.1576
인천광역시, 부평구, 부평구	-0.1355	0.1447	-0.1231	0.155
인천광역시, 부평구, 부평구갑	-0.0872	0.2581	-0.2009	0.1154
인천광역시, 부평구, 부평구를	-0.1814	0.3139	-0.2576	0.1913
인천광역시, 서구, 서구	-0.1361	0.1423	-0.1243	0.152
인천광역시, 서구, 서구갑	-0.083	0.2532	-0.1989	0.1138
인천광역시, 서구, 서구를	-0.145	0.2966	-0.2346	0.1587
인천광역시, 연수구, 연수구	-0.1837	0.1794	-0.1616	0.1925
인천광역시, 연수구, 연수구갑	-0.0866	0.2577	-0.2005	0.115
인천광역시, 연수구, 연수구를	-0.1781	0.3002	-0.2585	0.1975
인천광역시, 옹진군, 옹진군	-0.0678	0.0955	-0.0649	0.1046
인천광역시, 옹진군, 중구강화군옹진군	0.1249	0.1296	-0.0728	-0.0283
인천광역시, 중구, 중구	-0.1294	0.1433	-0.1187	0.1522
인천광역시, 중구, 중구강화군옹진군	-0.0656	0.2564	-0.184	0.1142
전라남도, 강진군, 강진군	-0.1215	0.1043	-0.1202	0.1443
전라남도, 강진군, 고흥군보성군장흥군강진군	-0.3342	0.2899	-0.3228	0.3228
전라남도, 고흥군, 고흥군	-0.0936	0.0814	-0.0966	0.12
전라남도, 고흥군, 고흥군보성군장흥군강진군	-0.3865	0.3509	-0.3577	0.3577
전라남도, 곡성군, 곡성군	-0.1477	0.1413	-0.1321	0.1617
전라남도, 곡성군, 순천시광양시곡성군구례군을	-0.5588	0.5658	-0.4777	0.4742
전라남도, 광양시, 광양시	-0.1677	0.1544	-0.1479	0.1684
전라남도, 광양시, 순천시광양시곡성군구례군을	-0.5506	0.5596	-0.473	0.4685
전라남도, 구례군, 구례군	-0.1443	0.1396	-0.1302	0.156
전라남도, 구례군, 순천시광양시곡성군구례군을	-0.5572	0.5653	-0.4766	0.4726
전라남도, 나주시, 나주시	-0.1768	0.1641	-0.1536	0.1834
전라남도, 나주시, 나주시화순군	-0.6705	0.6235	-0.547	0.5705
전라남도, 담양군, 담양군	-0.1345	0.1246	-0.12	0.1423
전라남도, 담양군, 담양군함평군영광군장성군	-0.5675	0.5778	-0.476	0.4889
전라남도, 목포시, 목포시	-0.3025	0.2548	-0.2771	0.292
전라남도, 무안군, 무안군	-0.1711	0.1455	-0.1532	0.1745
전라남도, 무안군, 영암군무안군신안군	-0.4299	0.4209	-0.3898	0.3834
전라남도, 보성군, 고흥군보성군장흥군강진군	-0.3865	0.3509	-0.3577	0.3577
전라남도, 보성군, 보성군	-0.1202	0.1091	-0.1145	0.1397
전라남도, 순천시, 순천시	-0.1603	0.147	-0.1416	0.1616
전라남도, 순천시, 순천시광양시곡성군구례군갑	-0.5642	0.5648	-0.4805	0.4797
전라남도, 신안군, 신안군	-0.1222	0.1091	-0.1118	0.1354
전라남도, 신안군, 영암군무안군신안군	-0.4495	0.4439	-0.4029	0.3964
전라남도, 여수시, 여수시	-0.1446	0.1342	-0.1299	0.1497
전라남도, 여수시, 여수시갑	-0.5336	0.5558	-0.4613	0.4502
전라남도, 여수시, 여수시를	-0.5633	0.5695	-0.4801	0.4771
전라남도, 여수시나선거구, 여수시	-0.5113	0.4965	-0.4409	0.4409
전라남도, 영광군, 담양군함평군영광군장성군	-0.5437	0.5551	-0.4595	0.4765
전라남도, 영광군, 영광군	-0.1733	0.158	-0.151	0.1808
전라남도, 영암군, 영암군	-0.1755	0.1488	-0.1541	0.1796
전라남도, 영암군, 영암군무안군신안군	-0.4468	0.4467	-0.4021	0.3936
전라남도, 완도군, 완도군	-0.1021	0.0923	-0.0975	0.1191
전라남도, 완도군, 해남군완도군진도군	-0.4515	0.4268	-0.401	0.401
전라남도, 장성군, 담양군함평군영광군장성군	-0.56	0.5749	-0.4702	0.4879
전라남도, 장성군, 장성군	-0.1352	0.1262	-0.1215	0.1437
전라남도, 장흥군, 고흥군보성군장흥군강진군	-0.4145	0.3836	-0.3764	0.3764
전라남도, 장흥군, 장흥군	-0.1492	0.1309	-0.135	0.1636
전라남도, 진도군, 진도군	-0.122	0.1051	-0.1174	0.1427
전라남도, 진도군, 해남군완도군진도군	-0.4056	0.3732	-0.3704	0.3704
전라남도, 함평군, 담양군함평군영광군장성군	-0.5339	0.5484	-0.4522	0.4731
전라남도, 함평군, 함평군	-0.2922	0.2882	-0.2567	0.2733

Region	Economic	Diplomatic	Civil	Society
전라남도,해남군,해남군	-0.1398	0.1183	-0.13	0.1539
전라남도,해남군,해남군완도군진도군	-0.3811	0.3447	-0.3541	0.3541
전라남도,화순군,나주시화순군	-0.6429	0.6143	-0.5286	0.5429
전라남도,화순군,화순군	-0.1655	0.1526	-0.1435	0.1717
전라북도,고창군,고창군	-0.1423	0.1246	-0.1291	0.1528
전라북도,고창군,정읍시고창군	-0.4502	0.4252	-0.4001	0.4001
전라북도,군산시,군산시	-0.3392	0.3357	-0.2941	0.3017
전라북도,군산시마선거구,군산시	-0.6	0.6	-0.5	0.5
전라북도,김제시,김제시	-0.1811	0.1628	-0.1561	0.1841
전라북도,김제시,김제시부안군	-0.5779	0.5742	-0.4853	0.4853
전라북도,남원시,남원시	-0.16	0.1754	-0.1379	0.1928
전라북도,남원시,남원시임실군순창군	-0.6184	0.6061	-0.5123	0.5184
전라북도,무주군,무주군	-0.1406	0.1783	-0.1212	0.1986
전라북도,무주군,완주군진안군무주군장수군	-0.4851	0.5234	-0.433	0.4138
전라북도,부안군,김제시부안군	-0.5253	0.5129	-0.4502	0.4502
전라북도,부안군,부안군	-0.1594	0.1456	-0.1399	0.1675
전라북도,순창군,남원시임실군순창군	-0.6126	0.6042	-0.5084	0.5126
전라북도,순창군,순창군	-0.1642	0.1754	-0.1414	0.1983
전라북도,완주군,완주군	-0.1864	0.1876	-0.161	0.2095
전라북도,완주군,완주군진안군무주군장수군	-0.5155	0.5437	-0.4507	0.4366
전라북도,익산시,익산시	-0.1711	0.1496	-0.1514	0.1732
전라북도,익산시,익산시갑	-0.4602	0.4752	-0.4143	0.402
전라북도,익산시,익산시을	-0.5255	0.4935	-0.4503	0.4581
전라북도,임실군,남원시임실군순창군	-0.6148	0.6049	-0.5099	0.5148
전라북도,임실군,임실군	-0.1411	0.1673	-0.123	0.1864
전라북도,장수군,완주군진안군무주군장수군	-0.5181	0.5454	-0.4522	0.4386
전라북도,장수군,장수군	-0.1639	0.176	-0.1416	0.196
전라북도,전주시덕진구,전주시덕진구	-0.1867	0.1572	-0.1689	0.1881
전라북도,전주시덕진구,전주시병	-0.4043	0.3717	-0.3696	0.3696
전라북도,전주시마선거구,전주시덕진구	0	-0.1	-0.1	0.1
전라북도,전주시마선거구,전주시완산구	0	-0.1	-0.1	0.1
전라북도,전주시완산구,전주시갑	-0.6229	0.6076	-0.5152	0.5229
전라북도,전주시완산구,전주시완산구	-0.1936	0.1705	-0.1708	0.1905
전라북도,전주시완산구,전주시을	-0.4856	0.5043	-0.4323	0.4206
전라북도,정읍시,정읍시	-0.1554	0.1302	-0.1429	0.1645
전라북도,정읍시,정읍시고창군	-0.4021	0.3691	-0.368	0.368
전라북도,진안군,완주군진안군무주군장수군	-0.5288	0.5525	-0.4585	0.4466
전라북도,진안군,진안군	-0.3207	0.3297	-0.271	0.307
제주특별자치도,서귀포시,서귀포시	-0.1083	0.2059	-0.1632	0.1365
제주특별자치도,제주시,제주시	-0.1674	0.1682	-0.1466	0.1811
제주특별자치도,제주시,제주시갑	-0.1366	0.2907	-0.2291	0.167
제주특별자치도,제주시,제주시을	-0.1076	0.2691	-0.2134	0.1327
제주특별자치도,제주특별자치도 서귀포시 대정읍선거구,서귀포시	-0.6	0.6	-0.5	0.5
제주특별자치도,제주특별자치도 서귀포시 대천동·중문동·예래동선거구,서귀포시	-0.6	0.6	-0.5	0.5
제주특별자치도,제주특별자치도 서귀포시 동홍동선거구,서귀포시	-0.2244	0.3496	-0.2809	0.2183
충청남도,계룡시,계룡시	-0.1127	0.1268	-0.1096	0.1338
충청남도,계룡시,논산시계룡시금산군	-0.0442	0.2226	-0.1746	0.0843
충청남도,공주시,공주시	-0.0974	0.123	-0.0898	0.1322
충청남도,공주시,공주시부여군청양군	-0.0078	0.2031	-0.1542	0.0562
충청남도,금산군,금산군	-0.0834	0.1209	-0.0782	0.1313
충청남도,금산군,논산시계룡시금산군	0.0319	0.1736	-0.1306	0.027
충청남도,금산군나선거구,금산군	0.1257	0.1162	-0.0767	-0.0442
충청남도,논산시,논산시	-0.0974	0.123	-0.0908	0.1333
충청남도,논산시,논산시계룡시금산군	-0.043	0.2252	-0.1745	0.0828
충청남도,당진시,당진시	-0.1344	0.2206	-0.1698	0.1632
충청남도,보령시,보령시	-0.0966	0.1184	-0.0877	0.1281
충청남도,보령시,보령시서천군	0.0641	0.1573	-0.1126	0.002
충청남도,부여군,공주시부여군청양군	0.0439	0.1691	-0.1241	0.0173

Region	Economic	Diplomatic	Civil	Society
충청남도,부여군,부여군	-0.0876	0.1141	-0.0803	0.1252
충청남도,서산시,서산시	-0.1212	0.1344	-0.1087	0.1445
충청남도,서산시,서산시태안군	0.0145	0.1878	-0.1422	0.041
충청남도,서천군,보령시서천군	-0.0777	0.2518	-0.1953	0.1083
충청남도,서천군,서천군	-0.1026	0.1286	-0.0922	0.141
충청남도,아산시,아산시	-0.1317	0.1435	-0.12	0.1533
충청남도,아산시,아산시갑	0.0081	0.1946	-0.1453	0.0439
충청남도,아산시,아산시을	-0.1149	0.2766	-0.217	0.1362
충청남도,아산시가선거구,아산시	0.0127	0.1915	-0.1426	0.0405
충청남도,예산군,예산군	-0.1119	0.1411	-0.0974	0.1536
충청남도,예산군,홍성군예산군	0.087	0.142	-0.0992	-0.0153
충청남도,천안시동남구,천안시	0.039	0.174	-0.1273	0.0208
충청남도,천안시동남구,천안시갑	0.009	0.2115	-0.1381	0.0641
충청남도,천안시동남구,천안시동남구	-0.1279	0.1447	-0.118	0.1541
충청남도,천안시동남구,천안시병	-0.0817	0.25	-0.1987	0.1146
충청남도,천안시서북구,천안시	-0.0347	0.2231	-0.1703	0.076
충청남도,천안시서북구,천안시서북구	-0.1561	0.1639	-0.1434	0.1742
충청남도,천안시서북구,천안시을	-0.1936	0.3235	-0.2643	0.1994
충청남도,천안시제5선거구,천안시서북구	0.0082	0.1945	-0.1452	0.0439
충청남도,청양군,공주시부여군청양군	0.05	0.1652	-0.1206	0.0128
충청남도,청양군,청양군	-0.0749	0.1108	-0.0694	0.1213
충청남도,태안군,서산시태안군	0.08	0.145	-0.1037	-0.0088
충청남도,태안군,태안군	-0.0867	0.112	-0.0791	0.1209
충청남도,홍성군,홍성군	-0.1054	0.1361	-0.0965	0.145
충청남도,홍성군,홍성군예산군	0.04	0.1734	-0.1267	0.02
충청북도,괴산군,괴산군	-0.0881	0.1207	-0.079	0.1316
충청북도,괴산군,보은군옥천군영동군괴산군	0.1107	0.1262	-0.0854	-0.033
충청북도,단양군,단양군	-0.0807	0.1192	-0.0734	0.1285
충청북도,단양군,제천시단양군	0.1139	0.1241	-0.0836	-0.0354
충청북도,보은군,보은군	-0.0748	0.1304	-0.0693	0.1419
충청북도,보은군,보은군옥천군영동군괴산군	0.1074	0.1284	-0.0874	-0.0305
충청북도,보은군선거구,보은군	0.1077	0.1282	-0.0872	-0.0308
충청북도,영동군,보은군옥천군영동군괴산군	0.0892	0.1405	-0.098	-0.0169
충청북도,영동군,영동군	-0.1006	0.1313	-0.0889	0.1429
충청북도,영동군제1선거구,영동군	-0.0294	0.2065	-0.1704	0.0818
충청북도,옥천군,보은군옥천군영동군괴산군	0.0906	0.1396	-0.0971	-0.018
충청북도,옥천군,옥천군	-0.0992	0.1317	-0.0878	0.1438
충청북도,음성군,음성군	-0.1007	0.1279	-0.0922	0.1377
충청북도,음성군,증평군진천군음성군	0.0268	0.1821	-0.1344	0.0299
충청북도,제천시,제천시	-0.0977	0.126	-0.089	0.1354
충청북도,제천시,제천시단양군	0.0478	0.1681	-0.1221	0.0141
충청북도,증평군,증평군	-0.095	0.1179	-0.0884	0.1271
충청북도,증평군,증평군진천군음성군	-0.0257	0.2171	-0.165	0.0692
충청북도,진천군,증평군진천군음성군	-0.0573	0.2382	-0.1834	0.0929
충청북도,진천군,진천군	-0.1265	0.1419	-0.1143	0.1536
충청북도,청주시상당구,청주시상당구	-0.1097	0.1977	-0.1639	0.1385
충청북도,청주시서원구,청주시서원구	-0.0693	0.1722	-0.1388	0.1047
충청북도,청주시제10선거구,청주시청원구	-0.1201	0.2691	-0.2228	0.1483
충청북도,청주시청원구,청주시청원구	-0.1009	0.1974	-0.1595	0.1288
충청북도,청주시흥덕구,청주시흥덕구	-0.106	0.2011	-0.1619	0.1313
충청북도,충주시,충주시	-0.0197	0.1407	-0.1057	0.0681

Table 11: 제 21대 총선 8values 평가 전체 결과
표

참고 문헌

- [1] 나혜인, “이재명 "민주당, 진보 아냐...중도·보수로 오른쪽 말아야", YTN, Feb. 2025, Accessed: Jun. 04, 2025. [Online]. Available: https://www.ytn.co.kr/_ln/0101_202502182313004510
- [2] IDRlabs, “8 Values Political Test.” [Online]. Available: <https://www.idrlabs.com/8-values-political/test.php>
- [3] P. Bernardelle, L. Fröhling, S. Civelli, R. Lunardi, K. Roitero, and G. Demartini, “Mapping and influencing the political ideology of large language models using synthetic personas,” in Companion Proceedings of the ACM on Web Conference 2025, 2025, pp. 864–867.
- [4] T. Olah and L. Erdey, “Social Conformity to Bots,” Available at SSRN 5033554.
- [5] 중앙선거관리위원회, [Online]. Available: <https://www.data.go.kr/data/15025527/fileData.do>
- [6] 8values-ko, “ideologies.js.” 2025.
- [7] 더불어민주당, “제21대 총선 (국회의원선거) 정책 공약집.” Mar. 2020.
- [8] 참여연대 사회경제1팀, “[보도자료] 투표 전 꼭 봐야할 21대 총선 재벌개혁 민생 공약.” Apr. 2020.
- [9] 2020총선시민네트워크, “<2020총선넷> 21대 총선 정당 공약 한줄 평가!.” Apr. 08, 2020.
- [10] YouTube, “YouTube Video (Title not provided in content).”
- [11] 신김, “더불어민주당 제21대 총선 주요 공약.” 2020.
- [12] 2. 기후위기 비상행동, “[보도자료] 2020총선넷, 21대 총선 정당 공약 평가 기자회견.” Apr. 08, 2020.
- [13] 이슬기, “통합당 151개 총선공약 발표...“경제·공정·안전·공동체”, 연합뉴스, Mar. 2020.
- [14] 미래통합당, “대한민국을 재설계 미래로 재도약 ‘내 삶을 디자인 하다’ 제21대 국회의원선거 정책 공약집 발간.” [Online]. Available: https://www.peoplepowerparty.kr/file_download?attach_file_id=FILE_000000000133108&file_sn=0
- [15] 위키백과 기여자, “대한민국 제21대 국회의원 선거.” May 21, 2025.
- [16] 정의당 부산시당, “[21대총선 공약] 공약집 '2020 정의로운 대전환' 발간.” Mar. 25, 2020.
- [17] 중앙선거관리위원회, “제21대 국선 정당·후보자 정보자료 (35개 정당).” 2020.