

데이터 수집을 위한 프로그래밍

정치적 불안정 효과

'여소야대' 발생 원인과 해결방안을 위한 데이터 분석

정보통신전자공학부 202021303 김민구

경제학과 202021310 김태경

수학과 202120956 주인영

컴퓨터정보공학부 202221189 이현경

2024 데이터 수집을 위한 프로그래밍 팀 프로젝트

TABLE OF CONTENTS

목차 소개

01 | 주제 선정 동기

02 | 데이터 수집

03 | 데이터 시각화 및 코드 분석

04 | 고찰 및 결론



시작하기 앞서 본 팀의 발표 주제는
객관적인 시선으로
문제를 해결하기 위한 평범한 주제일 뿐,
팀원들의 정치적 성향과는 무관함을 밝힙니다.

01

주제 선정 동기

- POINT 01 여소야대란?
- POINT 02 여소야대의 문제점
- POINT 03 여소야대의 발생 원인

01

주제 선정 동기_ 여소야대란?

“

”

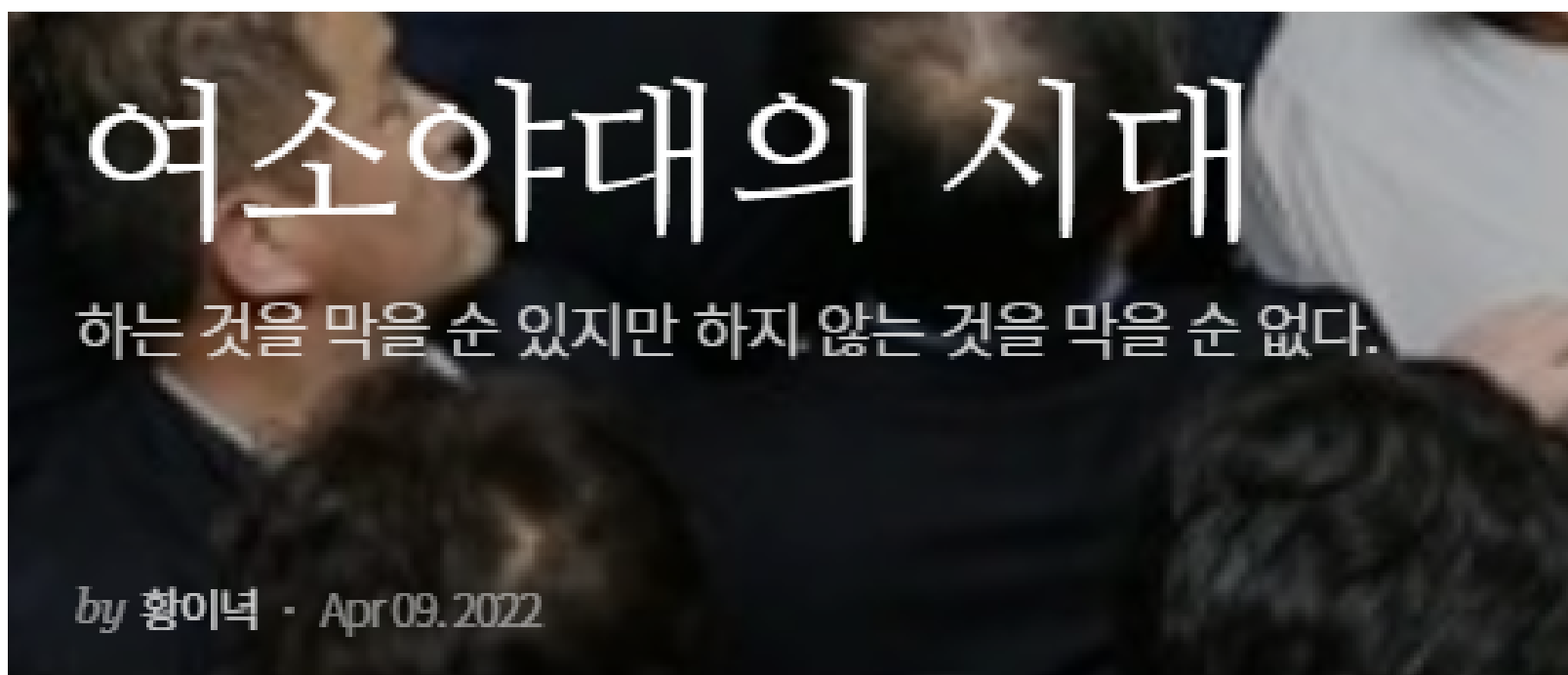
여소야대란 ?



'여'가 작고, '야'가 크다는 것으로, 정권을 창출한 여당의
의원 수가 적고, 야당의 의원 수가 많은 정치 형국을 가리키는 용어.

01

주제 선정 동기_기사 스크랩



여소야대의 가장 큰 문제점은 무언가 하는 것을 막을 순 있지만 ‘하지 않는 것’을 막을 순 없다는 것이다. 후쿠시마산 농수산물을 수입하는 걸 막을 순 있어도 WTO 소송에 적극적으로 임하지 않는 것을 막을 순 없다는 것이고, 여성가족부를 폐지하는 걸 막을 순 있어도 위기 청소년 지원 정책을 하지 않는 것을 막을 순 없다는 것이다. 보통 사람들은 여소야대의 상황에선 행정부와 입법부의 상호 견제가 원활히 이루어질 것이라 기대하지만, 오히려 정당의 이념에 사로잡힌 묻지마 견제의 부작용으로 정국이 불안해지는 것을 넘어 멈추어질 우려가 크다.

출처 : "여소야대의 시대", brunchstory, 황이녀

美전문가 “尹, ‘여소야대’ 야당과 외교정책 깊이 논의해야”

중앙일보 | 입력 2024.04.12 01:59 업데이트 2024.04.12 10:53

정시내 기자 [구독](#)



출처 : 美전문가 “尹, ‘여소야대’ 야당과 외교정책 깊이 논의해야”, 중앙일보, 정시내 기자

01

주제 선정 동기_ 여소야대의 문제점

01 | 정책 실행의 어려움

여당이 추진하는 법안이나 정책이 야당의 반대에 부딪혀 통과되기 어려움.
정부의 정책 집행이 지연되거나 무산될 수 있음

02 | 정책의 일관성 부족

야당이 다수인 경우, 여당이 추진하는 정책이 중단되거나 변경될 가능성이 큼.
정책의 일관성을 저해하고 장기적인 국가 계획 수립에 부정적인 영향을 미침.

03 | 경제적 불안정성

정치적 불안정성 → 경제적 불안정성
투자자들은 정치적 리스크를 고려하여 투자 결정을 내림. 따라서 경제 성장이 저해될 가능성이 있음.



어떻게 하면 해결할 수 있을까 ?

01

주제 선정 동기_발생 원인

발생 원인 ?

1. 정당 또는 대통령의 국정 운영이 미흡
2. 이전 총선, 대선에서 승리했던 곳에서 패배 ✓
3. 치열한 경합인 곳에서 패배 ✓

서울특별시로 한정하여 위 2,3번의 교집합을 '승부처'로 분류하였고,
승부처를 알아냄으로써 여소야대를 예방하고자 한다.

02

데이터 수집

- POINT 01 사용 데이터
- POINT 02 18대 대선, 19대 총선 데이터

02

데이터 수집_사용 데이터

출처 :

중앙선거관리위원회 : 선거인 수 집계, 투표 수 집계, 여당, 야당 투표율

- 18대, 19대, 20대 대선
- 19대, 20대, 21대, 22대 총선, 비례대표

 Google Docs 18대 대선.csv

 Google Docs 19대 대선.csv

 Google Docs 19대 총선_비례.csv


 Google Docs 19대 총선.csv

 Google Docs 20대 대선.csv


 Google Docs 20대 총선_비례.csv

 Google Docs 20대 총선.csv

 Google Docs 21대 총선_비례.csv

 Google Docs 21대 총선.csv

 Google Docs 22대 총선_비례.csv

 Google Docs 22대 총선.csv

02

데이터 수집_사용 데이터

ex) 19대 총선 데이터

선거구명	선거인수합계	투표수합계	투표율	여득표율 - 야득표율
강남구갑	253,525	127,181	50.16507	32.49
강남구을	204,733	124,113	60.62188	20.21
강동구갑	218,027	122,935	56.38522	3.76
강동구을	180,815	95,834	53.00113	-8.33
강북구갑	140,378	72,107	51.36631	-7.35
강북구을	146,336	75,876	51.85054	-11.63
강서구갑	245,015	132,414	54.04322	-6.23
강서구을	216,397	124,236	57.41115	0.71
관악구갑	235,207	128,829	54.7726	63.61
관악구을	211,859	113,913	53.76831	37.55
광진구갑	150,634	85,519	56.77271	-7.56

ex) 18대 대선 데이터

선거구명	선거인수합계	투표수합계	투표율	여득표율 - 야득표율
강남구	461,592	127,181	27.55269	20.68
강동구	397,018	300,362	75.6545	1.22
강북구	286,118	207,927	72.67176	-8.22
강서구	463,571	350,023	75.5058	-6.5
관악구	446,910	333,640	74.65485	-18.81
광진구	307,590	231,171	75.15556	-6.9
구로구	350,373	269,094	76.80215	-7.69
금천구	202,808	147,632	72.79397	-9.28
노원구	473,741	368,003	77.68021	-6.67
도봉구	296,521	225,230	75.95752	-4.16
동대문구	305,303	227,177	74.41034	-3.96

선거인 수와 각 당의 투표수를 이용하여 여당 득표율 - 야당 득표율을 계산

03

데이터 시각화 및 코드 분석

- **POINT 01** 대선 - 선그래프, box plot, heatmap
- **POINT 02** 총선, 비례대표 - 선그래프, box plot, heatmap
- **POINT 03** 승부처 교집합

03

코드 분석_선 그래프

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.font_manager as fm
```

```
!apt-get update -qq #한글 폰트 설치
!apt-get install fonts-nanum* -qq
```

```
font_path = '/usr/share/fonts/truetype/nanum/NanumGothic.ttf'
fontprop = fm.FontProperties(fname=font_path)
plt.rc('font', family='NanumGothic') #설치된 폰트 설정
```

```
from google.colab import drive #Google Drive 마운트
drive.mount('/content/drive')
```

```
file_path_18 = '/content/drive/My Drive/data/18대 대선.csv'
file_path_19 = '/content/drive/My Drive/data/19대 대선.csv'
file_path_20 = '/content/drive/My Drive/data/20대 대선.csv'
```

파일 경로 설정

try: # csv 파일 로드 및 선 그래프 생성

```
df_18 = pd.read_csv(file_path_18, encoding='utf-8')
print("18대 대선 CSV 데이터의 첫 몇 행:")
print(df_18.head()) # 대선 데이터 불러오기
```

```
df_19 = pd.read_csv(file_path_19, encoding='utf-8')
print("19대 대선 CSV 데이터의 첫 몇 행:")
print(df_19.head())
```

```
df_20 = pd.read_csv(file_path_20, encoding='utf-8')
print("20대 대선 CSV 데이터의 첫 몇 행:")
print(df_20.head())
```

```
required_columns = ['선거구명', '여득표율 - 야득표율']
if all(column in df_18.columns for column in required_columns) and \
    all(column in df_19.columns for column in required_columns) and \
    all(column in df_20.columns for column in required_columns):
```

필요한 열이 존재하는지 확인

```
df_18['선거구명'] = df_18['선거구명'].astype(str)
df_19['선거구명'] = df_19['선거구명'].astype(str)
df_20['선거구명'] = df_20['선거구명'].astype(str)
```

선거구명을 문자열로 변환

```
df_18_sorted = df_18.sort_values('선거구명')
x_18 = df_18_sorted['선거구명']
y_18 = df_18_sorted['여득표율 - 야득표율']
```

```
df_19_sorted = df_19.sort_values('선거구명')
x_19 = df_19_sorted['선거구명']
y_19 = df_19_sorted['여득표율 - 야득표율']
```

```
df_20_sorted = df_20.sort_values('선거구명')
x_20 = df_20_sorted['선거구명']
y_20 = df_20_sorted['여득표율 - 야득표율']
```

대선 데이터 정렬하기

선그래프 생성

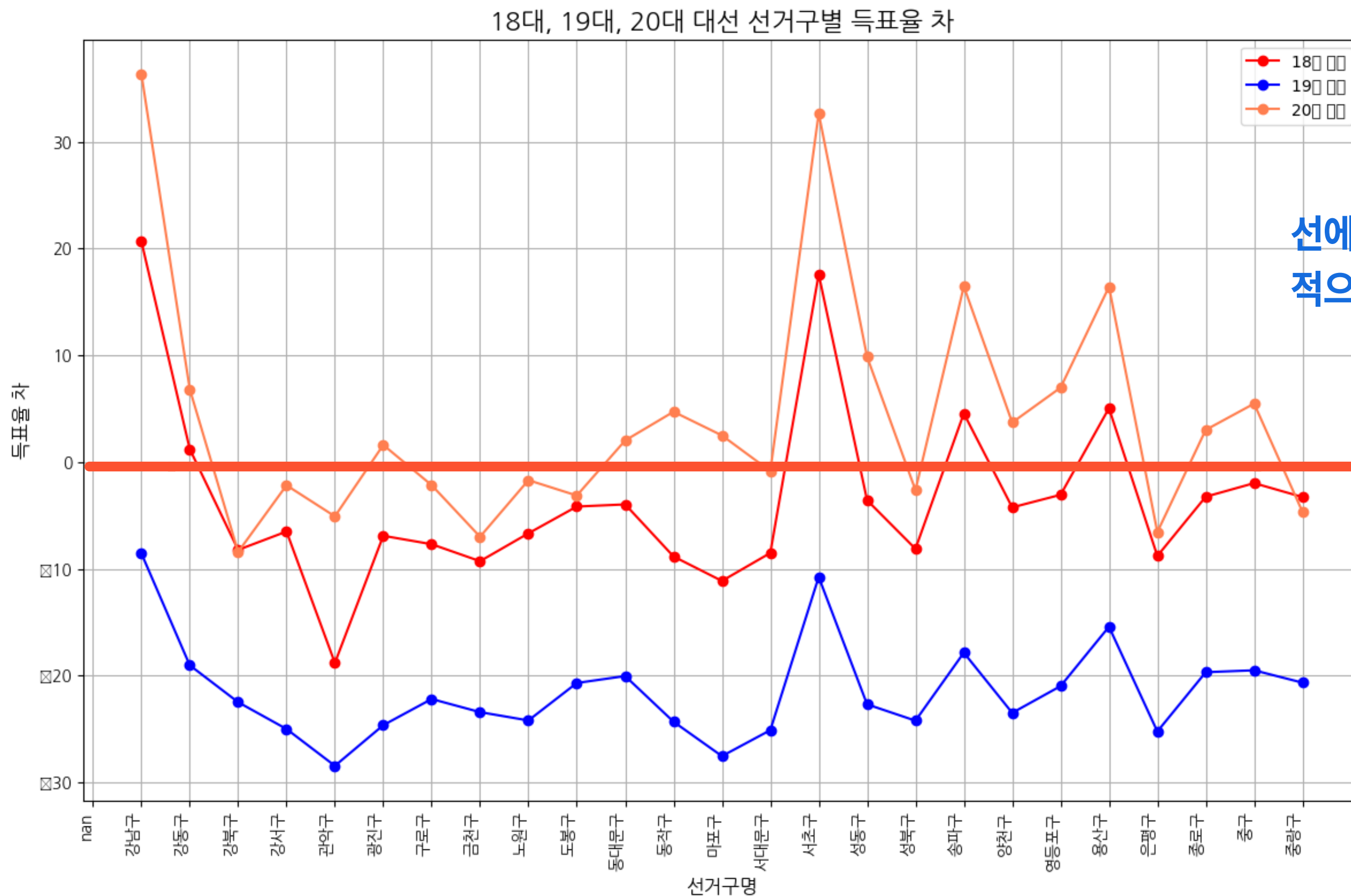
```
plt.figure(figsize=(12, 8))
plt.plot(x_18, y_18, marker='o', linestyle='-', color='red', label='18대 대선')
plt.plot(x_19, y_19, marker='o', linestyle='-', color='blue', label='19대 대선')
plt.plot(x_20, y_20, marker='o', linestyle='-', color='coral', label='20대 대선')

plt.title('18대, 19대, 20대 대선 선거구별 득표율 차', fontproperties=fontprop, fontsize=16)
plt.xlabel('선거구명', fontproperties=fontprop, fontsize=12)
plt.ylabel('득표율 차', fontproperties=fontprop, fontsize=12)
plt.xticks(rotation=90, ha='right', fontproperties=fontprop, fontsize=10)
plt.yticks(fontproperties=fontprop, fontsize=10)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

03

데이터 시각화_선 그래프(대선)

*결과



선에 가까울수록 득표율 차가 적으므로 승부처로 파악됨.

03

코드 분석_box plot

*이전은 선그래프 코드와 같음

```
df_all = pd.concat([df_18, df_19, df_20], ignore_index=True)
#새 데이터프레임을 하나로 합치기

plt.figure(figsize=(12, 8))
sns.boxplot(x='선거구명', y='여득표율 - 야득표율', data=df_all)
#box plot 생성

plt.axhline(y=0, color='red', linestyle='--', linewidth=2)
# y축 0에 해당하는 가로선을 빨간색으로 그리기

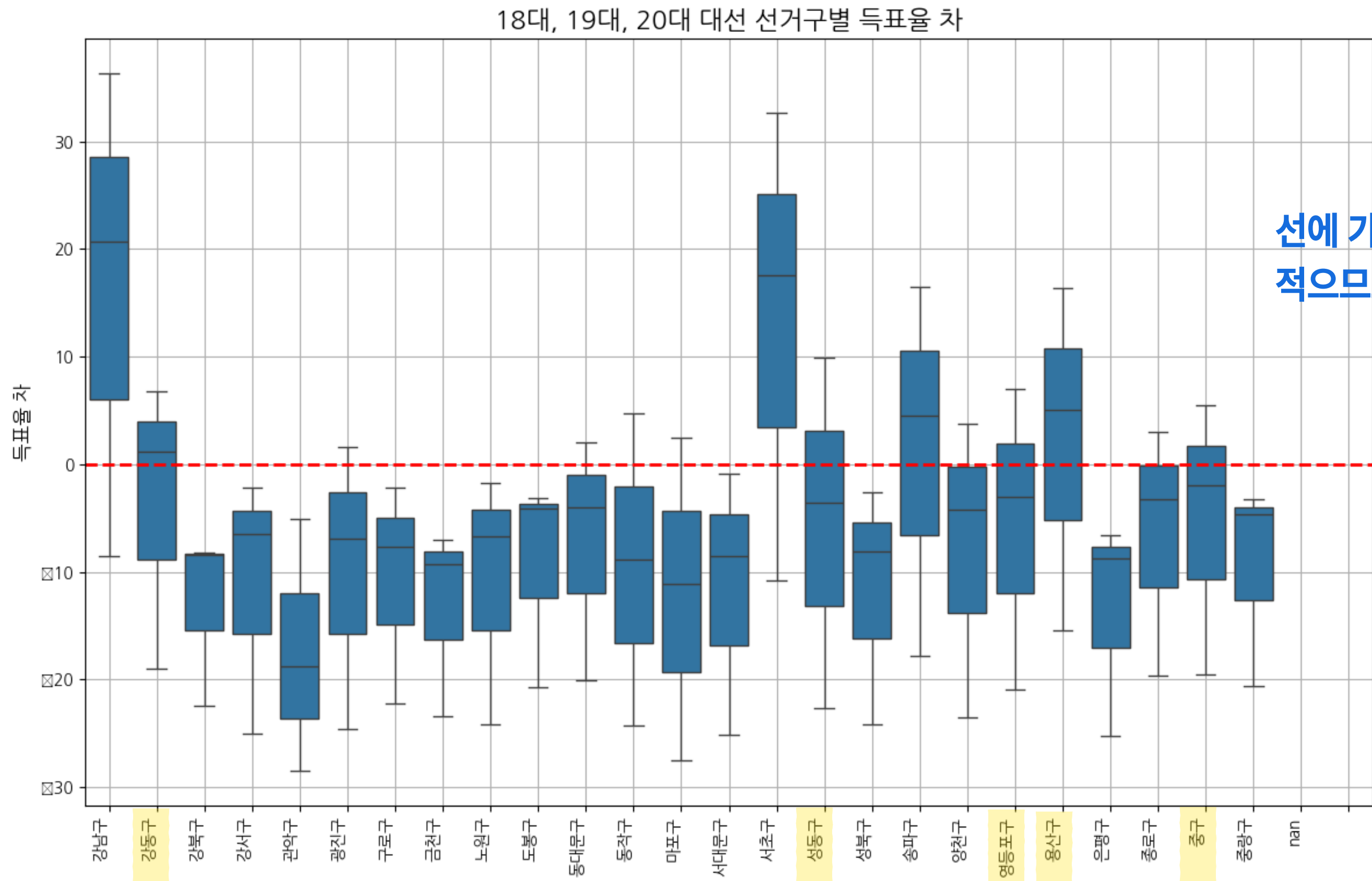
plt.title('18대, 19대, 20대 대선 선거구별 득표율 차', fontsize=16)
plt.xlabel('선거구명', fontsize=12)
plt.ylabel('득표율 차', fontsize=12)
plt.xticks(rotation=90, ha='right', fontsize=10)
plt.yticks(fontsize=10)
plt.grid(True)
plt.tight_layout()
plt.show()

else:
    print("필요한 열이 CSV 파일에 존재하지 않습니다.")
except FileNotFoundError as e:
    print(f"파일을 찾을 수 없습니다: {e.filename}")
except pd.errors.ParserError as e:
    print(f"파일 파싱 중 오류가 발생했습니다: {e}")
except Exception as e:
    print(f"다음과 같은 오류가 발생했습니다: {e}")
```


03

데이터 시각화_box plot(대선)

*결과



03

코드 분석_Heatmap

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.font_manager as fm
```

```
!apt-get update -qq
!apt-get install fonts-nanum* -qq
```

#한글 폰트 설치

```
font_path = '/usr/share/fonts/truetype/nanum/NanumGothic.ttf'
fontprop = fm.FontProperties(fname=font_path)
plt.rc('font', family='NanumGothic')
```

#설치된 폰트 설정

```
from google.colab import drive
drive.mount('/content/drive')
```

#Google Drive 마운트

```
file_path = '/content/drive/My Drive/data/18대 대선.csv'
```

파일 경로 설정

```
try:
    df = pd.read_csv(file_path, encoding='utf-8')
    print("CSV 데이터의 첫 몇 행:")
    print(df.head())
```

csv 파일 로드

```
required_columns = ['선거구명', '여득표율 - 야득표율']
if all(column in df.columns for column in required_columns):
```

필요한 열이 존재하는지 확인

히트맵 생성

```
plt.figure(figsize=(12, 8))
heatmap_data = df.pivot_table(index='선거구명', values='여득표율 - 야득표율')
```

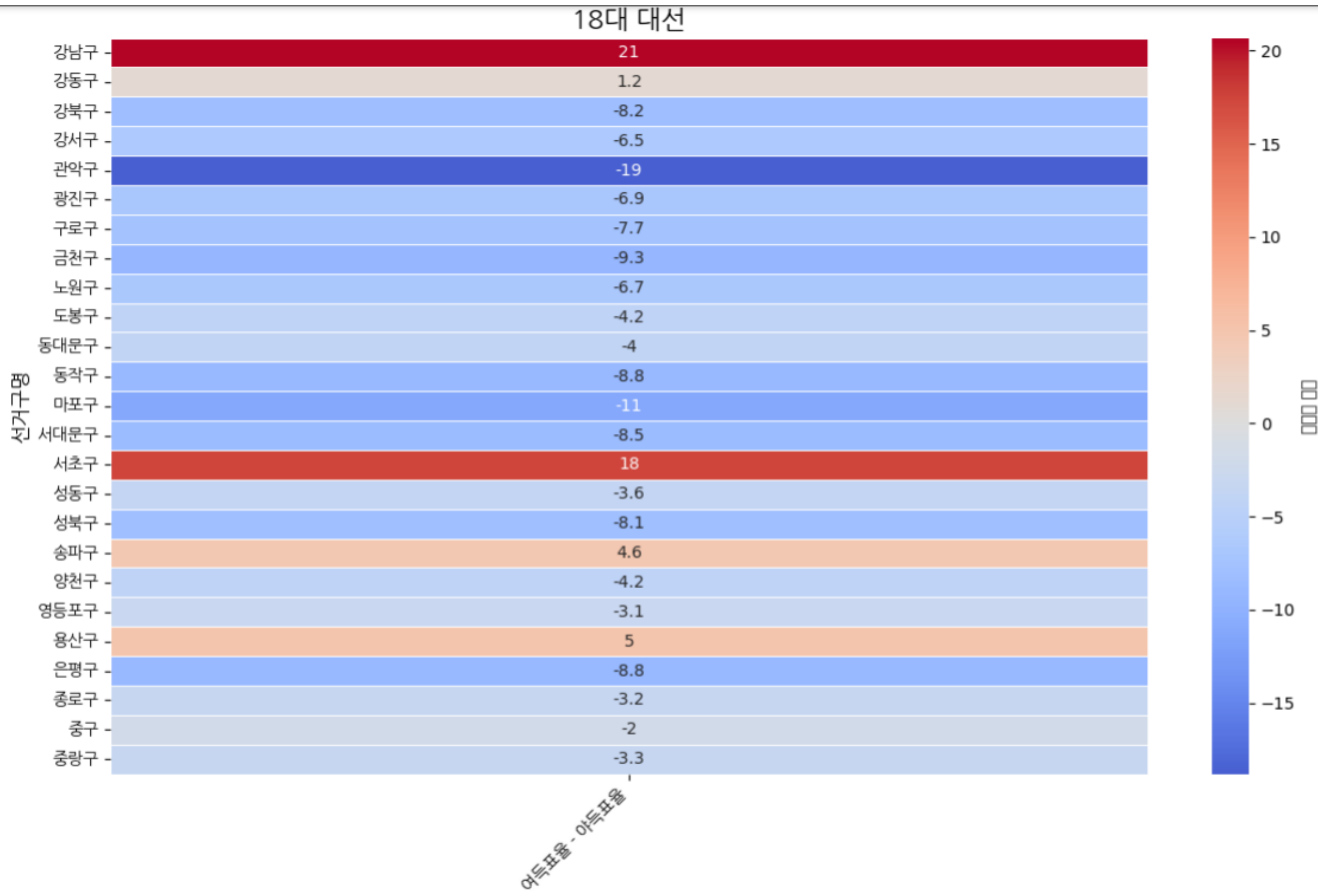
```
ax = sns.heatmap(heatmap_data, annot=True, cmap='coolwarm', center=0, linewidths=.5, cbar_kws={'label': '득표율 차이'})
ax.set_title('18대 대선', fontproperties=fontprop, fontsize=16)
ax.set_xlabel('색상강도', fontproperties=fontprop, fontsize=12)
ax.set_ylabel('선거구명', fontproperties=fontprop, fontsize=12)
```

```
plt.xticks(rotation=45, ha='right', fontproperties=fontprop, fontsize=10)
plt.yticks(rotation=0, fontproperties=fontprop, fontsize=10)
plt.tight_layout()
plt.show()
```

```
else:
    print("필요한 열이 CSV 파일에 존재하지 않습니다.")
except FileNotFoundError:
    print(f"파일을 찾을 수 없습니다: {file_path}")
except pd.errors.ParserError as e:
    print(f"파일 파싱 중 오류가 발생했습니다: {file_path}#{e}")
except Exception as e:
    print(f"다음과 같은 오류가 발생했습니다: {e}")
```

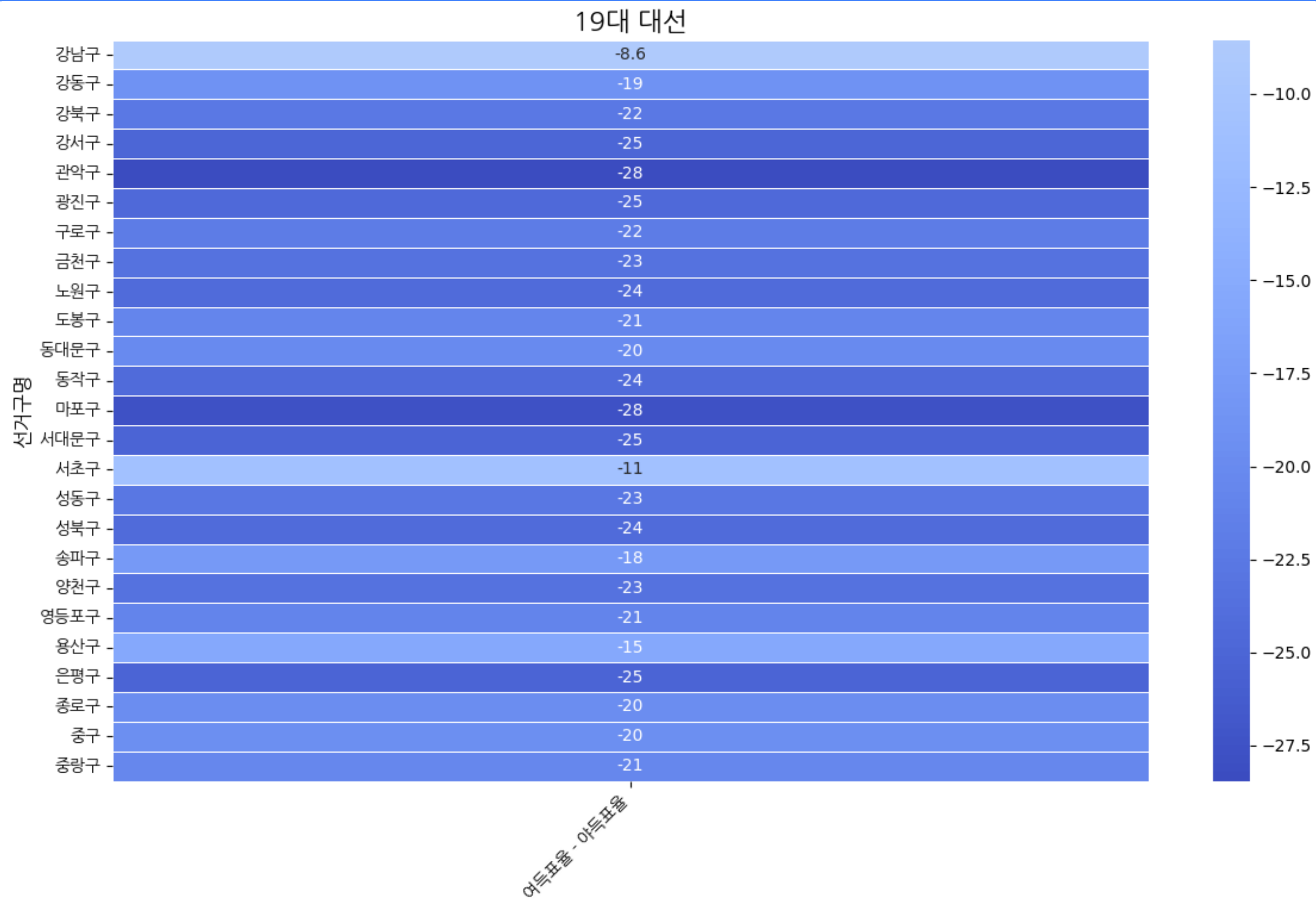
03

데이터 시각화_Heatmap(대선)



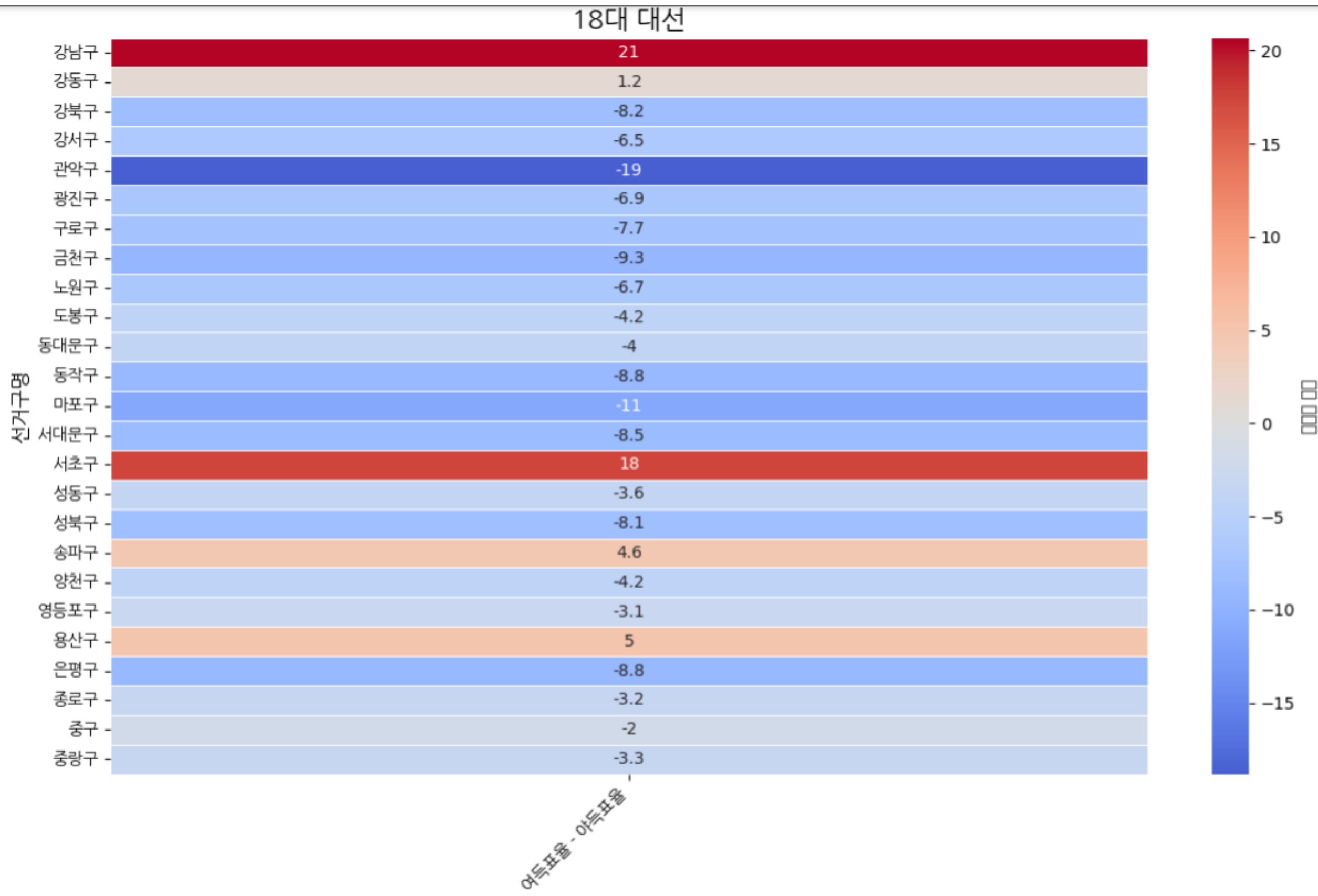
03

데이터 시각화_Heatmap(대선)



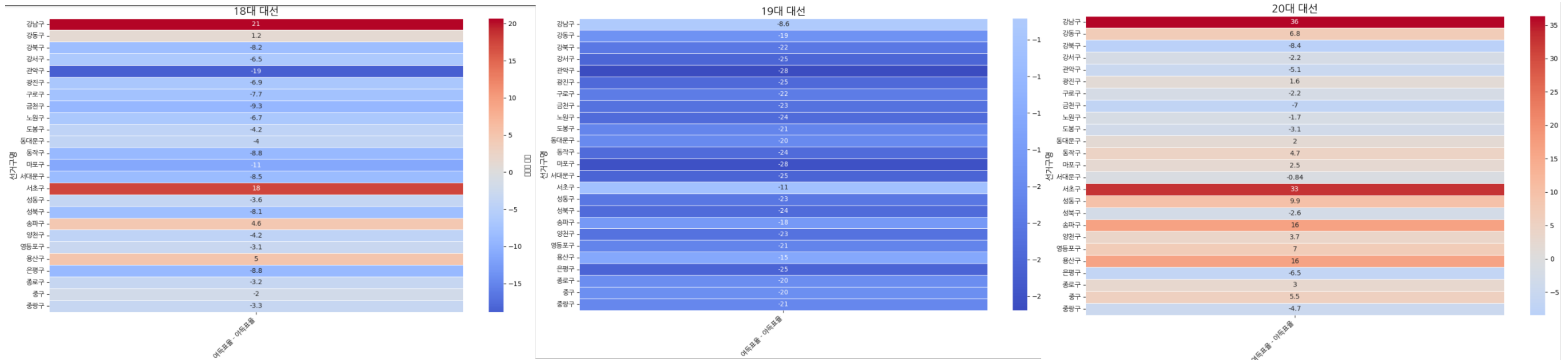
03

데이터 시각화_Heatmap(대선)



03

데이터 시각화_Heatmap(대선)



승부처로 파악되는 곳

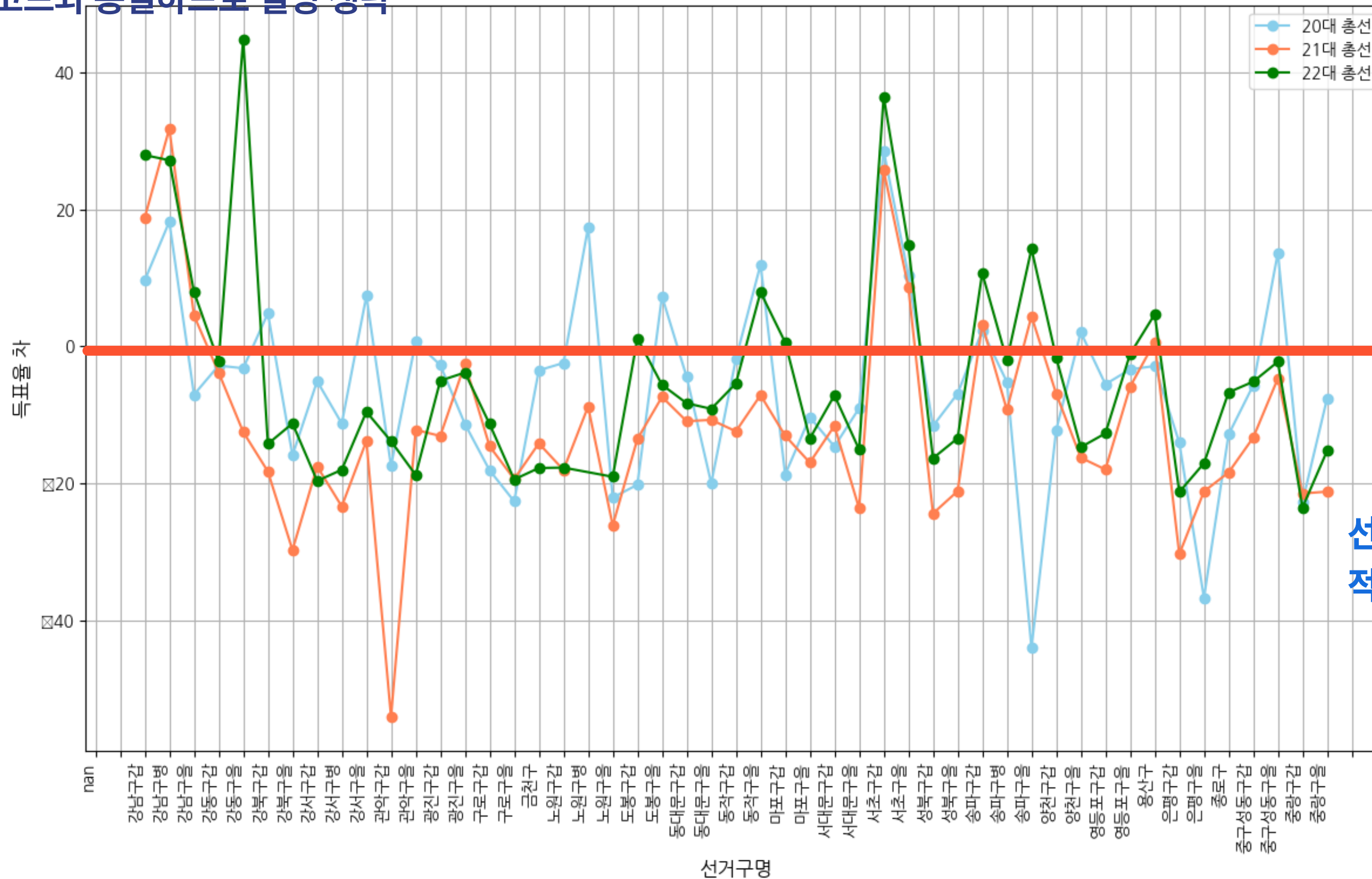
→ 강동구, 성동구, 영등포구, 용산구, 종로구, 중구

03

데이터 시각화_선 그래프(총선)

* 총선 역시 앞 코드와 동일하므로 설명 생략

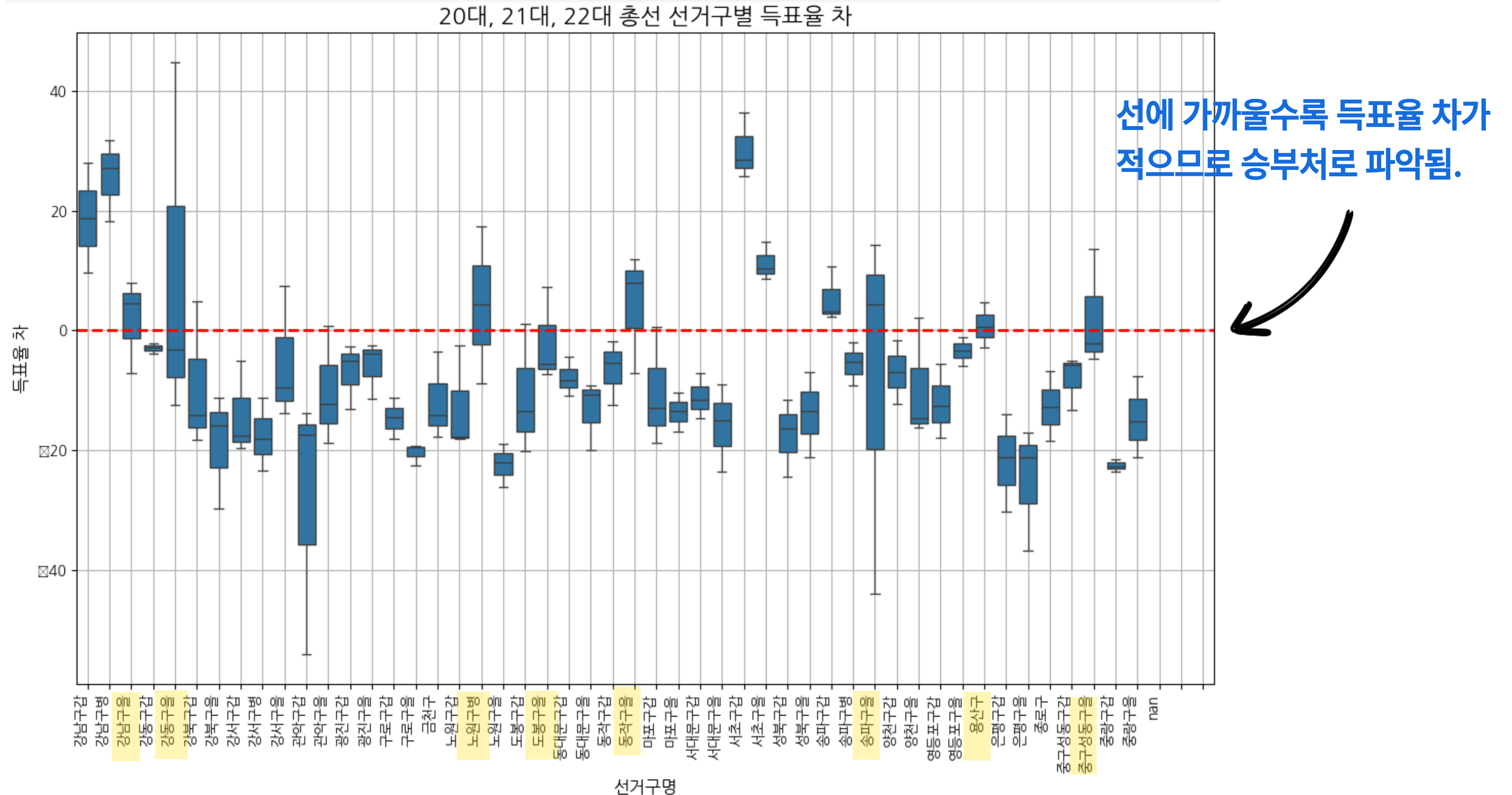
20대, 21대, 22대 총선 선거구별 득표율 차



선에 가까울수록 득표율 차가 적으므로 승부처로 파악됨.

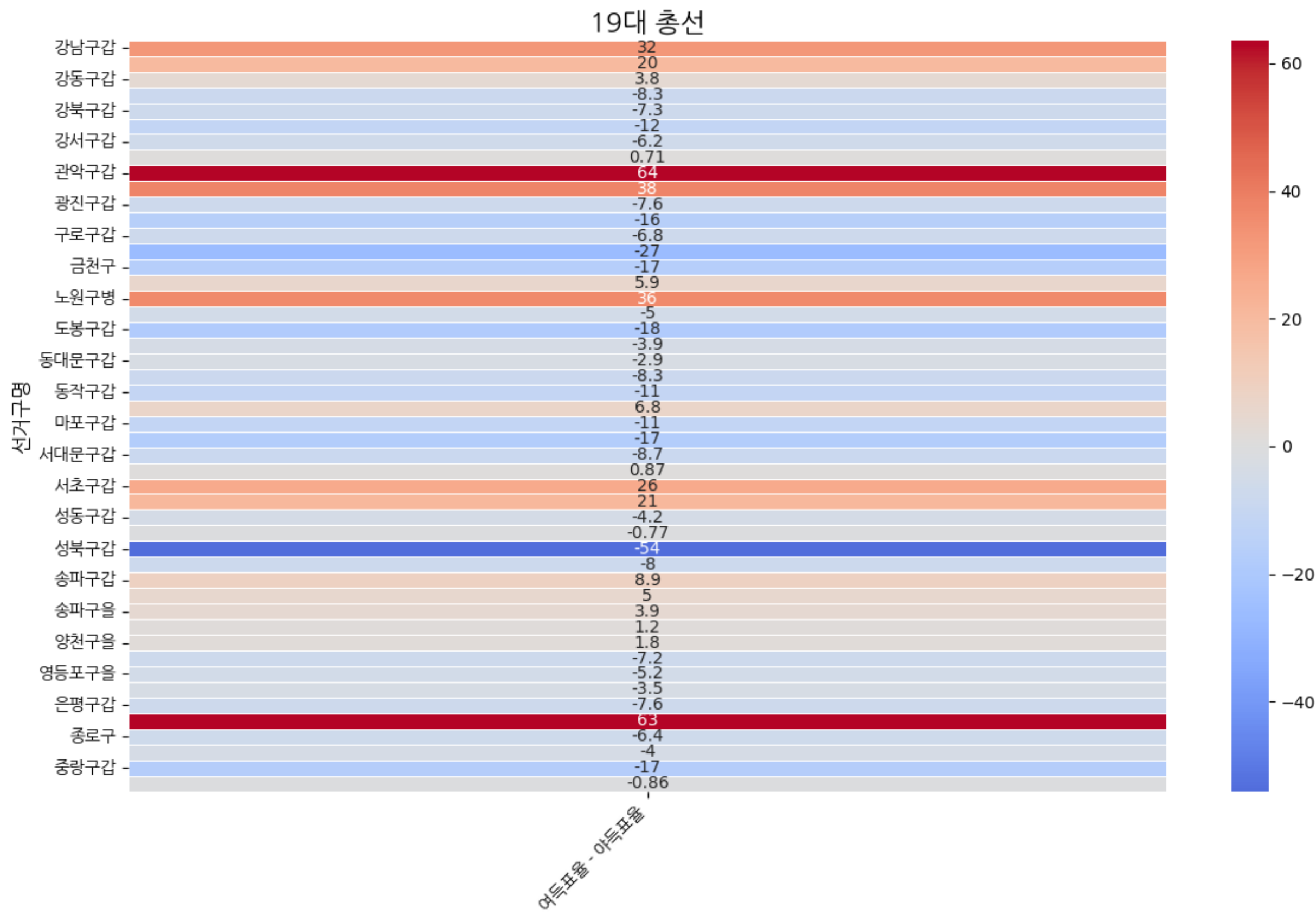
03

데이터 시각화_box plot(총선)



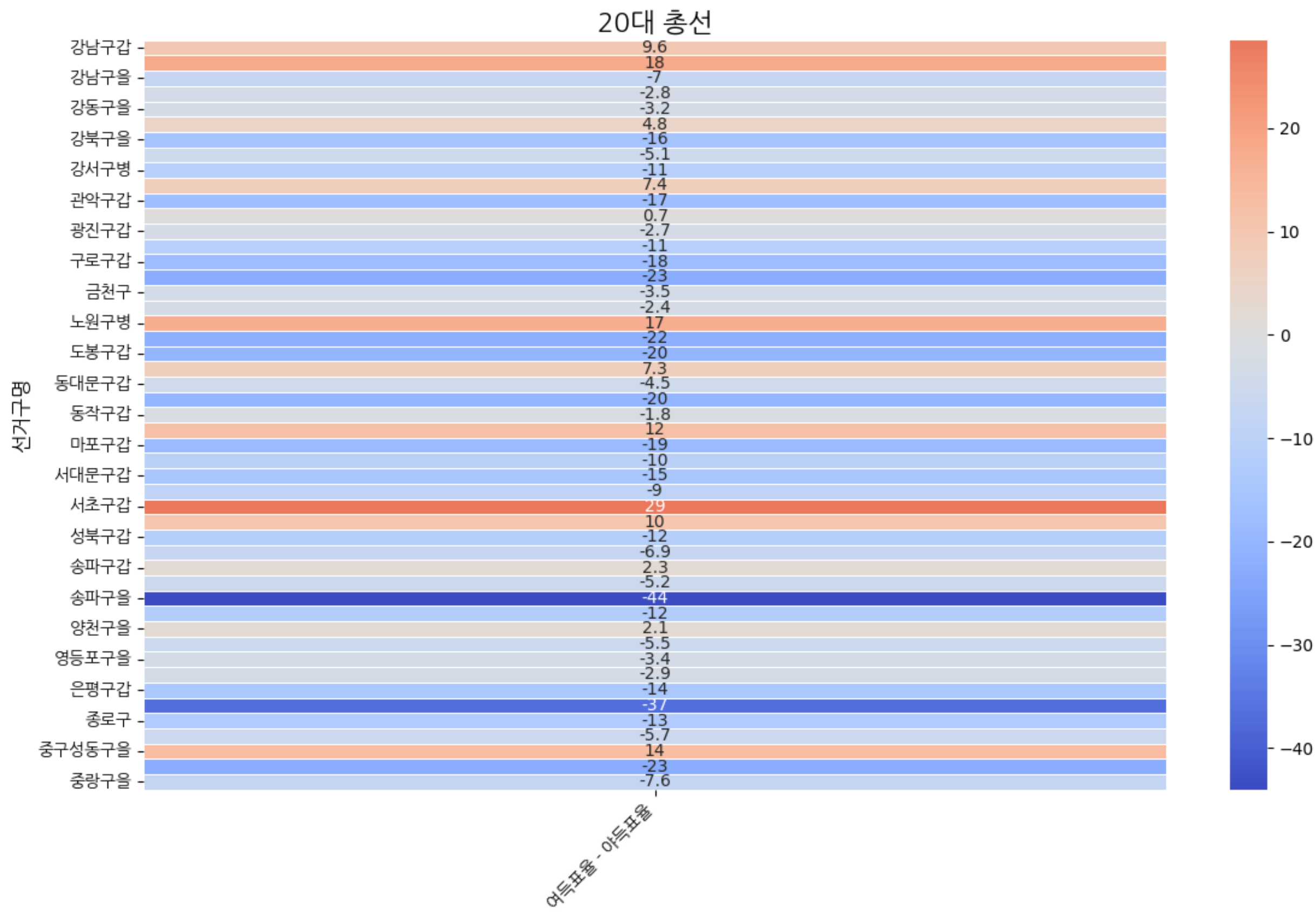
03

데이터 시각화_Heatmap(총선)



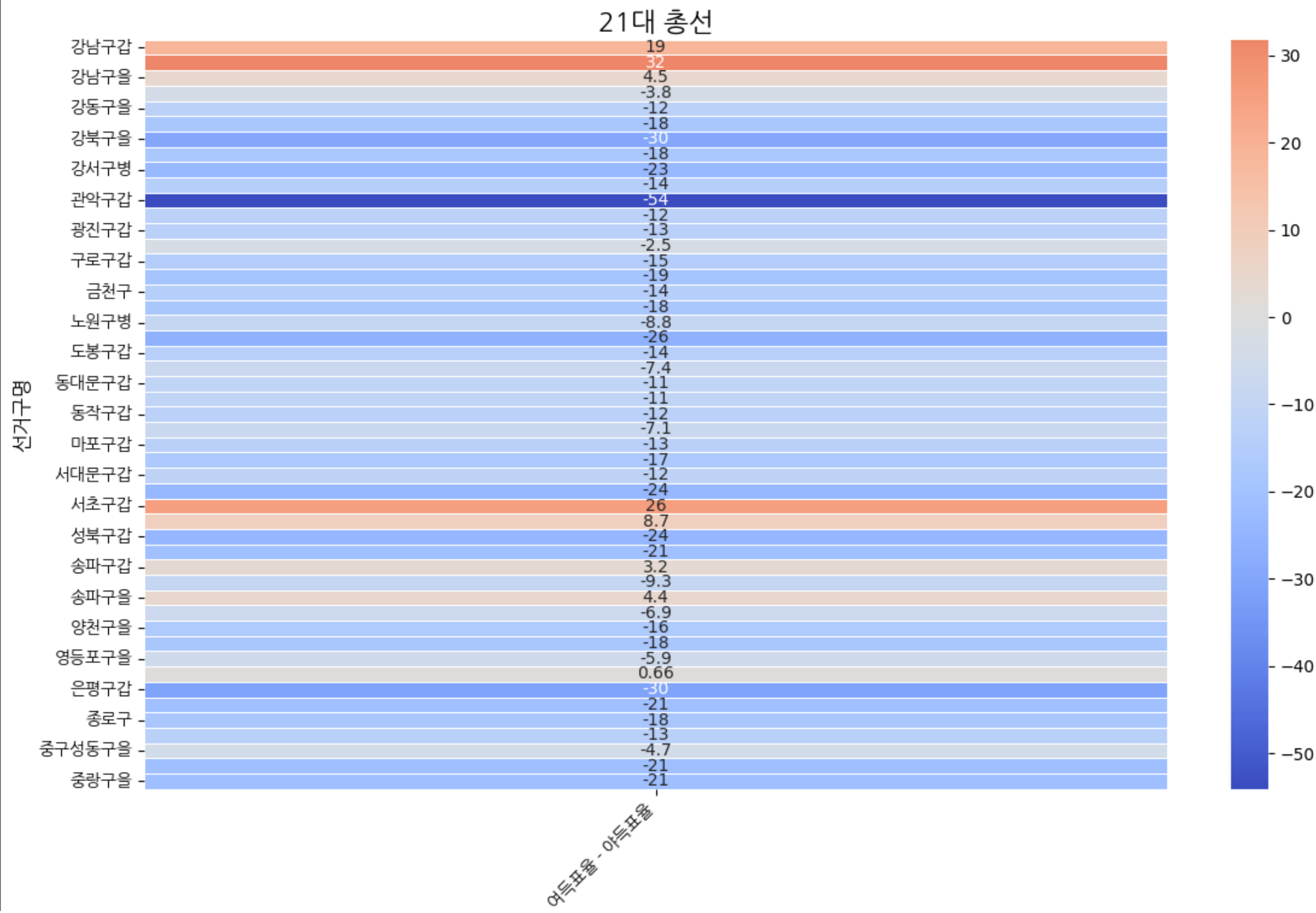
03

데이터 시각화_Heatmap(총선)



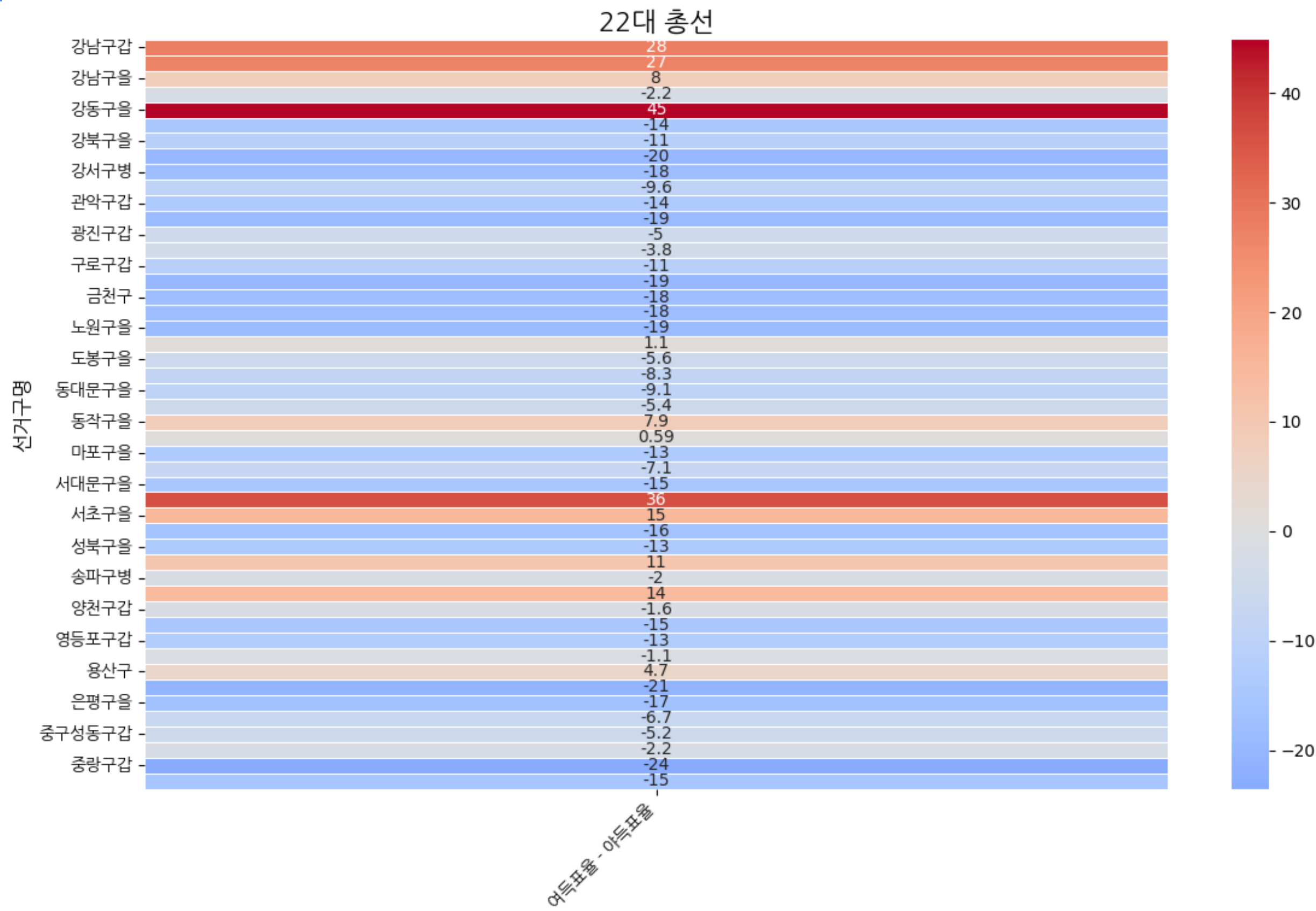
03

데이터 시각화_Heatmap(총선)



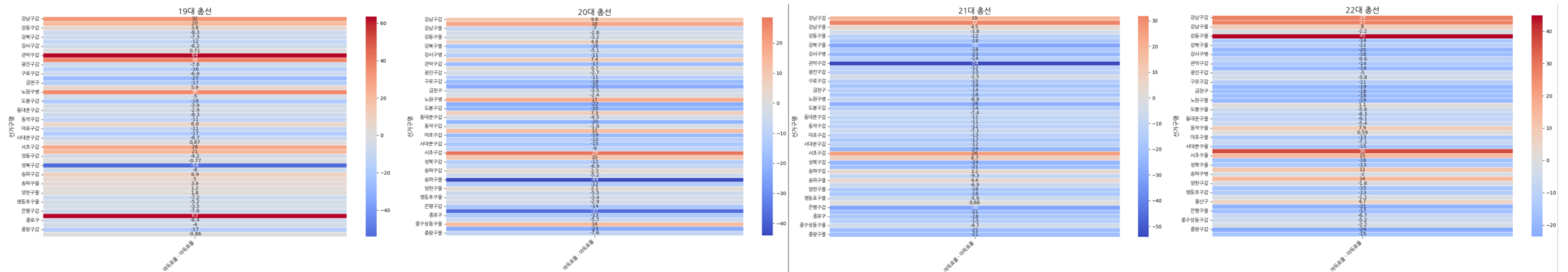
03

데이터 시각화_Heatmap(총선)



03

데이터 시각화_Heatmap



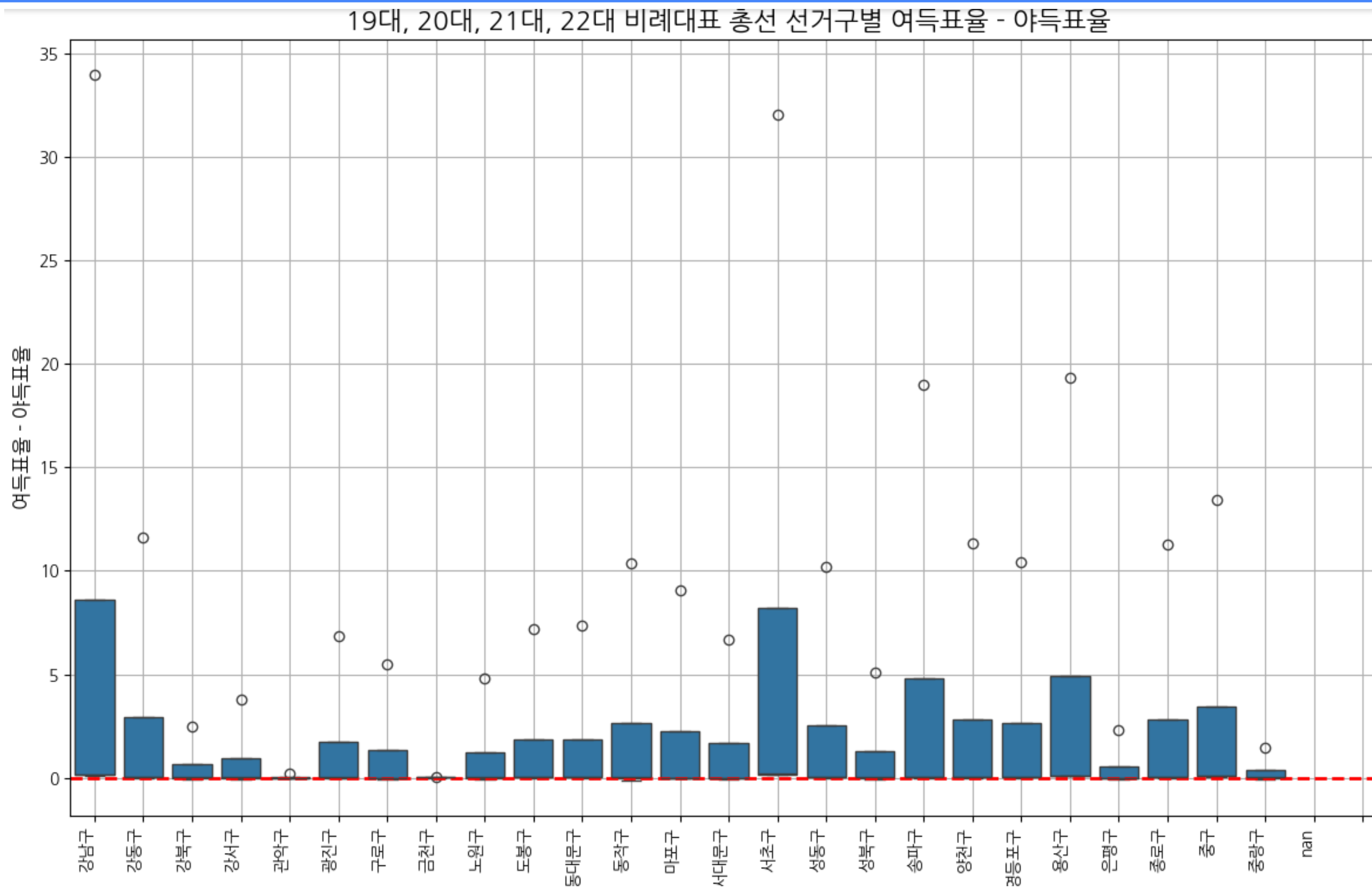
승부처로 파악되는 곳

→ 강남구을, 강동구을, 노원구병, 도봉구을, 동작구을, 송파구을, 용산구, 중구성동구을

03

데이터 시각화_box plot(비례대표)

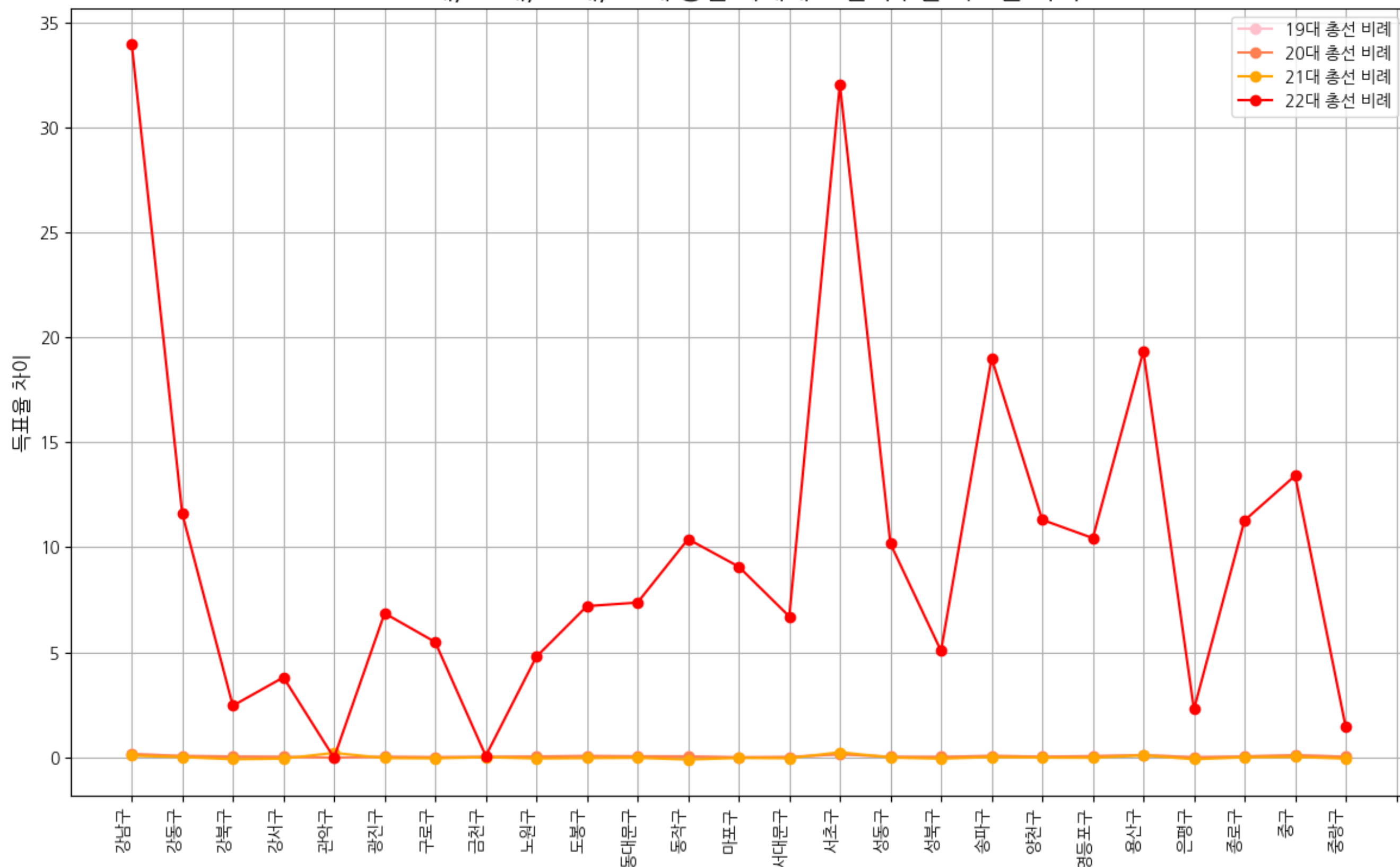
+ 추가



03

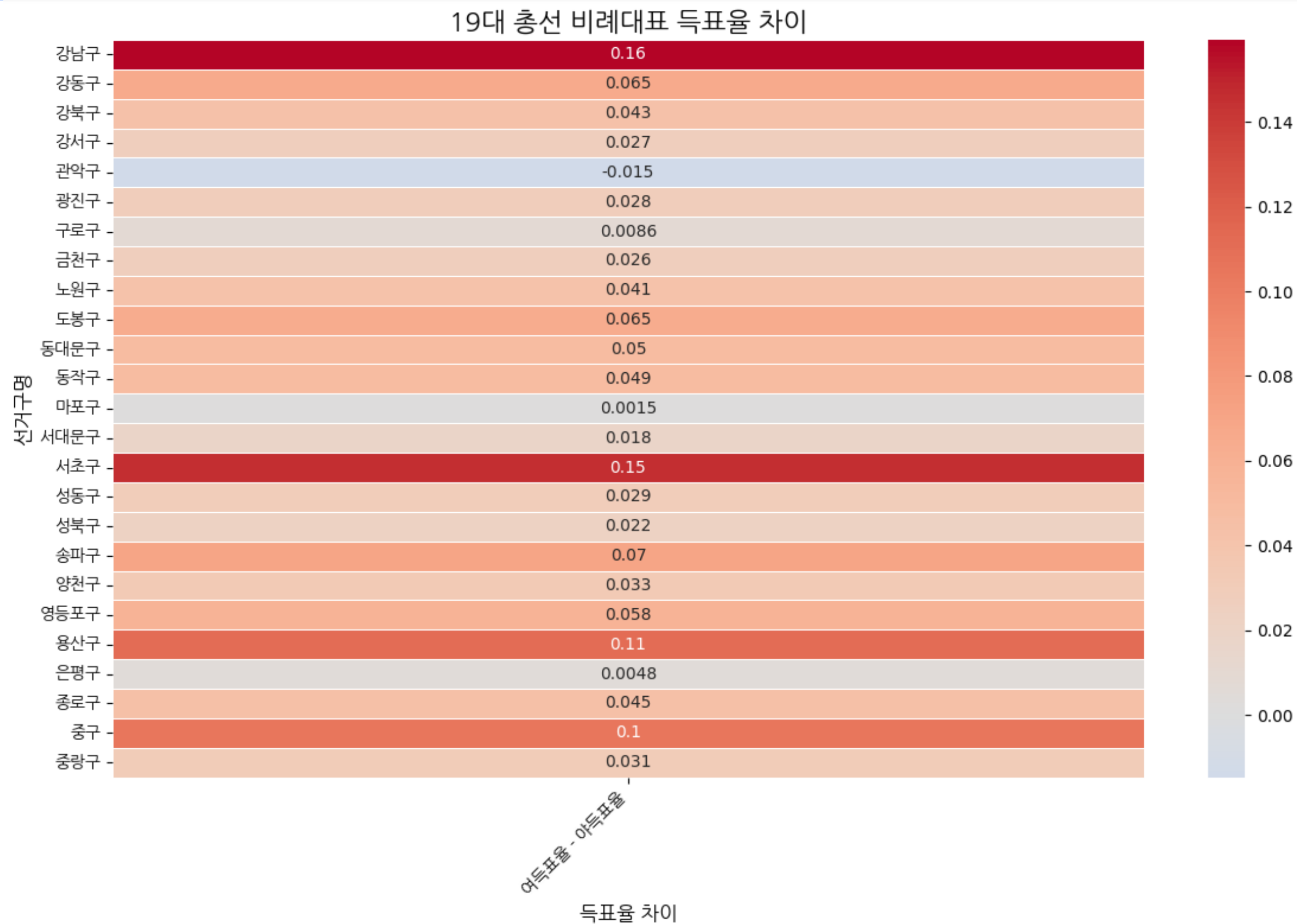
데이터 시각화_선 그래프(비례대표)

19대, 20대, 21대, 22대 총선 비례대표 선거구별 득표율 차이



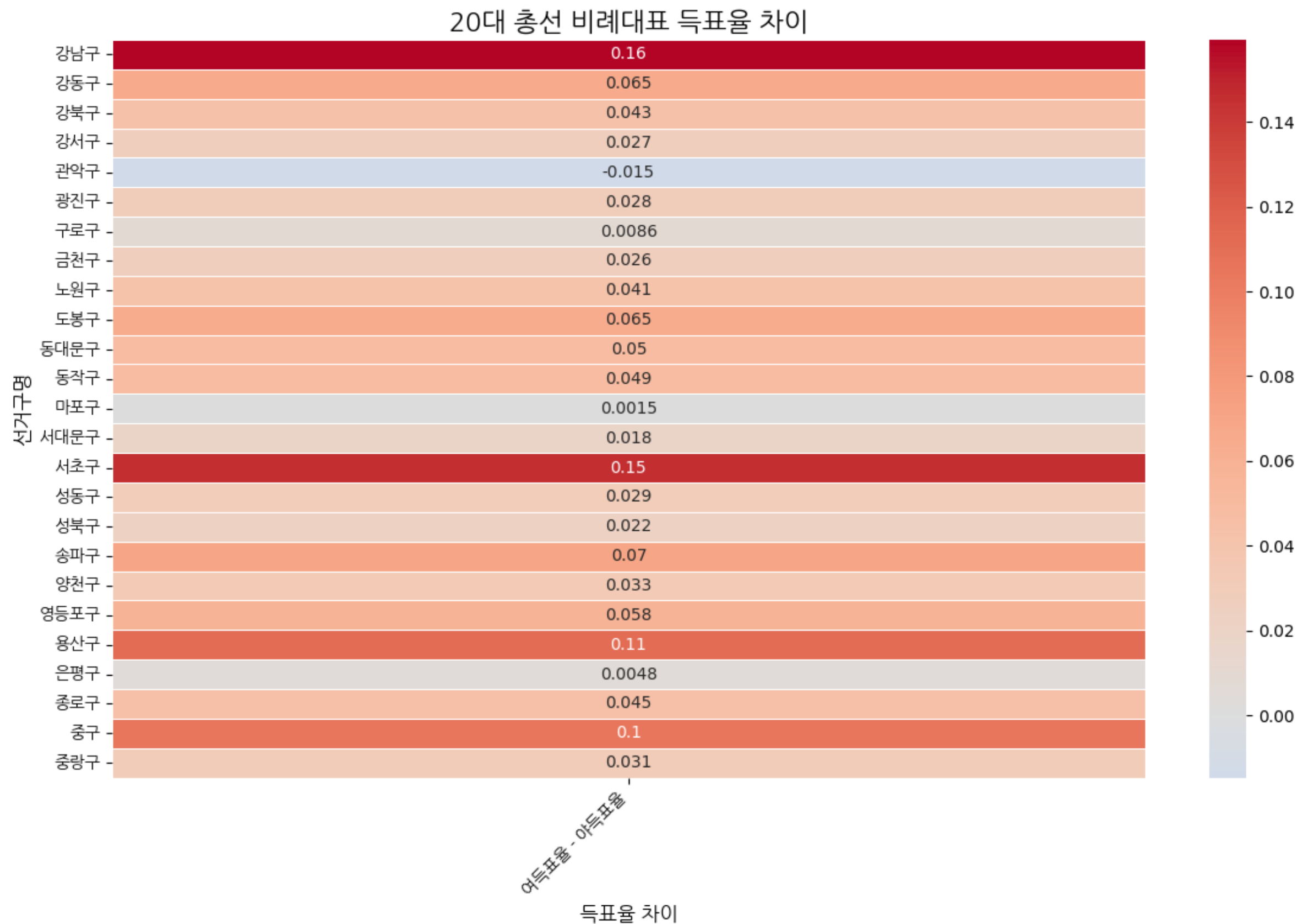
03

데이터 시각화_Heatmap(비례대표)



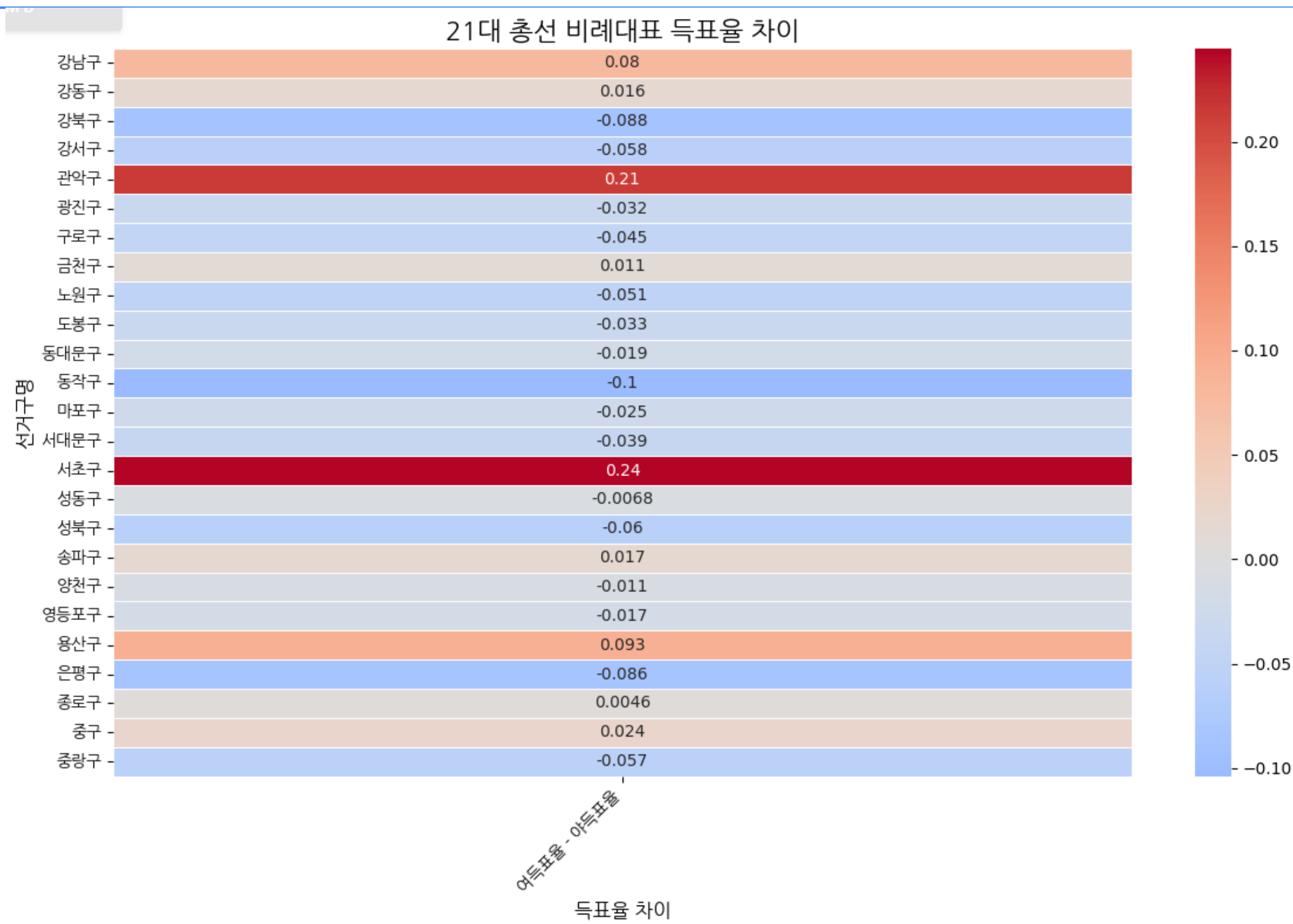
03

데이터 시각화_Heatmap(비례대표)



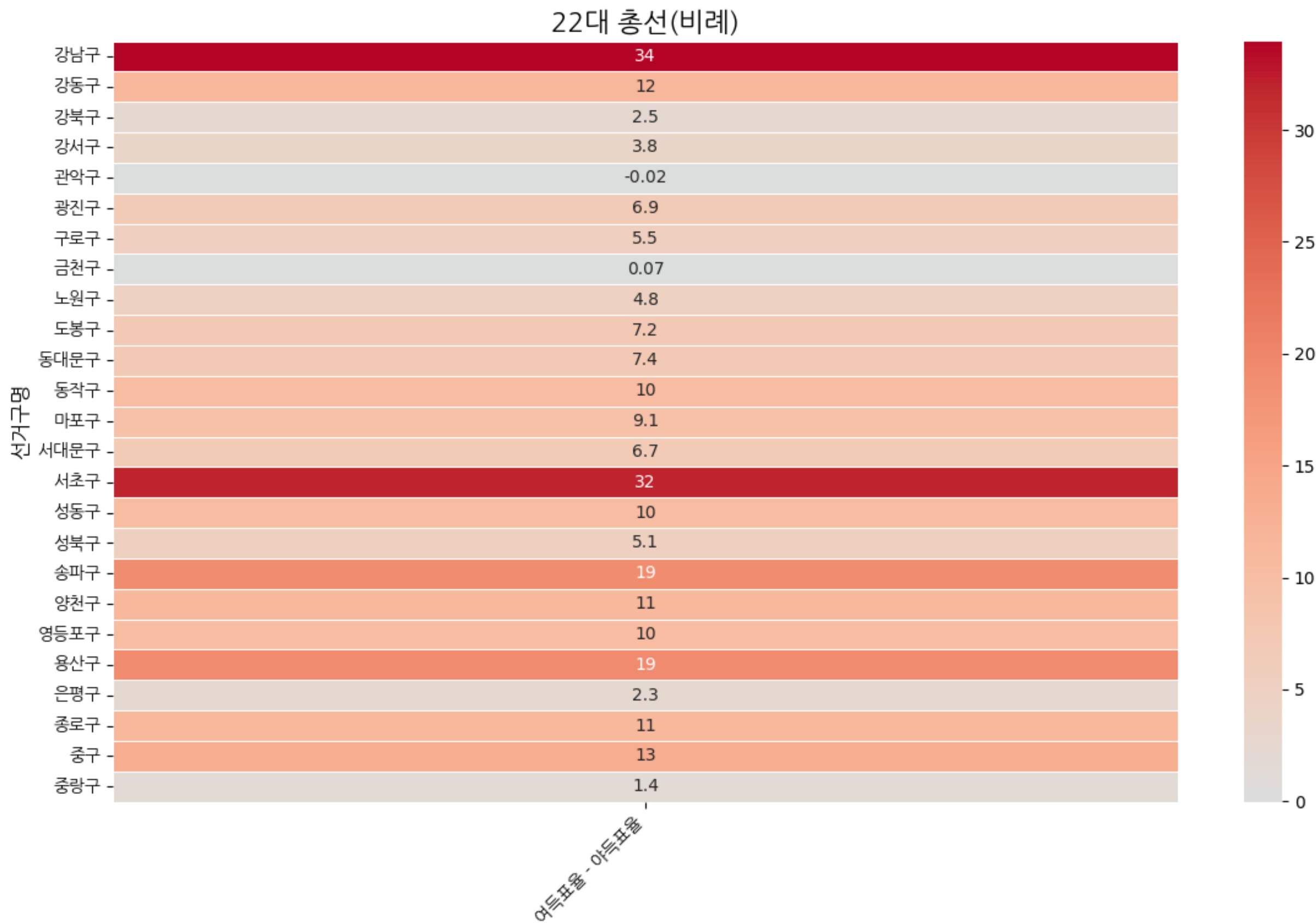
03

데이터 시각화_Heatmap(비례대표)



03

데이터 시각화_Heatmap(비례대표)



03

데이터 시각화_Heatmap

승부처로 파악되는 곳(대선)

→ 강동구, 성동구, 영등포구, 용산구, 종로구, 중구

+

승부처로 파악되는 곳(총선)

→ 강남구을, 강동구을, 노원구병, 도봉구을, 동작구을, 송파구을, 용산구, 중구성동구을

∨

승부처의 교집합

→ 강동구, 성동구, 용산구, 중구

04

고찰 및 결론

Result

결과

승부처의 교집합 → 강동구, 성동구, 용산구, 중구



Solution

해결방안

- 격전지역 맞춤형 공약 개발
- 주민과의 소통 강화
- 지속적인 지지 기반 강화
- 청년층, 노년층, 여성 등 다양한 계층의 목소리를 반영하는 정책을 제안

이러한 전략들을 통해 여당은 승부처에서의 득표율을 높이고,
승리 가능성을 높임으로써 여소야대를 해결할 수 있다.

• Thank You •

이상으로 발표를 마칩니다. 감사합니다 :)