기후위기 현황 및
관련 예산과 입법 분석

5조

2022년 9월 30일

이승기 김예진 박종서

목차

1. 주제 및 배경과 필요성

주제: 기후위기 현황 및 관련 예산과 입법 분석

[배경]

- 지속적인 탄소 배출과 환경 파괴에 따른 지구온난화로 세계 곳곳에서 이상기후변화가 일어나고 있다.

- IPCC 6차 보고서에 따르면 지구 온난화 수준이 가까운 시일 내에 1.5°C에 이른다면 여러 기후 위험의 증가를 피할 수 없고 인간 사회와 생태계에 다양한 리스크가 초래될 것이라고 예측되었다.

- 빠르게 변화하는 기후위기 속에서 온실가스 배출량 제어 및 기후변화대응이 무엇보다 절실한 때임을 직감했다.

- 여러 나라와 마찬가지로 우리나라 또한 기후위기에 효과적으로 대응하기 위해서는 제도적 기반이 중요하다.

[목표]

1. 현재 전세계적으로 겪고 있는 기후위기 분석

2. 대한민국 정부 예산 환경 분야에서의 기후변화 대응의 중요도와 변화 분석

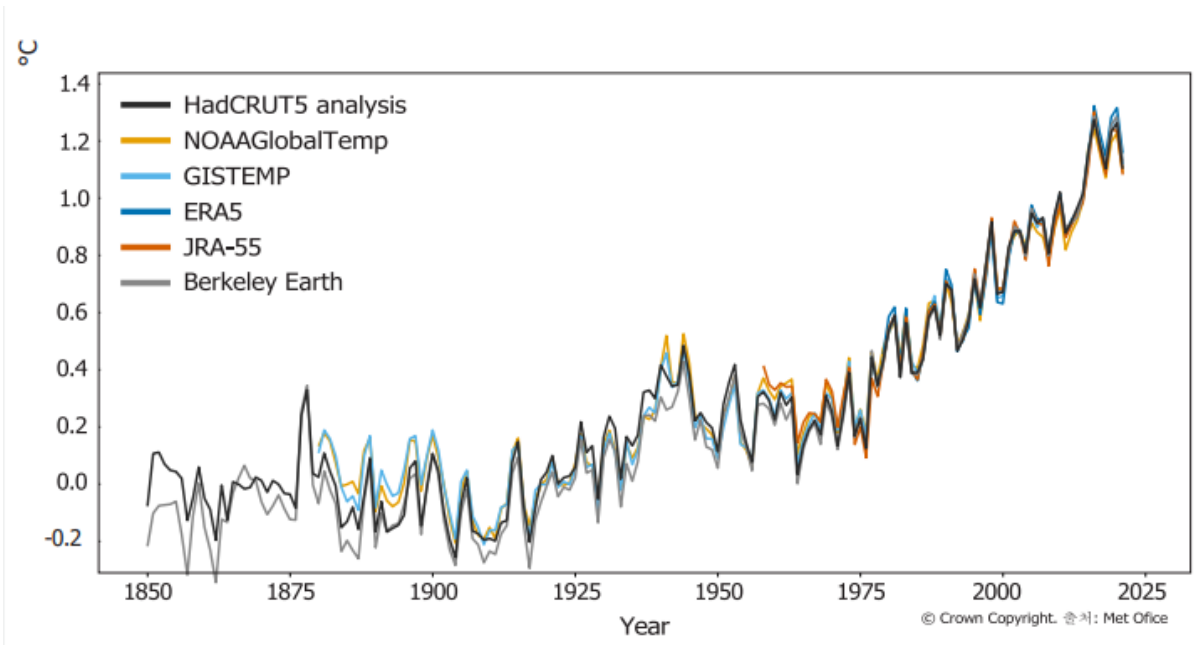
3. 국내에서의 기후변화 관련 입법 현황 분석을 통해 현재 우리나라가 기후변화에 어떻게 대응하고 있는지 파악

4. 앞으로 우리나라가 기후변화 대응에 있어서의 방향성 제시

2. 관련 분석(연구) 사례

> 전세계 및 국내 기후변화 현황 및 분석

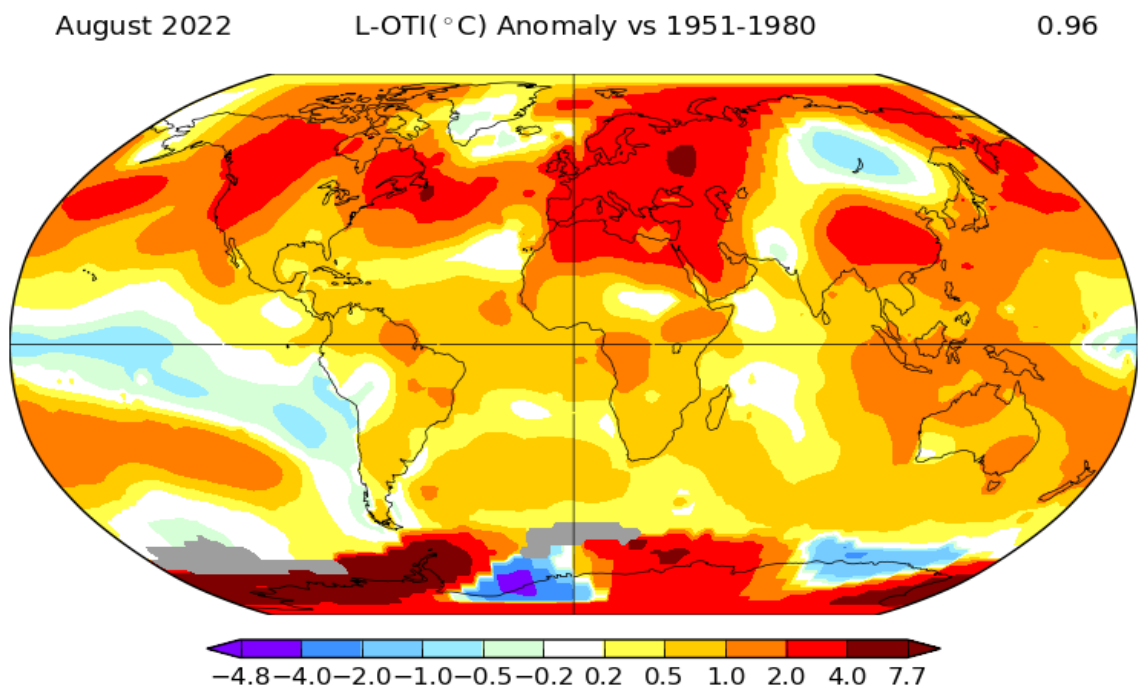
[그림1] 지구의 평균 기온 변화



[1]

- 6개의 세계온도 데이터 세트와 비교했을 때 2021년 전세계 연간 평균 온도는 1850-1900년 전세계 연간 평균 온도보다 1.11 ± 0.13 °C 더 높았던 것을 확인할 수 있었다.
- 지구의 평균 기온은 상승하고 있고 근래 들어서 가속화되고있다는 것을 알 수 있다.

[그림2] 2022년 지구의 근접 표면온도 변화



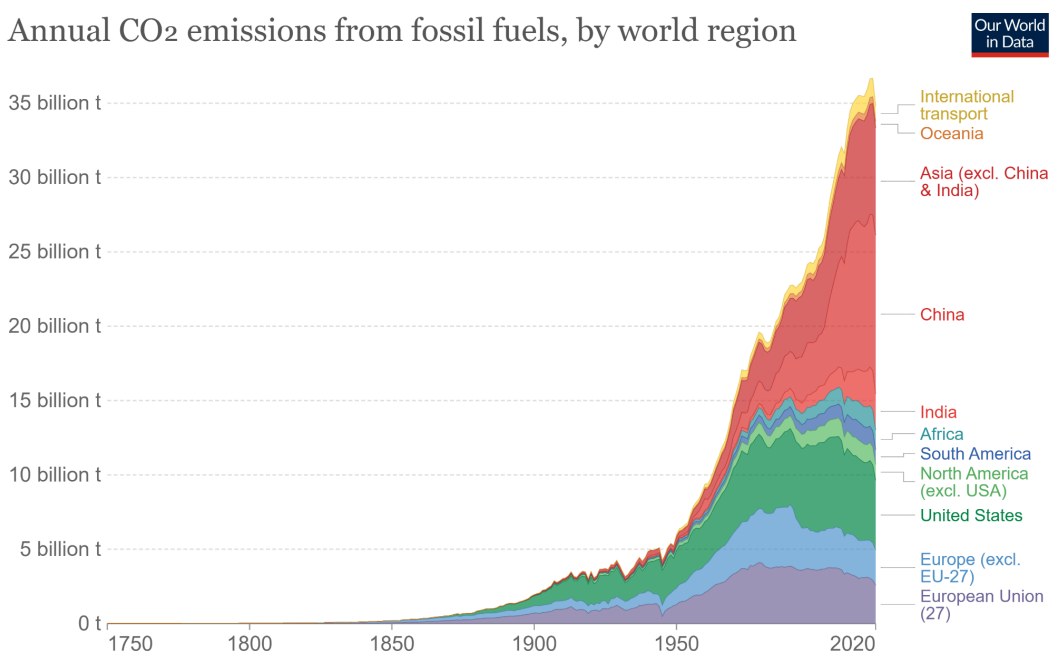
[2]

- 1951년~1980년 평균 지구 표면 온도 대비 2022년 8월의 지구 표면 온도의 차이를 시각화 한 결과, 색이 붉을수록 해당 지역의 온도가 크게 올랐음을 의미하고, 푸른색이 짙을 수록 온도가 떨어졌음을 의미한다.
- 북아시아, 호주, 남아프리카, 북아메리카 등 일부 지역을 제외하고 1951-1980년 근접 표면 온도 평균 대비 2022년 대부분의 지역의 근접 표면 온도는 상승했고, 이를 통해 지구가 점점 뜨거워지고 있다는 사실을 알 수 있다.

---

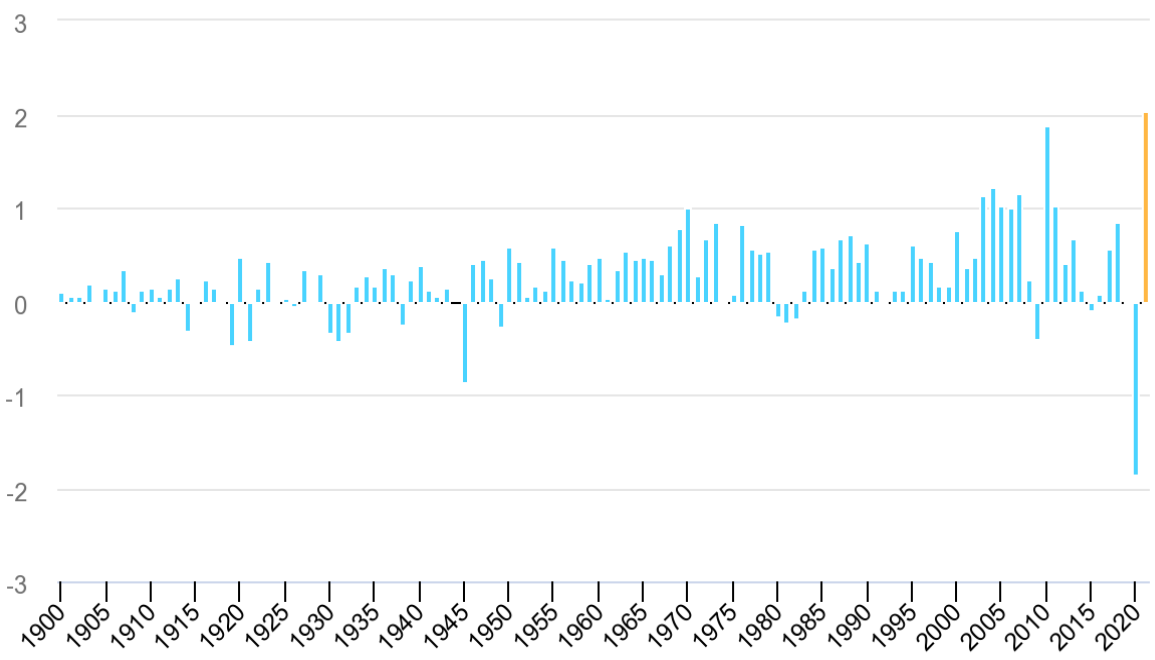
[1] 출처: 2021년 전 지구 기후현황 보고서, WMO(세계기상기구)
[2] 출처: NASA GISA: https://data.giss.nasa.gov/gistemp/maps/

[그림3] 나라별 화석연료의 CO2 배출량



- 산업혁명 이후 주요 연료로 사용된 화석연료의 CO2 배출량은 1950년대 이후 폭발적으로 증가했으며, 2020년 기준 화석연료를 사용해 이산화탄소를 가장 많이 배출한 국가는 중국,미국, EU, 인도가 있다.
- 위 나라들 중 중국을 제외한 나라는 2015년 파리 기후변화협약 이후 온실가스 감축을 목표로 배출량을 감소시키는데 노력하기 시작했고, 2018년 이후 CO2 배출량이 감소하고 있다. 석탄의존도가 높은 중국의 배출량은 크게 줄지 않고 있는 상황이다.

[그림4] 에너지 및 산업공정에서의 CO2 배출량

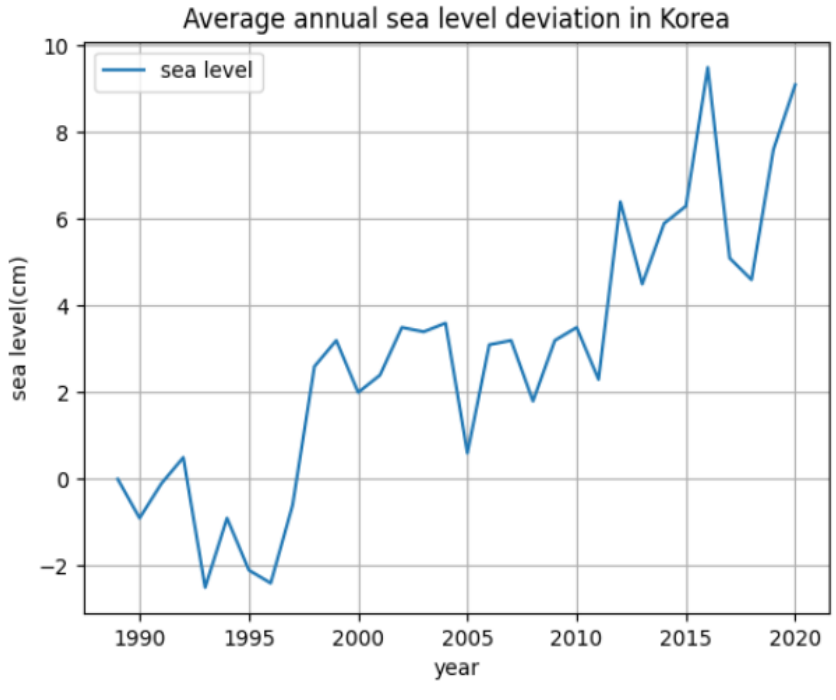


- 코로나19로 인해 산업활동이 줄어들었던 2020년을 제외하고 에너지 및 산업공정에서의 CO2 배출량은 감소보다는 지속적으로 발생하고 있다.
- 2021년부터는 배출량이 코로나 전보다 더 증가하고 있는 추세로 온난화가 진행되는데에 영향을 주고 있다.

---

[3] 출처: https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions
[4] 출처: https://www.iea.org/data-and-statistics/charts/annual-change-in-co2-emissions-from-energy-combustion-and-industrial-processes-1900-2021

[그림5] 국내 평균 해수면 상승량[5]



- **2021년 대한민국 해양의 해수면은 지구온난화의 영향으로 1989년에 비해 약 9.5cm 증가했다.**
- **위 추세대로 해수면이 상승한다면 2050년에는 여의도 면적 대비 83.75배인 242.86㎢ 만큼 국토가 감소하게 된다.**

기후위기에 대응한 국가적 노력

- **전세계적으로 기후위기의 심각성을 직감해 1997년 교토의정서와 2015년 파리협정 등 다양한 범세계적 노력을 해오고있다.**
- **이러한 노력에도 불구하고 오늘날 기후위기에 대한 심각성은 나날이 높아지는 상황에 따라 2018년 추가 협정과 블랙록의 2050 무탄소 선언 등 이전보다 더 강화된 기후위기에 대한 노력이 이어지고 있다.**

[5]출처: http://www.climate.go.kr/home/09_monitoring/main

3. 관련 기술 소개

- 데이터 수집 방식 : OPEN API와 Python을 이용한 크롤링
- 데이터 처리를 위해 사용한 IDE : Colaboratory, Pycharm
- tableau를 통한 전처리한 데이터 시각화 구현

※ 사용한 Python Package

- WordCloud를 이용한 워드클라우드 시각화
- BeautifulSoup, requests를 이용한 크롤링
- re 정규표현식을 이용한 데이터 전처리
- pandas를 이용한 데이터프레임 생성
- plotly, matplotlib를 이용한 그래프 시각화
- konply를 이용한 한글 형태소 분석
- PyPDF2를 이용한 PDF 텍스트 추출
- Counter를 이용한 데이터 개수 파악
- os를 이용한 경로 이동
- glob을 이용한 특정 조건을 만족하는 파일 리스트 작성

4. 분석 과정 설명

> 환경 관련 예산 데이터 분석

[데이터 수집 및 전처리]

: 열린재정에서 공개한 환경 분야 예산을 수집하고 이를 통해 환경 분야 예산 현황과 최근 10년간의 예산 변화를 확인하고자 open api를 통해 데이터를 수집하였다.

```
#open api 연결 및 데이터 전처리를 위한 패키지
import requests, bs4
from urllib.request import Request, urlopen
from urllib.parse import urlencode, quote_plus, unquote
import json
from pandas import json_normalize
#Open Api 통해 json파일 가져오기
url = "https://openapi.openfiscaldata.go.kr/ExpenditureBudgetInit5"
pSize='1000'  # 페이지당 요청 수
FSCL_YY='2023'  # 연도
리우='환경' # 분야
queryParams = '?' + urlencode({ quote_plus('Key') : "BDPZS10000448202209201031135CDWOM",
                                quote_plus('Type') : 'json',
                                quote_plus('pIndex') : '1',
                                quote_plus('pSize') : pSize,
                                quote_plus('FSCL_YY') : FSCL_YY,
                                quote_plus('FLD_NM') : '환경' })

response = urlopen(url + queryParams)
json_api = response.read().decode("utf-8")
#json파일 딕셔너리화
json_file = json.loads(json_api)
budget_json = json.loads(json_file)
budget_json
#데이터프레임화
budget_2023_df = json_normalize(budget_json['ExpenditureBudgetInit5'][1]['row'])
#데이터 전처리 - 필요한 열만 남기기
import pandas as pd
budget_2023_df.drop(['ACCT_NM','FLD_NM','BZ_CLS_NM','FIN_DE_EP_NM', 'ACTV_NM',
'SACTV_NM','Y_YY_DFN_MEDI_KCUR_AMT'], axis=1, inplace=True)
budget_2023_df.columns = ['회계연도','소관명','회계명','부문명','프로그램명','예산(천원)']

#open api부터 데이터프레임화까지 함수화
def budget(a):
  queryParams = '?' + urlencode({ quote_plus('Key') : "BDPZS10000448202209201031135CDWOM",
                                  quote_plus('Type') : 'json',
                                  quote_plus('pIndex') : '1',
                                  quote_plus('pSize') : pSize,
                                  quote_plus('FSCL_YY') : a,
                                  quote_plus('FLD_NM') : '환경' })
  response = urlopen(url + queryParams)
  json_api = response.read().decode("utf-8")
  json_file = json.loads(json_api)
  budget_json = json.loads(json_file)
  globals()[f'budget_{a}_df'] = json_normalize(budget_json['ExpenditureBudgetInit5'][1]['row'])
  globals()[f'budget_{a}_df'].drop(['ACCT_NM','FLD_NM','BZ_CLS_NM', 'FIN_DE_EP_NM', 'ACTV_NM',
'SACTV_NM','Y_YY_MEDI_KCUR_AMT'], axis=1, inplace=True)
  globals()[f'budget_{a}_df'].columns = ['회계연도','소관명','회계명','부문명','프로그램명','예산(천원)']
```

```
#2011년부터 2022년까지 데이터 가져오기
for i in range(2010, 2023):
      budget(i)
#데이터프레임 합치기
budget_sum_df = pd.concat([budget_2011_df, budget_2012_df, budget_2013_df, budget_2014_df, budget_2015_df,
budget_2016_df, budget_2017_df, budget_2018_df, budget_2019_df, budget_2020_df,budget_2021_df,budget_2022_df,
budget_2023_df])
#데이터프레임 실행
budget_sum_df.head()
```

	회계연도	소관명	회계명	부문명	프로그램명	예산(천원)
0	2011	농림수산식품부	수산발전기금	해양환경	해양환경보전	2300000
1	2011	농림수산식품부	수산발전기금	해양환경	해양환경보전	10600000
2	2011	농림수산식품부	수산발전기금	해양환경	해양환경보전	3000000
3	2011	환경부	일반회계	환경일반	회계간거래(전출금)	2816602911
4	2011	환경부	농어촌구조개선특별회계	상하수도·수질	상하수 및 토양지하수 관리	82537000

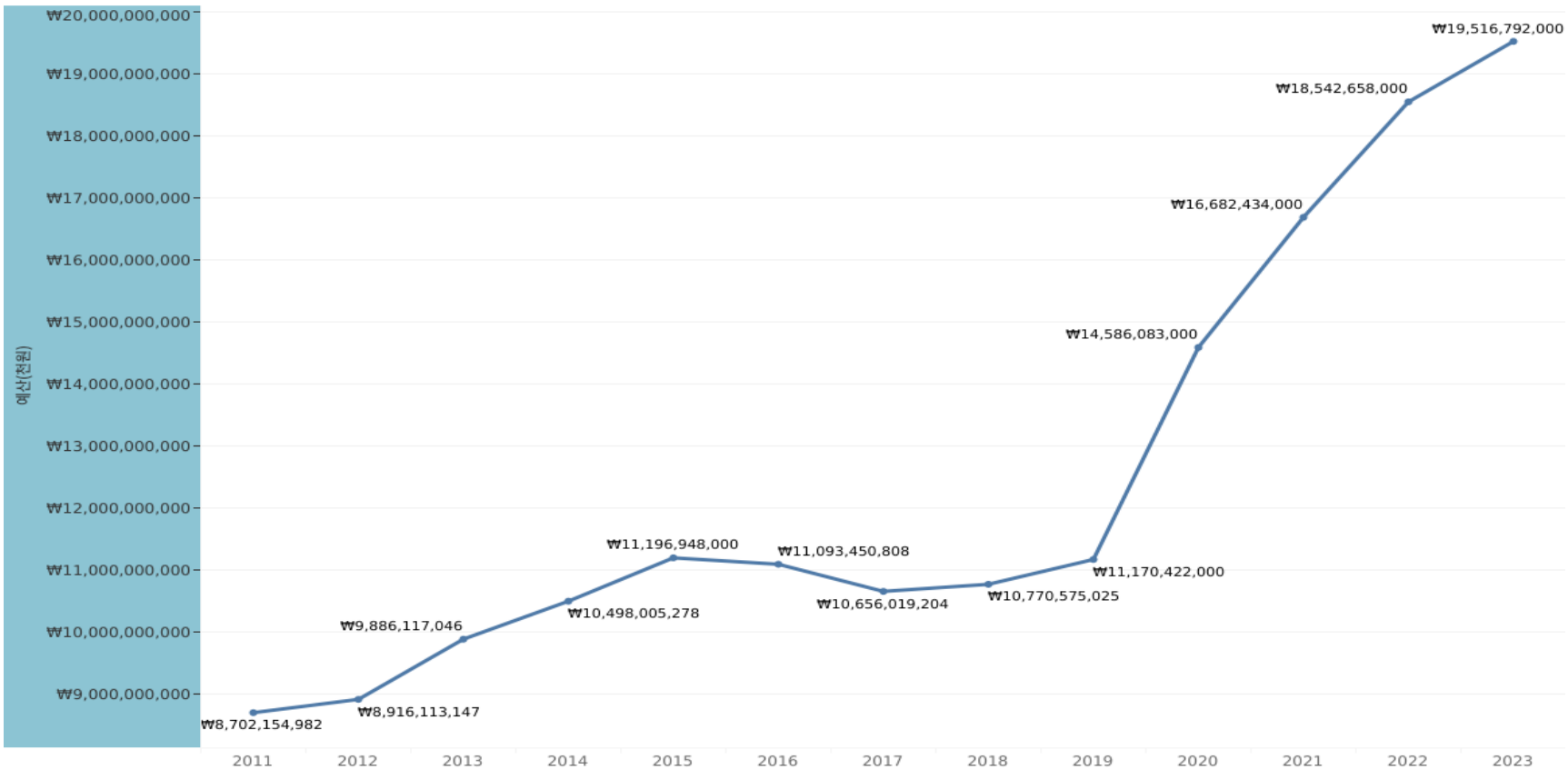
```
# csv파일로 저장
budget_sum_df.to_csv("budget_sum.csv", encoding='cp949')
```

[데이터 시각화]

: 데이터 전처리를 통해 저장한 파일을 데이터 시각화 BI 인 태블로(Tableau)를 이용해 다양한 시각화 결과물을 도출하였다.

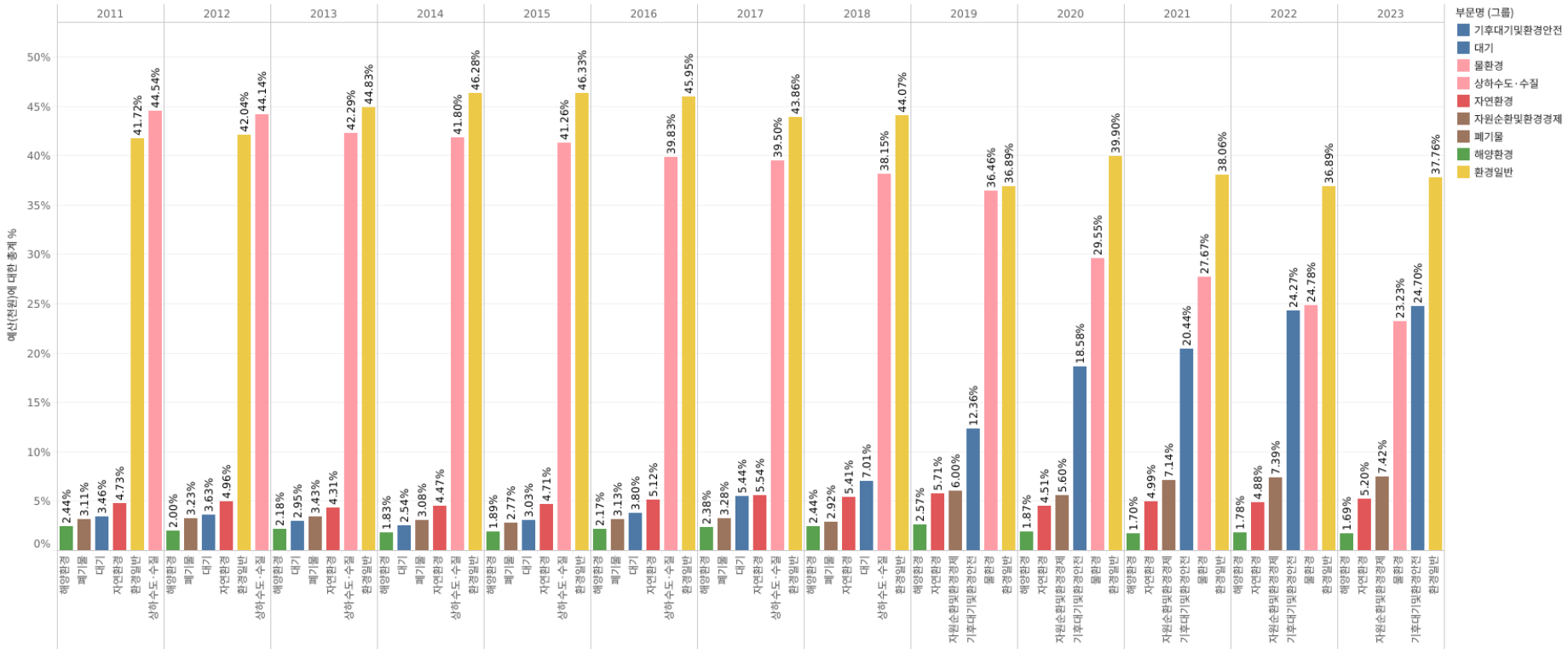
- 태블로 링크: https://public.tableau.com/views/_16638307333170/sheet2?:language=ko-KR&:display_count=n&:origin=viz_share_link

- 연도별 환경 분야 전체 예산 변화



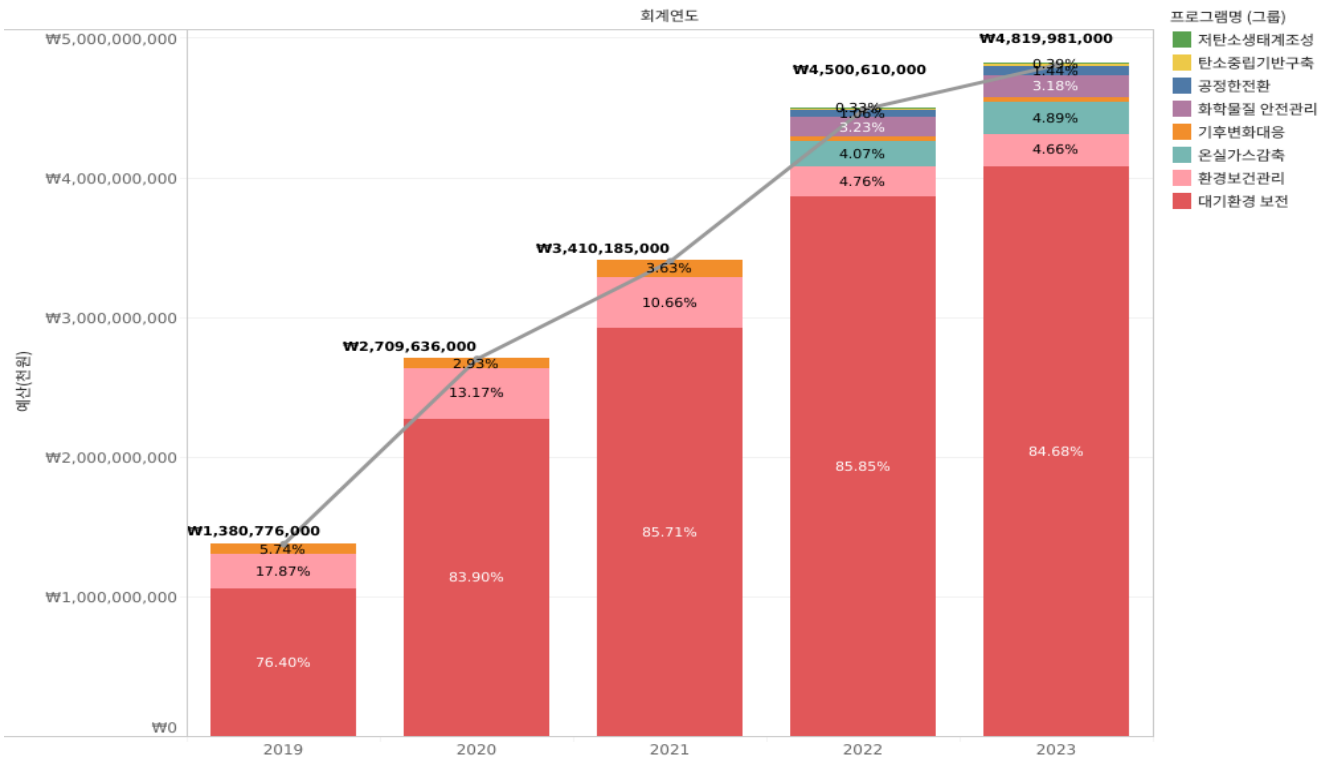
- 국제사회의 기후변화 대응노력에 동참하기 위해 2016년 파리협정에 가입하였다. 그에 따라 '2030 국가 온실가스감축 기본로드맵'을 수립하였고, 2018년 7월에는 국가 온실가스 감축목표의 이행가능성을 높이는데 초점을 두어 '2030 국가 온실가스감축 기본로드맵 수정안'을 마련하였다.
- 또한 2019년부터 미세먼지의 심각성을 깨닫고 관련 예산을 대폭 증액하면서 2020년 환경 분야 예산은 전년대비 매우 큰 증가폭을 보였다.
- 또한 기후위기에 대한 관심이 증가함에 따라 이후에도 환경 분야 관련 예산은 꾸준히 상승 중이며 최종적으로 2023년 정부의 환경 분야 관련 예산은 2011년 예산 대비 124% 증가했다.

- 환경 분야 부문별 예산 비율



- 일반지출에 속하는 환경일반 부문을 제외하고 2022년까지는 물환경에 관련된 부문이 환경 분야 중 제일 높은 예산 비중을 차지했다.
- 2018년 이후 기후대기 및 환경안전에 대한 중요성이 강조에 따라 예산 비중이 점차 상승면서 2023년 정부예산안에서는 기후대기 및 환경안전 부문이 물환경 부문의 예산을 뛰어넘게 되었다.

- 기후대기환경안전 부문별 프로그램 예산 비율



- 2018년까지 ‘대기’ 부문으로 분류되었던 예산을 2019년 부터 ‘기후대기 및 환경안전’ 부문으로 개편하여 세부 프로그램별 예산을 편성했다.
- 대기환경 보전 프로그램이 가장 큰 비중을 차지하는 가운데 기후위기 변화대응, 탄소중립기반구축 및 온실가스 감축에 관련한 예산을 새롭게 편성하면서 점차 그 비중을 늘려가고 있다.

> 의안 데이터 분석

1) 국내 기후변화 의안수 선 그래프

기후변화 의안 데이터는 의안명에 '기후'라는 키워드가 들어간 의안들의 목록을 공공데이터 포털 국회사무처_의안 정보 OPEN API를 이용하여 수집하였다.

```
import re
from bs4 import BeautifulSoup
import requests
import pandas as pd
import plotly.graph_objects as go
import plotly.io as pio
```

[데이터 전처리]

```
# 공공데이터 포털 국회사무처_의안 정보 오픈 API를 이용하여 크롤링
url = 'http://apis.data.go.kr/9710000/BillInfoService2/getBillInfoList'
params ={'ServiceKey' :
'Ws7z6yKz623hbrHshOvor+6YEclNv4A2kyaiwVd97XGoZpsn2TS41ZzgqzThFlqYnOaIYoPDsErGGbjc1/QIog==','numOfRows' :
'100', 'pageNo' : '1', 'ord' : 'A01', 'start_ord' : '18', 'end_ord' : '21', 'bill_name' : '기후'}
response = requests.get(url, params=params) # UTF-8 코드 형식로 데이터가 불러와짐
soup = BeautifulSoup(response.content.decode('utf-8'), 'xml') # 원활한 작업을 위해 디코딩
propose_date = soup.select('proposeDt')# <proposeDt>~</proposeDt> 부분에 있는 제안일자 데이터만 추출

# 불필요한 텍스트 제거
propose_date = str(propose_date) # 정규식을 적용하기 위하여 bs4.element.ResultSet -> str
rule1 = re.compile('(?<=\<proposeDt>)(.*?)(?=<\/proposeDt>)') # rule에 정규식 표현식 컴파일
date = rule1.findall(propose_date)# <proposeDt>, </proposeDt>를 제거하여 그 사이에 있는 제안일자 데이터만 추출
print(date)

# 국내 기후변화 의안수 데이터프레임 생성을 위한 제안연도와 의안수 리스트 생성
propose_year = [] # 제안연도 리스트
for i in date:
    j = i.split('-')[0] # 제안일자의 연,월,일 데이터중에서 연 데이터만 추출
    propose_year.append(j)
print(propose_year)
vacant_year = ['2010','2015','2018'] # 당해에 의안 발의가 없어서 결측치인 연도는 수동으로 따로 리스트에 추가
propose_year.extend(vacant_year)
year = sorted(list(dict.fromkeys(propose_year).keys()))# 제안연도의 데이터에서 중복이 없는 year 리스트 따로 생성
print(year)
propose_count = [] # 각 연도별 의안 개수를 propose_count 리스트에 원소로 추가
for j in year:
    if j in vacant_year: # 의안 발의가 없어서 결측치였던 연도에 해당하는 의안수는 0으로 처리
        propose_count.append(0)
    else:
        propose_count.append(propose_year.count(j))
print(propose_count)
```

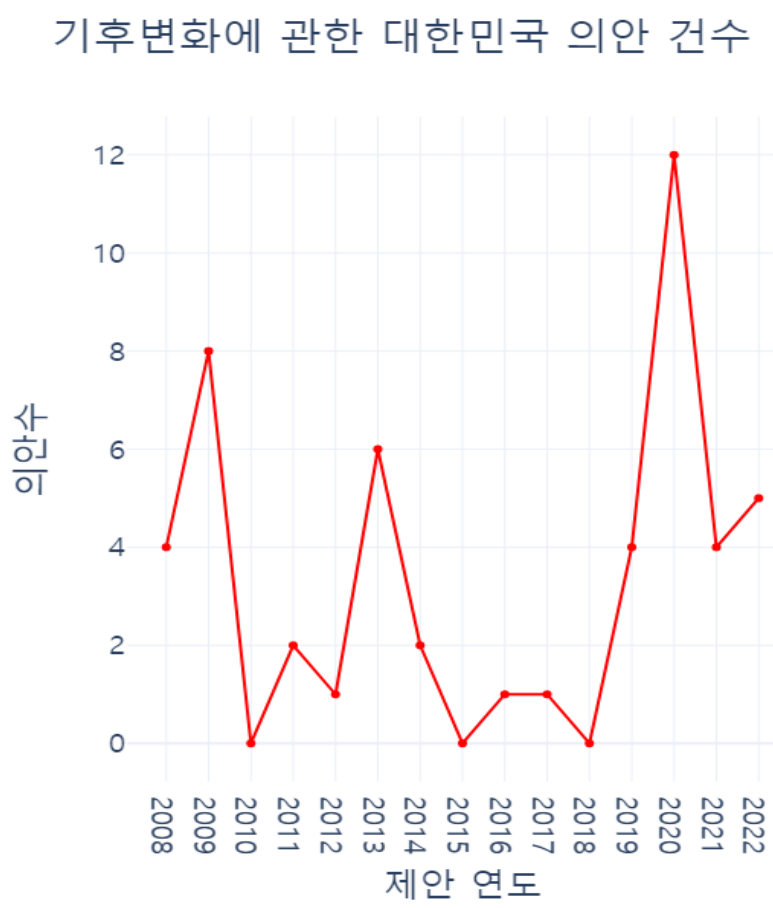
```
['2022-09-16', '2022-08-31', '2022-05-13', '2022-02-28', '2022-01-14', '2021-12-14', '2021-08-25', '2021-06-18', '2021-04-23', '2020-12-18', '2020-12-01', '2020-11-13',
 '2020-11-11', '2020-11-09', '2020-09-22', '2020-09-11', '2020-08-03', '2020-07-28', '2020-07-08', '2020-07-02', '2020-06-30', '2019-09-25', '2019-08-26', '2019-06-04',
 '2019-04-16', '2017-07-27', '2016-12-20', '2014-12-01', '2014-11-05', '2013-07-02', '2013-06-28', '2013-06-11', '2013-05-01', '2013-04-02', '2013-04-02', '2012-09-21',
 '2011-12-27', '2011-06-28', '2009-12-07', '2009-12-01', '2009-09-16', '2009-08-26', '2009-08-26', '2009-04-01', '2009-02-12', '2009-01-14', '2008-11-25', '2008-11-07',
 '2008-08-26', '2008-06-26']
['2022', '2022', '2022', '2022', '2022', '2021', '2021', '2021', '2021', '2020', '2020', '2020', '2020', '2020', '2020', '2020', '2020', '2020', '2020', '2020',
 '2019', '2019', '2019', '2019', '2017', '2016', '2014', '2014', '2013', '2013', '2013', '2013', '2013', '2013', '2012', '2011', '2011', '2009', '2009', '2009', '2009',
 '2009', '2009', '2009', '2009', '2008', '2008', '2008', '2008']
['2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
[4, 8, 0, 2, 1, 6, 2, 0, 1, 1, 0, 4, 12, 4, 5]
```

[데이터 시각화]

```
# 국내 기후변화 의안수 데이터프레임 생성
data = {"제안 연도":year, "의안수":propose_count}
table = pd.DataFrame(data)
print(table)
```

	제안 연도	의안수
0	2008	4
1	2009	8
2	2010	0
3	2011	2
4	2012	1
5	2013	6
6	2014	2
7	2015	0
8	2016	1
9	2017	1
10	2018	0
11	2019	4
12	2020	12
13	2021	4
14	2022	5

```
# 국내 기후변화 의안수 선 그래프 생성
pio.templates.default = "plotly_white"
graph = go.Scatter(x=table['제안 연도'], y=table['의안수'], line={'color':'red','width':2})
layout = go.Layout(title='기후변화에 관한 대한민국 의안 건수', font={'family':'Malgun Gothic',
'size':20},xaxis={'title':'제안 연도'},yaxis={'title':'의안수'},width=600,height=700)
fig = go.Figure(data=graph, layout=layout)
fig.show()
```



- 1948년~2007년에는 기후변화 관련 의안이 발의 되지 않아 2008년 부터의 의안을 조사하였다.
- 제 18대 ~ 제 21대 국회(2008년~2024년)에서 발의한 의안들 중에서 의안명에 '기후'라는 단어가 들어간 의안들을 대상으로 조사한 결과, 총 50건중 3건의 위원 추천안을 제외하면 47건의 의안이 발의되었음을 확인할 수 있었다.
- 또한 기후변화와 관련된 의안들이 꾸준히 발의되었던 것이 아니라 각 연도별 상황에 따라서 편차가 큰 경향을 보이며 의안이 발의됐음을 확인했다.
- 특히, 2018년 10월 대한민국 송도에서 '48차 기후변화에 관한 정부간 협의체(IPCC) 총회'가 개최된 이듬해부터 급격한 증가 추세를 보이다 2021년에 급격히 감소한 것을 확인할 수 있었다.

2) 2020~2022 국내 기후변화&코로나 대응책 의안수 선 그래프

기후변화 의안 데이터는 의안명에 '기후'라는 키워드가 들어간 의안들의 목록을 이용하고 코로나 대응책 의안 데이터는 의안명에 '감염병','의료','코로나','검역법'이라는 키워드가 들어간 의안들의 목록을 공공데이터 포털 국회사무처_의안 정보 OPEN API를 이용하여 수집하였다.

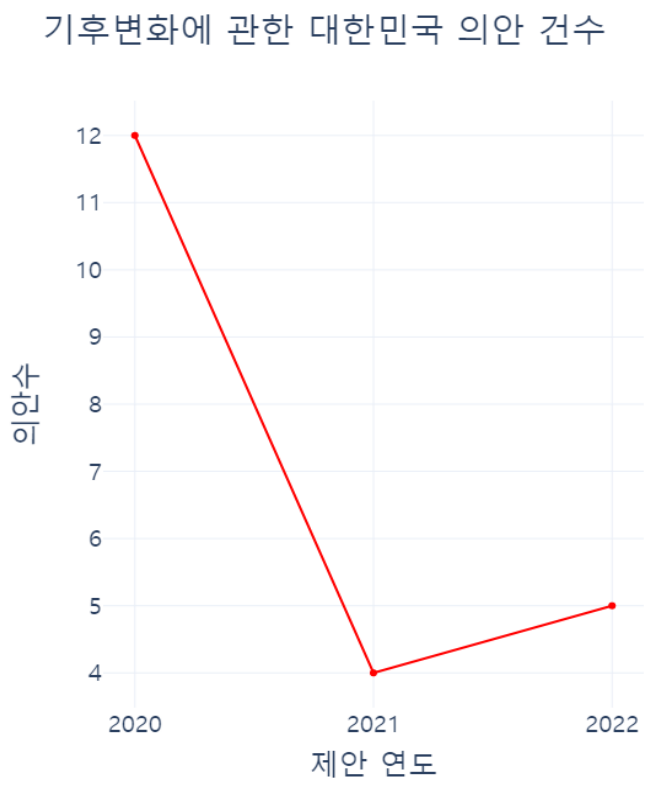
```
import re
from bs4 import BeautifulSoup
import requests
import pandas as pd
import plotly.graph_objects as go
import plotly.io as pio
```

[데이터 전처리]

```
# 각 키워드별 2020,2021,2022년 의안수 리스트 생성 함수
def count_list(keyword, page_num):
    url = 'http://apis.data.go.kr/9710000/BillInfoService2/getBillInfoList'
    params ={'ServiceKey' :
'Ws7z6yKz623hbrHshOvor+6YEclNv4A2kyaiwVd97XGoZpsn2TS41ZzgqzThFlqYnOaIYoPDsErGGbjc1/QIog==', 'numOfRows' :
'100', 'pageNo' : page_num, 'ord' : 'A01', 'start_ord' : '21', 'end_ord' : '21', 'bill_name' : keyword}
    response = requests.get(url, params=params) # UTF-8 코드 형식로 데이터가 불러와짐
    soup = BeautifulSoup(response.content.decode('utf-8'), 'xml') # 한글로 보기 편하게 디코딩
    propose_date = soup.select('proposeDt')

    # 불필요한 텍스트 제거
    propose_date = str(propose_date) # 정규식을 적용하기 위하여 bs4.element.ResultSet -> str로 변환
    rule1 = re.compile('(?<=\<proposeDt>)(.*?)(?=<\/proposeDt>)') # rule에 정규식 표현식 컴파일
    date = rule1.findall(propose_date)

    propose_year = [] # 제안연도 리스트
    for i in date:
        j = i.split('-')[0] # 제안일자의 연,월,일 데이터중에서 연 데이터만 추출
        propose_year.append(j)

    year = sorted(list(dict.fromkeys(propose_year).keys())) # 제안연도의 데이터에서 중복이 없는 year 리스트 따로 생성

    propose_count = [] # 각 연도별 의안 개수를 propose_count 리스트에 원소로 추가
    for j in year:
        propose_count.append(propose_year.count(j))

    # 당해에 의안 발의가 없어서 결측치인 연도는 수동으로 따로 year 리스트에 추가
    # propose_count 리스트에서 year 리스트에 추가된 연도의 인덱스 위치와 일치하는 위치에 0 추가
    if '2020' not in year:
        year.insert(0,'2020')
        propose_count.insert(0,0)
    if '2021' not in year:
        year.insert(1,'2021')
        propose_count.insert(1, 0)
    if '2022' not in year:
        year.insert(2,'2022')
        propose_count.insert(2, 0)

    return propose_count

# 키워드별 각 연도별 의안수를 모두 더한 최종 의안수 리스트 생성 함수
def make_final_countlist(keyword, num):
    for i in range(1,num):
        count.append(count_list(keyword, i))
    return count
```

[데이터 시각화]

```
# 그래프 생성 함수
def make_graph(count, keyword, range, color):
    data = {"제안 연도": ['2020', '2021', '2022'], "의안수": count}
    table = pd.DataFrame(data)

    # 국내 기후변화 의안수 선 그래프 생성
    pio.templates.default = "plotly_white"
    graph = go.Scatter(x=table['제안 연도'], y=table['의안수'], line={'color': color, 'width': 2})
    layout = go.Layout(title=keyword+'에 관한 대한민국 의안 건수', font={'family': 'Malgun Gothic', 'size': 20},
xaxis={'title': '제안 연도'}, yaxis={'title': '의안수'}, width=600, height=700)
    fig = go.Figure(data=graph, layout=layout)
    fig.update_yaxes(range=[0, range])
    fig.show()

# 2020~2022년 국내 기후변화 의안수 그래프 생성
make_graph(count_list('기후',1),'기후변화',15,'red')

# 2020년~2022년 국내 코로나 의안수 그래프 생성
count = []
make_final_countlist('감염병',3)
make_final_countlist('의료',5)
make_final_countlist('코로나',2)
make_final_countlist('검역법',2)

final_count = [0,0,0]
for li in count:
    for j in range(len(final_count)):
        final_count[j] += li[j]

make_graph(final_count,'코로나',250,'black')
```



- 그 원인을 분석해보니 전체적인 의안 발의가 코로나 대응책으로 집중되면서 기후변화에 관한 의안건수가 눈에 띄게 감소했던 것임을 발견할 수 있었다.
- 2020년 대비 2021년 기후변화 의안수는 67% 감소한 비해2020년 대비 2021년 코로나 의안수는 294%나 증가하였다.하지만 2022년 이후에는 코로나 의안 건수가 하락세를 보이면서 기후변화 의안 건수가 다시 상승세를 회복한 것을 확인할 수 있었다.

3) 국내 기후변화 의안 처리결과 현황 막대&원그래프

2102646	계	철도완성차 기후환경시험장 건축 및 풍동시험장비개발에 대한 감사요구안(김희국의원 등 10인)	의원	2020-08-03			소관위심사
ZZ21015	계	국가기후환경회의 위원 추천의 건	기타	2020-07-28			접수
2101645	첫	탈탄소 사회로의 전환을 위한 기후위기대응 및 특별위원회 설치 결의안(강은미의원 등 12인)	의원	2020-07-08	2020-09-24	대안반영폐기	대안반영폐기
2101312	첫	기후위기 비상선언 결의안(김성환의원 등 109인)	의원	2020-07-02	2020-09-24	대안반영폐기	대안반영폐기
2101198	첫	기후위기 비상 대응 촉구 결의안(한정애의원 등 48인)	의원	2020-06-30	2020-09-24	대안반영폐기	대안반영폐기
2022678	첫	기후위기 대응 · 탄소 순배출 제로 목표 설정 촉구 결의안(이정미의원 등 10인)	의원	2019-09-25	2020-05-29	임기만료폐기	
ZZ20288	계	국가기후환경회의 위원 추천의 건	기타	2019-08-26			
2020802	첫	미세먼지 해결을 위한 국가기후환경위원회법안(김삼화의원 등 11인)	의원	2019-06-04	2020-05-29	임기만료폐기	
ZZ20266	계	대통령 직속 '(가칭)국가기후환경회의' 설립 관련 위원 추천의건	기타	2019-04-16			
2008221	첫	기후변화대응법안(송옥주의원 등 47인)	의원	2017-07-27	2020-05-29	임기만료폐기	
2004485	첫	기후변화에 대응한 국민건강관리에 관한 법률안(천정배의원 등 10인)	의원	2016-12-20	2020-05-29	임기만료폐기	
1912761	첫	기후변화건강관리기본법안(민현주의원 등 11인)	의원	2014-12-01	2016-05-29	임기만료폐기	

기후변화 의안 데이터 수집은 의안명에 '기후'라는 키워드가 들어간 의안들의 목록을 공공데이터 포털 국회사무처_의안 정보 OPEN API를 이용하여 진행하였는데

과거 국회의원들이 발의한 의한은 심사진행상태 항목은 결측치 처리되있으면서 의결결과 항목에는 데이터가 정상적으로 기입되어있었고

현재 국회의원들이 발의한 의한은 심사진행상태 항목에는 데이터가 정상적으로 기입이 되어있었지만 의결결과 항목에는 접수중이거나 심사중인 의안에 대해서는 결측치로 처리되어 있어서

과거 국회의원들이 발의한 의안과 현재 국회의원들이 발의한 의안들에 대한 경우를 나눠서 데이터 전처리를 진행하였다.

또한, 기후변화 관련 추천의안은 심사진행상태와 의결결과 항목 모두 결측치 처리되어 있으면서 텍스트를 추출할 주요내용 및 제안이유 내용 또한 결측치로 남아있어서 제외시켰다.

```
import re
from bs4 import BeautifulSoup
import requests
import pandas as pd
import plotly.graph_objects as go
import plotly.io as pio
import matplotlib.pyplot as plt
import matplotlib
```

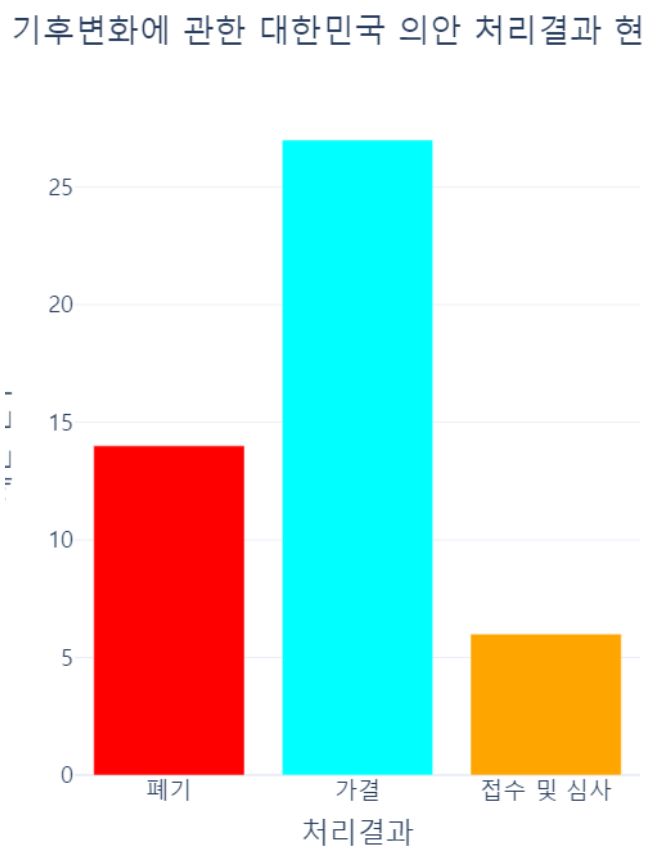
[데이터 전처리]

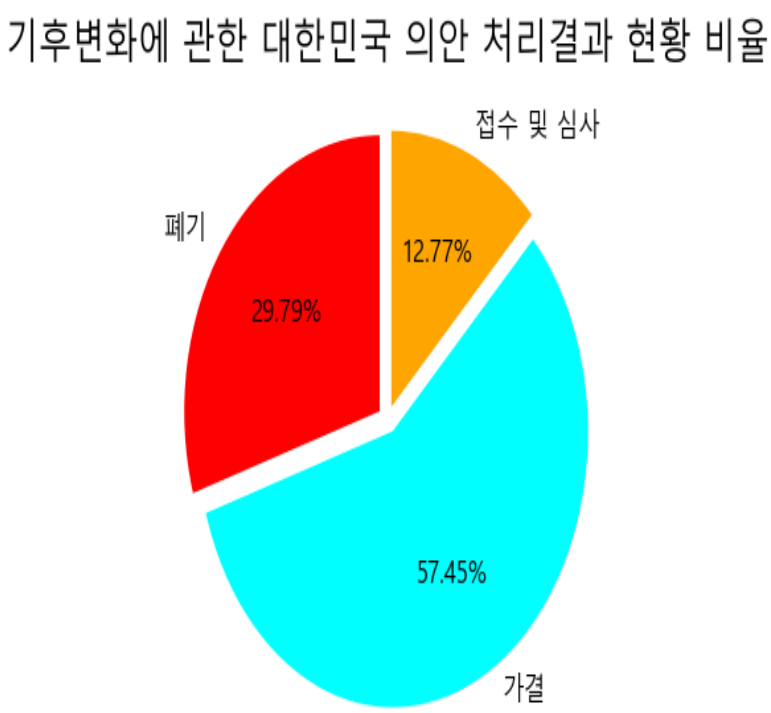
```
# 공공데이터 포털 국회사무처_의안 정보 오픈 API를 이용하여 크롤링
# 현재 국회의원인 21대 국회의원들이 발의한 의안은 심사진행상태 항목('procStageCd')을 이용하고
# 과거의 18~20대 국회의원들이 발의한한 의안은 의결결과 항목('generalResult')을 이용

url = 'http://apis.data.go.kr/9710000/BillInfoService2/getBillInfoList'
params1 ={'ServiceKey' :
'Ws7z6yKz623hbrHshOvor+6YEclNv4A2kyaiwVd97XGoZpsn2TS41ZzgqzThFlqYnOaIYoPDsErGGbjc1/QIog==', 'numOfRows' :
'50', 'pageNo' : '1', 'ord' : 'A01', 'start_ord' : '18', 'end_ord' : '20', 'bill_name' : '기후'}

params2 ={'ServiceKey' :
'Ws7z6yKz623hbrHshOvor+6YEclNv4A2kyaiwVd97XGoZpsn2TS41ZzgqzThFlqYnOaIYoPDsErGGbjc1/QIog==', 'numOfRows' :
'50', 'pageNo' : '1', 'ord' : 'A01', 'start_ord' : '21', 'end_ord' : '21', 'bill_name' : '기후'}

comp1 = 'generalResult'
comp2 = 'procStageCd'
```

```
# 데이터프레임 생성
def make_table(params, comp):

    # OPEN API를 이용하여 데이터 추출
    response = requests.get(url, params=params) # UTF-8 코드 형식로 데이터가 불러와짐
    soup = BeautifulSoup(response.content.decode('utf-8'), 'xml') # 한글로 보기 편하게 디코딩

    bill_name = soup.select('billName')
    bill_result = soup.select(comp) # 선택한 태그 안에 있는 내용만 추출

    # 불필요한 텍스트 제거
    bill_name = str(bill_name) # 정규식을 적용하기 위하여 bs4.element.ResultSet -> str로 변환
    bill_result = str(bill_result)

    rule1 = re.compile('(?<=\<billName>)(.*?)(?=<\/billName>)') # rule1에 정규식 표현식 컴파일
    p_bill_name = rule1.findall(bill_name)

    rule2 = re.compile('(?<=\<generalResult>)(.*?)(?=<\/generalResult>)') # rule2에 정규식 표현식 컴파일 -
generalResult 태그 안에 있는 내용 모두 추출
    rule3 = re.compile('(?<=\<procStageCd>)(.*?)(?=<\/procStageCd>)')  # rule2에 정규식 표현식 컴파일 -
procStageCd 태그 안에 있는 내용 모두 추출

    # 과거 18~20대에 발의된 추천안은 아예 결과가 없는 상태로 조회가 되어 해당 의안의 상태가 bill_result에 없어
bill_name에서만 삭제하면됨
    if comp == comp1:
        p_result = rule2.findall(bill_result)
        for i in p_bill_name:
            if i.find('추천') != -1 or i.find('추천의건') != -1:
                p_bill_name.remove(i)
        data = {"의안명": p_bill_name, "의결결과": p_result}
        global table1
        table1 = pd.DataFrame(data)

    # 현재 21대에 발의된 추천안은 결과가 나와있기 때문에 bill_result와 bill_name에서 모두 삭제해야함
    elif comp == comp2:
        p_result = rule3.findall(bill_result)
        for i in p_bill_name:
            if i.find('추천') != -1 or i.find('추천의건') != -1:
                index = p_bill_name.index(i)
                p_bill_name.remove(i)
                p_result.pop(index)
        data = {"의안명": p_bill_name, "의결결과": p_result}
        global table2
        table2 = pd.DataFrame(data)

make_table(params1, comp1)
make_table(params2, comp2)

# 데이터프레임 의결결과 컬럼 데이터 전처리
table = pd.concat([table1,table2])
table.reset_index(drop=True, inplace=True)

table.loc[table['의결결과'] =='원안가결', '의결결과'] = "가결"
table.loc[table['의결결과'] =='본회의의결', '의결결과'] = "가결"
table.loc[table['의결결과'] =='공포', '의결결과'] = "가결"

table.loc[table['의결결과'] =='임기만료폐기', '의결결과'] = "폐기"
table.loc[table['의결결과'] =='대안반영폐기', '의결결과'] = "폐기"
table.loc[table['의결결과'] =='철회', '의결결과'] = "폐기"

table.loc[table['의결결과'] =='소관위접수', '의결결과'] = "접수 및 심사"
table.loc[table['의결결과'] =='소관위심사', '의결결과'] = "접수 및 심사"
```

[데이터 시각화]

```
# 국내 기후변화 의안 처리결과 현황 막대&원 그래프 생성
result = table['의결결과'].unique()
num1 = len(table.loc[table['의결결과'] =='가결'])
num2 = len(table.loc[table['의결결과'] =='폐기'])
num3 = len(table.loc[table['의결결과'] =='접수 및 심사'])
count_list = [num1,num2,num3]
graph_data = {'처리결과':result, '제안 건수':count_list}
final_table = pd.DataFrame(graph_data)

# plotly를 이용하여 막대 그래프 생성
pio.templates.default = "plotly_white"
graph1 = go.Bar(x=final_table['처리결과'], y=final_table['제안 건수'], marker={"color":["red","auqa","orange"]})
layout = go.Layout(title='기후변화에 관한 대한민국 의안 처리결과 현황',font={'family':'Malgun Gothic', 'size':18},
                   xaxis={'title':'처리결과'},yaxis={'title':'제안 건수'},width=600,height=700)
fig = go.Figure(data=graph1, layout=layout)
fig.show()

# matplotlib을 이용하여 원 그래프 생성
matplotlib.rcParams['font.family'] ='Malgun Gothic'
matplotlib.rcParams['axes.unicode_minus'] = False
graph2 = plt.pie(final_table['제안 건수'], explode=(0.05,0.05,0.05), autopct='%1.2f%%',
labels=final_table['처리결과'],
                 startangle = 90, textprops = {'fontsize': 12}, colors=["red","aqua","orange"])
plt.title('기후변화에 관한 대한민국 의안 처리결과 현황 비율', fontsize=18)
plt.show()
```





- 적은 수에 기간별 큰 편차를 가진 의안 발의 건수도 문제였지만 발의한 의안의 의결 결과 또한 문제라는 것을 확인할 수 있었다.
- 2008년에서부터 발의된 총 47건의 의안들중 27건 만이 가결 되었으며 그 중, 14건이 폐기 상태이며, 6건이 접수중이거나 심사중인 상태인 것을 확인했다.

4) 각 문서별 워드클라우드 및 단어 빈도수 TOP 15 그래프 생성

기후변화 의안 데이터는 의안명에 '기후'라는 키워드가 들어간 의안들의 목록을 공공데이터 포털 국회사무처_의안 정보 OPEN API를 이용하여 데이터를 수집했고
기후변화 보고서 데이터는 관계부처합동으로 제작된 '제3차 국가 기후변화 적응대책'을 PDF로 다운로드받아 데이터를 수집하였으며
기후변화 법령 데이터는 OPEN API로는 폐기된 법령은 접근 불가하고 워드 파일을 다운 받아 진행하는 것은 워드 파일 형식이 신버전(docx)가 아닌 구버전(doc)이라 파이썬 라이브러리로는 적용 불가하여 다운받은 워드 파일들의 내용을 수동으로 텍스트 파일에 각각 저장함으로써 데이터를 수집하였다.
각 문서별 데이터 수집 방식이 달라 워드클라우드 생성 파일과 각 문서의 워드클라우드 및 그래프 생성 파일을 각각 따로 만들어 진행하였다.

```
import matplotlib.pyplot as plt
import numpy as np
from pandas import DataFrame
from PIL import Image
from wordcloud import WordCloud
from collections import Counter
from konlpy.tag import Okt
```

[데이터 시각화]

```
# 워드클라우드 생성 파일
class MakeCloud():

# 워드클라우드 생성 함수
   def wc(self, data, hue1, hue2, name): # 데이터, 색상1, 색상2, 파일명
        engine = Okt() # 한글 형태소 분석에 konlpy 패키지의 Okt 이용
        all_nouns = engine.nouns(data)# 데이터로부터 명사 추출

        value_to_remove = ['기금', '위원회', '정부', '국가', '사항', '계획', '지방자치단체', '것임','국민', '중립',
'장관', '경우', '대통령령', '녹색', '대한'] # 주제와는 연관성이 없어 워드클라우드 생성시 제거할 단어 목록

        # 한글자이거나 제거 목록에 있는 단어는 제외시킨 명사 리스트 생성
        nouns = [n for n in all_nouns if len(n) > 1 and n not in value_to_remove]
        count = Counter(nouns)

        tags = count.most_common(5000)# 단어 빈도수 상위 5000개 조회하여 tags 변수에 저장
        global table # 다른 파일에서 wc 함수를 사용할때 table를 사용할 수 있게 전역변수 처리
        table = DataFrame(tags)
        table.columns = ['단어', '빈도수']
        table.index = table.index + 1 # 순위를 0이 아닌 1위부터로 볼 수 있게 인덱스 조정

        image = Image.open('co2.png') # CO2 텍스트 이미지에 맞게 워드클라우드를 생성할 수 있게 객체에 저장
        mask = np.array(image)

        # 색깔 설정 함수(WordCloud 생성시 색깔을 지정할 수 있는 color_func 파라미터를 이용)
        def color_func(word, font_size, position,orientation,random_state=None, **kwargs):
        # color_func의 기본 파라미터들
                return ("hsl({:d},{:d}%, {:d}%)".format(np.random.randint(hue1, hue2), np.random.randint(80,
100),np.random.randint(10, 60))) # HTML 색상 표기법중 하나인 hsl를 color_func에 이용
        wc = WordCloud(font_path="malgun", background_color="white", max_font_size=500,
                    width=300, height=100, mask=mask, color_func=color_func)
        cloud = wc.generate_from_frequencies(dict(tags))

        plt.imshow(cloud)
        plt.axis('off')
        plt.savefig(name)
        plt.show()
```

```
# 국내 기후변화 의안 워드클라우드 및 단어 빈도수 TOP 15 그래프 생성 파일
import re
from bs4 import BeautifulSoup
import requests
import Cloud
from Cloud import MakeCloud
import plotly.graph_objects as go
import plotly.io as pio
```

[데이터 전처리]

```
# 공공데이터 포털 국회사무처_의안 정보 오픈 API를 이용하여 크롤링
url = 'http://apis.data.go.kr/9710000/BillInfoService2/getBillInfoList'
params ={'ServiceKey' :
'Ws7z6yKz623hbrHshOvor+6YEclNv4A2kyaiwVd97XGoZpsn2TS41ZzgqzThFlqYnOaIYoPDsErGGbjc1/QIog==', 'numOfRows' :
'50', 'pageNo' : '1', 'ord' : 'A01', 'start_ord' : '18', 'end_ord' : '21', 'bill_name' : '기후'}
response = requests.get(url, params=params) # UTF-8 코드 형식로 데이터가 불러와짐
soup = BeautifulSoup(response.content.decode('utf-8'), 'xml') # 한글로 보기 편하게 디코딩
text = soup.select('summary') # summary 태그 안에 있는 내용만 추출

# 불필요한 텍스트 제거
text = str(text) # 정규식을 적용하기 위하여 bs4.element.ResultSet -> str로 변환
text = re.sub('([,.!?·?\"\'])', '', text) # 특수문자 제거
text = re.sub('<.+>', '', text) # <> 태그 제거
```

[데이터 시각화]

```
# 워드클라우드 생성
Cloud1 = MakeCloud()
Cloud1.wc(text,120,140,'WordCloud1.jpg')


# 단어 빈도수 TOP 15 그래프 생성
table = Cloud.table
table = table.loc[1:15,:]
pio.templates.default = "plotly_white"
graph = go.Bar(x=table['단어'], y=table['빈도수'], marker={"color":"green"})
layout = go.Layout(title='국내 기후변화 의안 단어 빈도수 TOP 15',font={'family':'Malgun Gothic', 'size':18},
xaxis={'title':'단어'},yaxis={'title':'빈도수'},width=600,height=700)
fig = go.Figure(data=graph, layout=layout)
fig.show()
```

```
# 국내 기후변화 보고서 워드클라우드 및 단어 빈도수 TOP 15 그래프 생성 파일
from PyPDF2 import PdfFileReader # PyPDF2 = PDF파일 텍스트 추출 패키지
import re
import Cloud
from Cloud import MakeCloud
import plotly.graph_objects as go
import plotly.io as pio
```

[데이터 전처리]

```
# 보고서 PDF파일로부터 텍스트 추출
pdf = PdfFileReader(open('2020.pdf','rb'))
with open('보고서 내용.txt','wb') as file:
    for i in range(5,111):
        pdf_text = re.sub('([,.!•→?○**··↓?\"\'])', '', str(pdf.pages[i].extractText().split('\n')))
        pdf_text = pdf_text.replace('u3000', '')
        pdf_text = pdf_text.replace('u3000500', '')
        file.write(pdf_text.encode('utf-8'))
    with open('보고서 내용.txt', 'r', encoding='utf-8') as f:
        text = f.read()
```

[데이터 시각화]

```
# 워드 클라우드 생성
Cloud2 = MakeCloud()
Cloud2.wc(text,220,240,'WordCloud2.jpg')
f.close()
file.close()

# 단어 빈도수 TOP 15 그래프 생성
table = Cloud.table
table = table.loc[1:15,:]
pio.templates.default = "plotly_white"
graph = go.Bar(x=table['단어'], y=table['빈도수'], marker={"color":"navy"})
layout = go.Layout(title='국내 기후변화 보고서 단어 빈도수 TOP 15',font={'family':'Malgun Gothic', 'size':18},
xaxis={'title':'단어'},yaxis={'title':'빈도수'},width=600,height=700)
fig = go.Figure(data=graph, layout=layout)
fig.show()
```

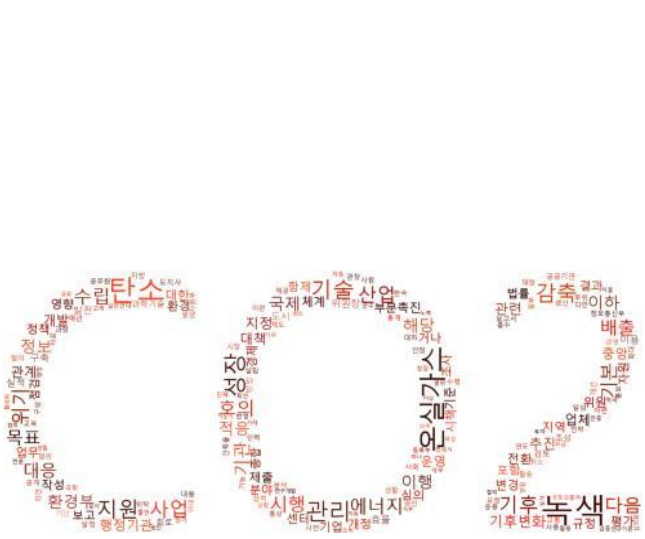
```
# 국내 기후변화 법령 워드클라우드 및 단어 빈도수 TOP 15 그래프 생성 파일
import glob
import os
import Cloud
from Cloud import MakeCloud
import plotly.graph_objects as go
import plotly.io as pio
```

[데이터 전처리]

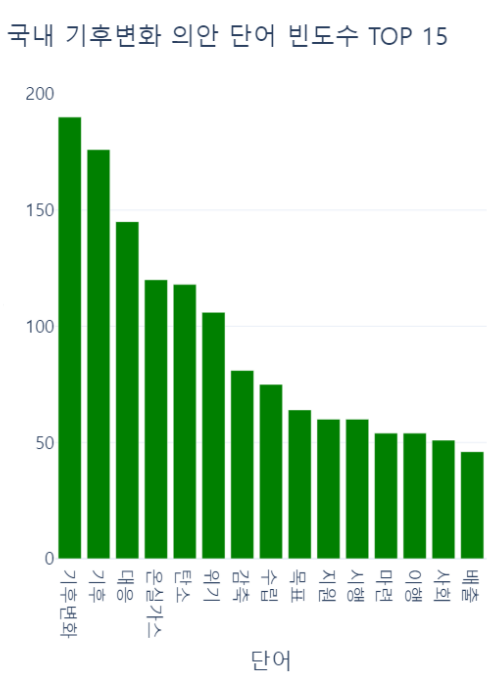
```
os.chdir('C:/Users/playdata/Desktop/bootcamp_project/Project3/law_data') # 법률 파일들이 있는 파일로 작업 환경 이동
read_files = glob.glob("*.txt")

with open('기후 관련 법률 합본.txt', 'wb') as file: # 법률 파일들을 '기후 관련 법률 합본' 이라는 하나의 txt 파일로 합치기
    for f in read_files:
        line = '**********' + f + '**********' + '\n\n'
        file.write(line.encode('utf-8'))
        with open(f,'rb') as infile:
            file.write(infile.read())
    file.close()
    infile.close()

with open('기후 관련 법률 합본.txt', 'r', encoding='utf-8') as f:
    text = f.read()
```

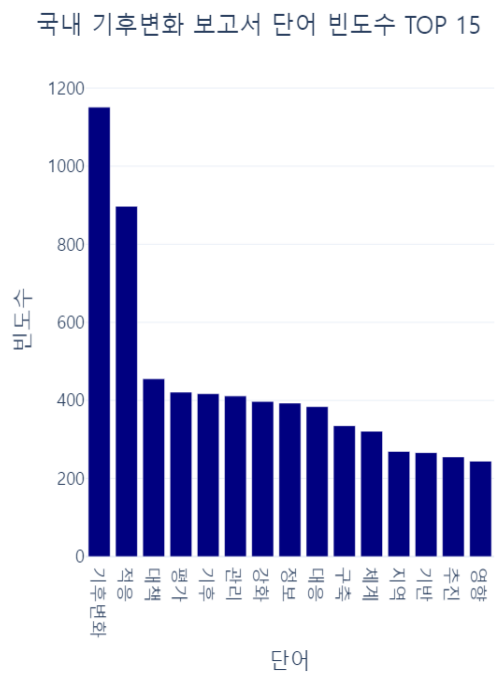
[데이터 시각화]

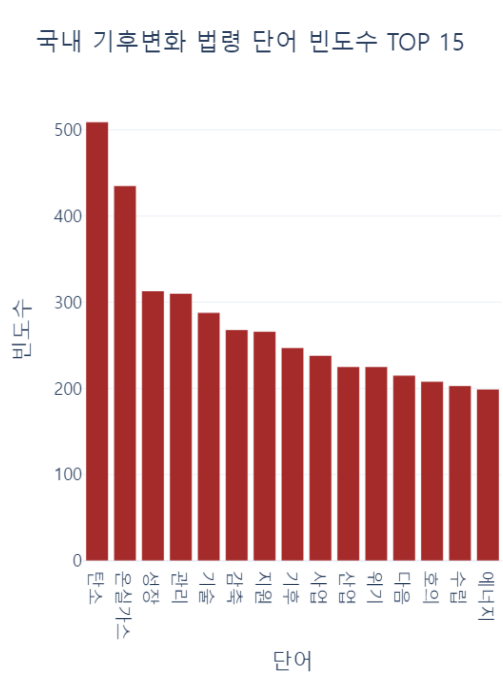
```
# 워드 클라우드 생성
os.chdir('C:/Users/playdata/Desktop/bootcamp_project/Project3') # 작업 환경 원상복귀
Cloud3 = MakeCloud()
Cloud3.wc(text,0,15,'WordCloud3.jpg')
f.close()

# 국내 기후변화 법령 단어 빈도수 TOP 15 그래프 생성
table = Cloud.table
table = table.loc[1:15,:]
pio.templates.default = "plotly_white"
graph = go.Bar(x=table['단어'], y=table['빈도수'], marker={"color":"brown"})
layout = go.Layout(title='국내 기후변화 법령 단어 빈도수 TOP 15',font={'family':'Malgun Gothic', 'size':18},
xaxis={'title':'단어'},yaxis={'title':'빈도수'},width=600,height=700)
fig = go.Figure(data=graph, layout=layout)
fig.show()
```



- 조사한 47건의 의안들(녹색)과 기후변화에 대해 관계부처 합동으로 조사한 국내 보고서(파랑) 그리고 국내 기후변화 관련 법령들(갈색)에서 각각 언급된 단어들의 빈도수를 조사하여 워드클라우드로 시각화해보았다.
- 공통된 부분이 많이 관찰될 것이라 예상했던 것과는 다르게 각 문서 별로 제각각의 특징을 보이고 있었다.







- 워드클라우드로는 각 문서별 특징을 살펴보기 어려워 추가적으로 각 문서별 단어 빈도수 상위 15개를 뽑아 그래프로 구체화 해보았다.
- 의안과 보고서는 발의하거나 조사하는 그 당시 상황을 초점으로 두고 진행하고 법령은 먼 과거에서부터 쌓여온 가결되어 공포된 의안이라 차이가 보였다.
- 의안과 보고서에서는 결과적인 현상에 대한 단어의 빈도수가 높았고 법령에서는 원인에 대한 단어가 빈도수가 높았다.
- 이를 통해 사람들이 과거에는 기후 변화의 원인에 주목하였다면 현재에는 기후 변화로 인한 결과에 더 주목하고있다는 점을 알 수 있었다.

5. 분석결과 및 의의

[결과]

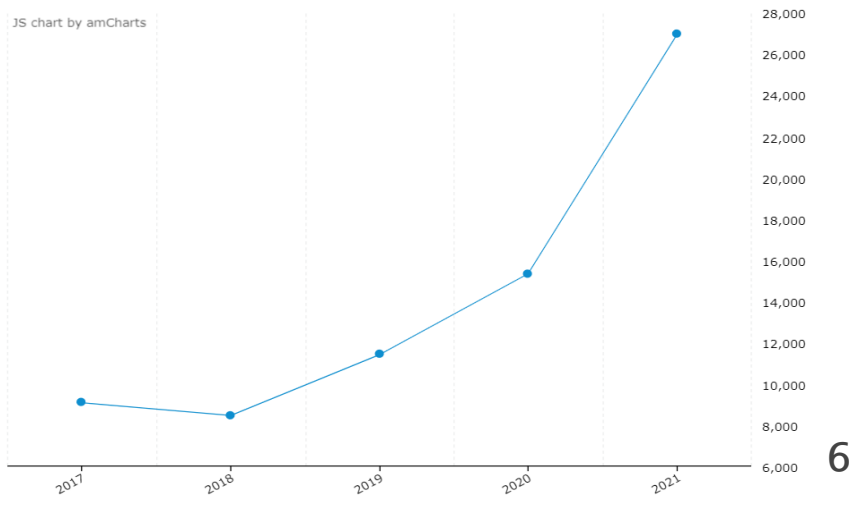
- 1900년대 초에 비해 현재 지구의 평균온도는 약 1°C 상승했다. 지구온난화를 막기 위해 1992년 UN기후변화협약을 시작으로 현재 2015년에 진행한 파리협약까지 다양한 범국가적인 노력이 진행되고있다.

- 이러한 세계의 발걸음에 맞춰 우리나라 또한 2023년 환경예산을 1조 9천억으로 2011년 예산액의 124%에 해당하는 금액을 산정했다.

- 기후변화에 관한 의안, 보고서와 법령은 환경예산이 증가하는 2019년도부터 눈에 띄는 증가를 보였으나, 2020년 발생한 코로나19로 인해 대폭 감소했다가 코로나19 사태가 잠잠해진 2022년부터 다시 그 수가 증가하는 추이를 보이고있다.

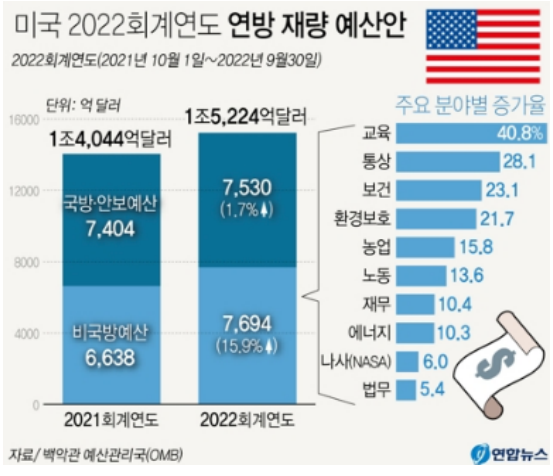
- 기후변화에 관한 의안, 보고서와 법령의 주요 키워드를 통해 과거에는 '탄소'와 '온실가스' 같은 원인 요소에 초점을 두었더라면, 현재는 '기후변화'와 '기후'와 같은 결과에 초점을 두고 있는 것을 알 수 있다.

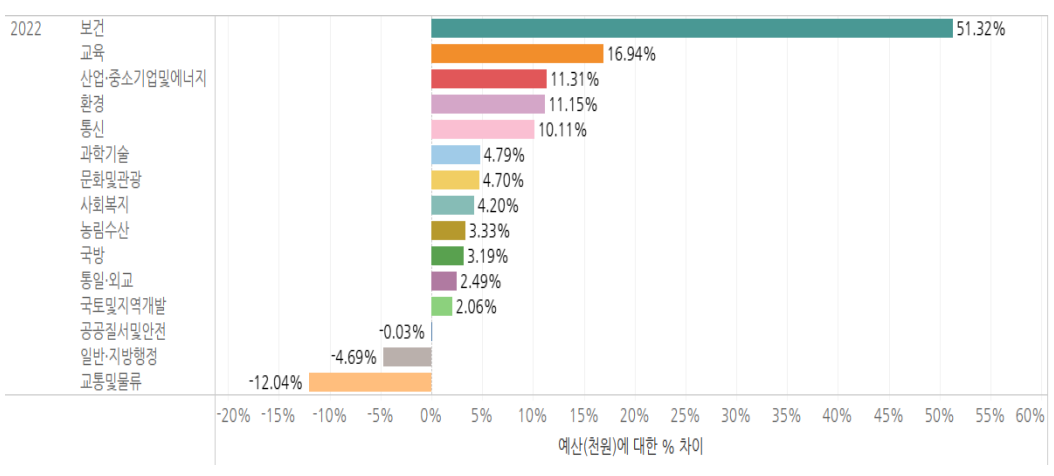
[의의]



6

➢ 2017년부터 '기후변화'와 관련된 기사를 검색한 결과, 2017년 9,069건, 2018년 8,456건에서 2019년 11,413건으로 큰 증가폭을 보였다. 꾸준히 증가하는 검색결과를 통해 '기후변화'에 대한 국민들의 관심도가 높아졌다는 것을 알 수 있다.





➢ 2021년 대비 2022년 미국의 환경보호 예산 증가율은 21.7%를 보여주는 반면, 동기간 한국 환경 예산 증가율은 11.15%로 미국과 대비해 약 10.55%p 낮은 추세를 보인다.

➢ 현재 한국의 환경 예산이 과거에 비해 눈에 띄게 상승한 것은 사실이나, 기후변화에 대한 국민들의 관심이 높아진 만큼 경제 강국들과 대비해 보았을 때 환경 예산 증가 추이를 더 높여도 좋을 것 같다.

---

[6] 출처: 빅카인즈(https://www.bigkinds.or.kr/)

6. 참고문헌

- IPCC 6차 보고서
- ‘기후변화 5대 영향 영역과 적응입법 아젠더’ (국회미래연구원)
- ‘한국기후변화평가보고서’ (환경부)
- ‘제3차 국가 기후변화 적응대책 (관계부처합동)