

1. 프로젝트 주제 및 데이터

- 1) 주제: 국가 지식재산 확보를 위해 공교육에 투자하는 것은 합리적인 방법인가?
- 2) 데이터: 모두 제안서와 동일하나 '주요국_특허_출원_·_등록.csv' 파일의 경우 2000년 이후 데이터로 통계청에서 다시 다운로드 받았다.

2. 입력 데이터 처리

- 1) **pandas** 라이브러리를 이용하여 csv 파일들을 분석에 용이하게 전처리한 데이터들을 새로 생성하였다.
- 2) 통계청에서 받은 파일을 한글 처리를 위해 'csv utf-8'형식으로 다시 저장하였다.
- 3) 공통적으로 각 데이터들의 기간은 2000년대 이후로 적용하고 국가들의 전 연도별 평균값을 계산한 "국가 평균" 행을 생성하였다.
- 4) 국가별 평균이 0인 국가의 데이터는 삭제하였다.
 - (1) 교육단계별_연간_학생1인당_공교육비: **edu_cost.py** 실행 시 전처리 완료된 '교육단계별_연간_학생1인당_공교육비_전처리.csv' 파일이 새로 생성된다. 또한 연도별로 초/중/고등 교육비가 나누어져 있어 세 값의 평균으로 연도별 평균 column들을 새로 생성하고 기존 값은 삭제하였다. 또한 연도별 평균값들의 전체 평균을 국가 평균 column으로 새로 생성하였다.
 - (2) 학업_성취도: **student_score.py** 실행 시 전처리 완료된 '학업_성취도_전처리.csv' 파일이 새로 생성된다. 또한 정보가 거의 없는 2000년도의 정보는 삭제하였다. 그리고 (1)의 과정과 유사하게 남/여/수학/과학/읽기 등으로 나뉜 정보들의 전체 평균으로 연도별 평균 column들을 새로 생성하고 앞의 값들은 삭제하였으며 연도별 평균값들의 전체 평균을 국가 평균 column으로 새로 생성하였다.
 - (3) 주요국_특허_출원_·_등록: **patent.py** 실행 시 전처리 완료된 '주요국_특허_출원_전처리.csv' 두 가지 파일이 새로 생성된다. 또한 데이터를 특허 출원수와 등록수 두 가지로 분리하고 연도별 평균 출원수와 등록수를 계산한 국가 평균 column을 생성하였다.
 - (4) 100만명당_RD_연구개발자: **RD_researcher.py** 실행 시 전처리 완료된 '100만명당_RD_연구개발자_전처리.csv' 파일이 새로 생성된다. 또한 2000년대 이전 column들은 삭제하고 연도별 평균을 계산한 국가 평균 column을 생성하였다.
 - (5) 총_연구원수 : **total_researcher.py** 실행 시 전처리 완료된 '총_연구원수_전처리.csv' 파일이 새로 생성된다. 또한 2000년대 이전 column들은 삭제하고 연도별 평균을 계산한 국가 평균 column을 생성하였다.

3. 문제 해결 방법

- 1) 각 국가별 공교육비, 학업 성취도, 특허 출원 수, 특허 등록수, RD 연구개발자 수, 총 연구원 수 총 6가지 종류의 데이터간의 상관성 분석을 위해 세 가지 비교 방법을 사용하였다.
 - (1) **국가별 비교**: 두 가지 데이터의 국가별 평균 값을 산점도 그래프와 선 그래프로 그려보고 비교하여 두 데이터간의 상관관계를 분석
 - (2) **연도별 비교**: 특정한 국가에서 두 데이터들의 연도별 값 변화를 선 그래프로 그려보고 비교하여 두 데이터간의 상관관계를 분석
 - (3) **전체 변수 비교**: 전체 데이터들의 국가별 평균 값을 산점도 행렬로 그려보고 전체 변수

들의 상관지수를 출력하여 분석

2) 위의 비교 방법을 실행하기 위해 주어진 데이터들을 가공하여 원하는 정보와 형태로 만들어주는 *compareData* 모듈을 만들었다. *compareData* 모듈은 *pandas* 라이브러리를 사용하고, 포함하고 있는 함수들은 다음과 같다.

(1) **findSame(df1(dataframe), df2(dataframe))**: 두 개의 dataframe에서 공통된 국가들의 리스트를 반환한다.

(2) **makesameList(df1(dataframe), df2(dataframe))**: 국가별 선그래프를 그리기 위한 데이터를 제공하는 함수. 두 개의 dataframe에서 공통된 국가들의 국가 평균값을 각각 하나의 리스트에 저장하여 두 리스트를 원소로 하는 하나의 리스트를 반환한다. 이때 선그래프 scale을 맞추기 위해 두 데이터에서 한국의 국가 평균값의 비율을 계산하여 하나의 데이터에 곱했다.

```
def makesameList(df1, df2):
    '''두 개의 dataframe에서 공통된 국가들의 국가 평균 값으로 이루어진 리스트 반환'''
    a=[]
    b=[]
    ratio = 1
    same_country = findSame(df1, df2)
    # 데이터간 scale 맞추기 위해 한국을 기준으로 ratio값 구해 곱함
    ratio = float(df2.loc[same_country]["국가 평균"])/float(df1.loc[same_country]["국가 평균"])
    for i in same_country:
        a.append(round(float(df1.loc[i]["국가 평균"])*ratio))
        b.append(round(float(df2.loc[i]["국가 평균"])))
    return [a,b]
```

그림 1 compareData 모듈의 makesameList 함수

(3) **makesameDf(df1(dataframe), df2(dataframe), xlabel(str), ylabel(str))**: 산점도 그래프를 그리기 위한 데이터를 제공하는 함수. **makesameList** 함수로 얻은 리스트들을 dataframe으로 만들어 반환한다. 이때 데이터의 column이름은 'xlabel'과 'ylabel'으로 입력한다.

(4) **makefullDf(df(dataframe))**: 연도별 비교를 위해 주어진 데이터를 가공하는 함수. 전체 연도에서 하나라도 데이터 결함이 없는 나라들만 추리고 국가 평균 column을 삭제하며 column을 정수 연도로 변경한 dataframe을 반환한다.

```
def makefullDf(df):
    '''데이터 결함이 없는 나라들만 추리고 국가 평균을 삭제한 데이터 프레임 리턴'''
    droplist=[]
    for i in df.index:
        lists = df.loc[i]
        j=0
        while j<len(lists):
            # 하나라도 결함 있는 국가는 삭제 리스트에 포함
            if lists[j]!='-':
                droplist.append(i)
                break
            j += 1
    newdf = df.drop(droplist)
    newdf = newdf.drop(["국가 평균"], axis=1)
    # column명 연도만 표기하는 정수로 변경
    newdf.columns = [int(x.split("년")[0]) for x in newdf.columns]
    return newdf
```

그림 2 compareData 모듈의 makefullDf 함수

(5) **makecountryList(df1(dataframe), df2(dataframe), country(str))**: 연도별 선 그래프를 그리기 위한 데이터를 제공하는 함수. dataframe인 'df1'과 'df2'를 **makefullDf** 함수로 가공하고 두 데이터에 공통으로 포함된 나라인 country의 연도별 데이터를 각각 하나의 리스트에 저장하여 두 리스트를 원소로 하는 하나의 리스트를 반환한다. **makesameList**와 유사하게 데이터 scale을 맞추기 위해 가장 처음값의 비율을 계산하여 하나의 데이터에 곱해준다.

(6) **findSamemore(dflist(list))**: 여러개의 dataframe으로 이루어진 리스트에서 모두 공통된 국가들의 리스트를 반환한다.

(7) **makesameListmore(dflist(list))**: 산점도 행렬 그래프를 그리기 위한 데이터를 제공하는 함수. 비교할 dataframe들의 list를 받아서 전체 data에서 공통된 국가들의 국가 평균 값들을 각각 리스트에 저장하고 각 리스트들을 원소로 하는 하나의 리스트를 반환한다.

(8) **makesameDfmore(dflist(list), labellist(list))**: 산점도 행렬 그래프를 그리기 위한 데이터를 제공하는 함수. **makesameListmore** 함수로 얻은 리스트들을 dataframe으로 만들어 반환한다. 이때 데이터의 column 이름은 'labellist' 리스트 안의 원소들로 한다.

```
def makesameListmore(dflist):
    '''여러개의 dataframe에서 공통된 국가들의 평균 데이터 값으로 이루어진 리스트 반환'''
    a=[]
    for x in range(len(dflist)):
        same_country = findSamemore(dflist)

        for i in same_country:
            for j in range(len(dflist)):
                a[j].append(round(float(dflist[j].loc[i]["국가 평균"])))
    return a

def makesameDfmore(dflist, labellist):
    '''여러개의 dataframe에서 공통된 국가들의 평균 데이터 값으로 이루어진 데이터 프레임 반환'''
    df = pd.DataFrame(makesameListmore(dflist), columns = findSamemore(dflist))
    df.index = labellist
    # 행열 전환하여 리턴 (산점도 그래프 그리기 위해)
    return df.transpose()
```

그림 3 compareData 모듈의 makesameListmore, makesameDfmore 함수

3) **compareData** 모듈을 통해 가공한 데이터들을 그래프를 통해 분석할 수 있게 해주는 **graphs** 모듈을 만들었다. **graphs** 모듈은 **numpy**, **matplotlib**, **seaborn**, **pandas** 라이브러리를 사용하였다. **graphs** 모듈에 있는 함수들은 다음과 같다.

```
# 그래프에서 한글 표시하기 위한 작업
matplotlib.rc('font', family='Malgun Gothic')
matplotlib.rcParams['axes.unicode_minus'] = False

def plotScatter(xlabel, ylabel, df, string):
    '''pyplot과 seaborn으로 산점도 그래프 그리기'''
    sns.regplot(x=xlabel, y=ylabel, data=df)
    plt.xlim(df[xlabel].min()-1, df[xlabel].max()+1)
    plt.grid()
    plt.title("%s %s와 %s 비교" % (string, xlabel, ylabel))
    plt.show()

    z=np.polyfit(df[xlabel], df[ylabel], 1) # 기울기와 절편 확인
    print('기울기:', z[0], '절편:', z[1])

def plotLine(label1, label2, lists, stringlist):
    '''line subplot 동시에 그리기'''
    plt.plot(label1, lists[0])
    plt.plot(label2, lists[1])
    plt.title("%s %s(파랑)와 %s(주황) 비교" % (stringlist[0], stringlist[1], stringlist[2]))
    plt.xticks(rotation=75)
    plt.show()

def plotScattermatrix(df):
    '''여러 변수들에 대한 scatter matrix 그래프 그리기'''
    scatter_matrix(df, figsize=(10, 10));
    plt.tight_layout()
    plt.show()
    # df.corr로 각 변수 간 상관지수 출력
    print(df.corr())
```

그림 4 graphs 모듈 전체 구조

(1) **plotScatter(xlabel(str), ylabel(str), df(dataframe), string(str))**: dataframe인 'df'의 'xlabel' column과 'ylabel' column의 값을 각각 x, y값으로 하는 산점도 그래프와 추세선을 그려 출력하고 추세선의 기울기와 절편을 출력한다. 이때 추세선 기울기는 음/양 여부로 두 변수의 비례관계를 예측할 수 있으나 값의 크기가 두 변수의 상관성을 나타내는 것은 아니다.

(2) `plotLine(label1(list),label2(list),lists(list),stringlist)`: 'lists'[0]과 'lists'[1]에 각각 저장된 두 데이터들을 하나의 선 그래프에 출력하여 두 값들의 변화 추세를 비교 확인할 수 있게 한다. 'lists'[0]은 'label1'을 x값으로 하고 'lists'[1]은 'label2'를 x값으로 한다. 국가별 비교의 경우 'label1'과 'label2'가 두 데이터의 공통 국가 리스트로 동일하다.

(3) `plotScattermatrix(df(dataframe))`: 여러 데이터들이 포함된 한 dataframe의 모든 column들의 상관관계를 볼 수 있는 scatter matrix를 그리고 각 변수 간 상관계수를 표로 출력한다. 이때 상관계수의 크기는 `plotScatter`의 추세선과는 달리 두 변수의 상관성을 나타낸다.

4. 프로그램의 전체적인 구조

: 위의 `compareData` 모듈과 `graphs` 모듈을 이용하여 각각의 방법으로 데이터를 비교하는 프로그램을 작성하였다. (주의) 프로그램 실행을 위해서는 `numpy`, `matplotlib`, `seaborn`, `pandas` 라이브러리가 설치되어 있어야 한다.

1) `plot_act_nations.py`: 두 가지 데이터의 국가별 평균값을 비교하는 프로그램. 데이터 이름을 key로 하고 이름에 대응하는 데이터를 value로 하는 dictionary를 생성했다. 프로그램은 비교 가능한 데이터 목록을 보여주고 비교할 데이터 이름 두 개를 입력받아 두 데이터의 산점도 그래프와 선 그래프를 보여준다. 이때 데이터 목록에 없는 데이터 이름을 입력하거나 동일한 데이터를 두 개 입력하면 오류로 처리하고 다시 입력받으며, 4회 이상 오류 발생시 분석을 중단한다. 또한 “계속 진행하시겠습니까?”라는 질문에 ‘네’라고 답하면 다시 데이터를 입력받아 분석을 진행한다.

```

F RESTART: C:\Users\kimyo\Desktop\project_dataset\plot_act_nations.py
비교 가능한 데이터 목록: ['공교육비', '학업 성취도', '특허 출원수', '특허 등록수', 'RD 연구원수', '총 연구원수']
비교할 데이터를 선택하세요 (첫 번째): 교육비
비교 가능한 데이터가 아닙니다.
비교할 데이터를 선택하세요 (첫 번째): 공교육비
비교할 데이터를 선택하세요 (두 번째): 공교육비
비교 가능한 데이터가 아닙니다.
비교할 데이터를 선택하세요 (두 번째): 학업성취
비교 가능한 데이터가 아닙니다.
비교할 데이터를 선택하세요 (두 번째): 학업성취도
비교 가능한 데이터가 아닙니다.
시도 가능 횟수를 초과하여 분석을 종료합니다.

계속 진행하시겠습니까? ('네' 입력시 계속): 네

비교 가능한 데이터 목록: ['공교육비', '학업 성취도', '특허 출원수', '특허 등록수', 'RD 연구원수', '총 연구원수']
비교할 데이터를 선택하세요 (첫 번째): 공교육비
비교할 데이터를 선택하세요 (두 번째): 학업 성취도
국가별 공교육비 vs 학업 성취도
기울기: 0.059688840946126594 절편: 460.9834779755284

계속 진행하시겠습니까? ('네' 입력시 계속): 아니요
  
```

그림 5 `plot_act_nations.py` 실행 결과 (그래프는 다른 창에 생성됨)

2) `plot_act_years.py`: 특정한 국가의 두 가지 데이터의 연도별 값의 변화를 비교하는 프로그램. `plot_act_nations.py`와 마찬가지로 데이터 이름을 key로 하고 이름에 대응하는 데이터를 value로 하는 dictionary를 생성했다. 프로그램은 비교 가능한 데이터 목록을 보여주고 비교할 데이터 이름 두 개를 입력받은 후 두 데이터에 공통으로 포함되어 비교 가능한 국가들의 목록을 보여준다. 그리고 비교할 국가를 입력받아 해당 국가의 연도별 데이터들의 변화량을 볼 수 있는 선 그래프를 보여준다. 이때 데이터 목록에 없는 데이터 이름을 입력하거나 동일한 데이터를 두 개 입력한 경우, 그리고 비교 불가능한 국가를 입력한 경우를 오류로 처리하고 다시 입력받으며, 4회 이상 오류 발생시 분석을 중단한다. 또한 “계속 진행하시겠습니까?”

라는 질문에 ‘네’라고 답하면 다시 데이터를 입력받아 분석을 진행한다.

3) **plot_act_scattermatrix.py**: 모든 데이터들의 국가별 평균 값들의 상관관계를 분석하는 프로그램. **plot_act_nations.py**와 마찬가지로 데이터 이름을 key로 하고 이름에 대응하는 데이터를 value로 하는 dictionary를 생성했다. 프로그램은 각 국가별 공교육비, 학업 성취도, 특허 출원 수, 특허 등록수, RD 연구개발자 수, 총 연구원 수 총 6가지 종류의 데이터들의 전체 상관관계를 볼 수 있는 산점도 행렬 그래프를 출력하며 각 데이터들 간의 상관계수를 표로 출력한다.

```

F RESTART: C:\Users\kimyo\Desktop\project_dataset\plot_act_scattermatrix.py
각 변수 간 상관지수 확인

```

	공교육비	학업 성취도	특허 출원수	특허 등록수	RD 연구원수	총 연구원수
공교육비	1.000000	0.402963	0.177175	0.167767	0.622297	0.210667
학업 성취도	0.402963	1.000000	0.185303	0.191110	0.687246	0.154919
특허 출원수	0.177175	0.185303	1.000000	0.997750	0.174030	0.951437
특허 등록수	0.167767	0.191110	0.997750	1.000000	0.174355	0.936565
RD 연구원수	0.622297	0.687246	0.174030	0.174355	1.000000	0.144128
총 연구원수	0.210667	0.154919	0.951437	0.936565	0.144128	1.000000

그림 6 plot_act_scattermatrix.py 실행 결과

```

F RESTART: C:\Users\kimyo\Desktop\project_dataset\plot_act_years.py
비교 가능한 데이터 목록: ['공교육비', '학업 성취도', '특허 출원수', '특허 등록수', 'RD 연구원수', '총 연구원수']
비교할 데이터를 선택하세요 (첫 번째): 공교육
비교 가능한 데이터가 아닙니다.
비교할 데이터를 선택하세요 (첫 번째): 공교육비
비교할 데이터를 선택하세요 (두 번째): 특허 출원수
연도별 공교육비 vs 특허 출원수
비교 가능 국가: ['이스라엘', '멕시코', '미국', '네덜란드', '폴란드', '스페인', '오스트레일리아']
국가 선택: 한국
비교 가능한 국가가 아닙니다
국가 선택: 일본
비교 가능한 국가가 아닙니다
국가 선택: 프랑스
비교 가능한 국가가 아닙니다
시도 가능 횟수를 초과하여 분석을 종료합니다.

계속 진행하시겠습니까? ('네' 입력시 계속): 네

비교 가능한 데이터 목록: ['공교육비', '학업 성취도', '특허 출원수', '특허 등록수', 'RD 연구원수', '총 연구원수']
비교할 데이터를 선택하세요 (첫 번째): 공교육비
비교할 데이터를 선택하세요 (두 번째): 특허 출원수
연도별 공교육비 vs 특허 출원수
비교 가능 국가: ['이스라엘', '멕시코', '미국', '네덜란드', '폴란드', '스페인', '오스트레일리아']
국가 선택: 멕시코
멕시코의 연도별 공교육비와 특허 출원수값 비교

계속 진행하시겠습니까? ('네' 입력시 계속): 아니오

```

그림 7 plot_act_years.py 실행 결과 (그래프는 다른 창에 생성됨)

5. 데이터 분석 결과

: 위의 프로그램을 사용해서 “국가 지식재산 확보를 위해 공교육에 투자하는 것은 합리적인

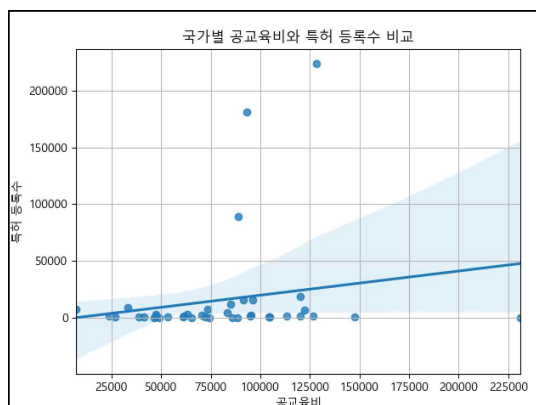


그림 8 국가별 공교육비와 특허 등록수 산점도 그래프

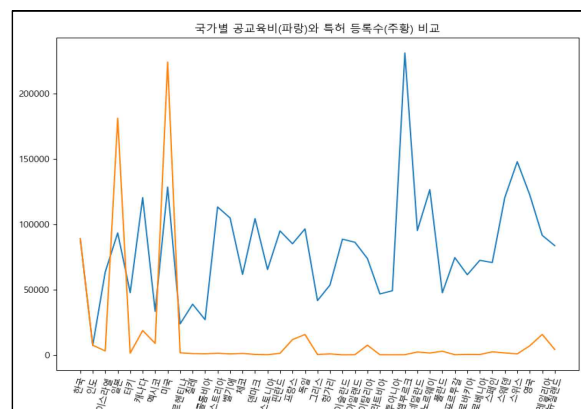


그림 9 국가별 공교육비와 특허 등록수 선 그래프

1) 공교육비와 특허 등록수의 관계 : 주제의 직접적인 해답

Figure 1 consists of two line charts comparing the number of trademark registrations (blue line) and the number of trademark applications (orange line) in the Netherlands and Spain from 2000 to 2018.

Left Chart: 네덜란드의 연도별 광고육비(파랑)와 특허 등록수(주황) 비교

Year	Registrations (Blue)	Applications (Orange)
2000	2800	2800
2001	2900	2900
2002	3000	2700
2003	3100	2500
2004	3200	2400
2005	3400	2400
2006	3600	2300
2007	3800	2300
2008	4000	2000
2009	4100	1900
2010	4100	1900
2011	4200	2000
2012	4400	1900
2013	4600	2000
2014	4500	1600
2015	4600	1400
2016	4700	2000
2017	4800	2300
2018	4900	1900

Right Chart: 스페인의 연도별 광고육비(파랑)와 특허 등록수(주황) 비교

Year	Registrations (Blue)	Applications (Orange)
2000	2200	2200
2001	2300	2300
2002	2400	1300
2003	2500	1900
2004	2600	2000
2005	2800	2700
2006	3000	2200
2007	3200	2600
2008	3400	2300
2009	3600	2600
2010	3800	2700
2011	4000	2800
2012	4100	2700
2013	4200	3200
2014	4100	2500
2015	4000	2400
2016	4100	2100
2017	4300	1800
2018	4400	1200

2) 총 연구원수와 특허 등록수의 관계 : 가장 직접적인 상관관계

국가별 총 연구원수와 특허 출원수 비교

Y-axis: 특허 출원수 (Patent Applications)
X-axis: 총 연구원수 (Total number of researchers)

총 연구원수 (X)	특허 출원수 (Y)
10,000	10,000
15,000	20,000
20,000	15,000
25,000	18,000
30,000	40,000
45,000	35,000
80,000	48,000
85,000	58,000

국가별 총 연구원수(파랑)와 특허 출원수(주황) 비교

국가	연구원수 (파랑)	특허 출원수 (주황)
미국	85,000	60,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	2,000
호주	10,000	2,000
캐나다	10,000	2,000
중화인민공화국	10,000	2,000
대한민국	80,000	48,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	2,000
호주	10,000	2,000
캐나다	10,000	2,000
중화인민공화국	10,000	2,000
대한민국	80,000	48,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	2,000
호주	10,000	2,000
캐나다	10,000	2,000
중화인민공화국	10,000	2,000
대한민국	80,000	48,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	2,000
호주	10,000	2,000
캐나다	10,000	2,000
중화인민공화국	10,000	2,000
대한민국	80,000	48,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	2,000
호주	10,000	2,000
캐나다	10,000	2,000
중화인민공화국	10,000	2,000
대한민국	80,000	48,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	2,000
호주	10,000	2,000
캐나다	10,000	2,000
중화인민공화국	10,000	2,000
대한민국	80,000	48,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	2,000
호주	10,000	2,000
캐나다	10,000	2,000
중화인민공화국	10,000	2,000
대한민국	80,000	48,000
일본	10,000	2,000
독일	10,000	2,000
영국	10,000	2,000
프랑스	10,000	2,000
중국	10,000	2,000
한국	80,000	48,000
이탈리아	10,000	2,000
스페인	10,000	2,000
러시아	10,000	2,000
인도	10,000	2,000
브라질	10,000	

그림 13 국가별 공교육비와 특허 등록수 선 그래프

체이기 때문이라는 추론을 할 수 있다.

3) 공교육비와 학업 성취도의 관계 : 어느정도 비례하는 관계

1)의 결과 공교육비와 특허 등록수는 큰 연관성은 없었지만 **공교육의 가장 기본적인 목적인 학생들의 학업 성취는 어느정도 연관이 있었다**. 공교육비와 학업 성취도의 국가별 평균을 비교한 결과 추세선의 기울기가 0.060로 양수값을 보였고 선 그래프의 추세도 상당부분 일치하는 것을 확인할 수 있으며 두 변수의 상관계수가 0.403으로 어느정도 상관관계가 있으나 0.6 이상의 높은 상관성은 보이지 않는 것을 확인할 수 있다.

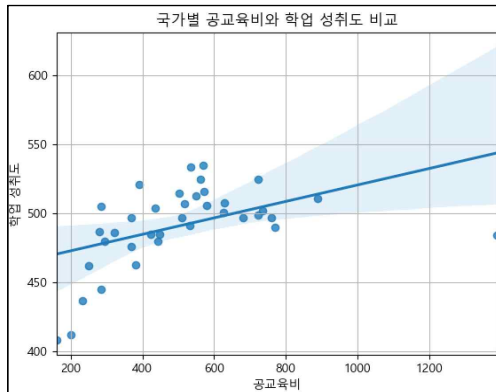


그림 14 국가별 공교육비와 학업 성취도 산점도 그래프

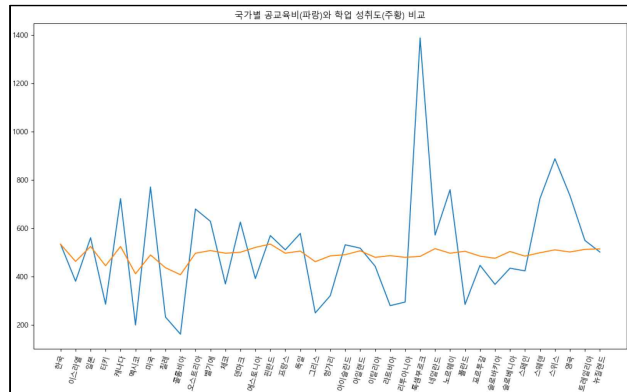


그림 15 국가별 공교육비와 특허 등록수 선 그래프

4) 결론 : 공교육과 더불어 연구원 양성에 큰 노력 필요

결론적으로, “국가 지식재산 확보를 위해 공교육에 투자하는 것은 합리적인 방법인가?”라는 질문의 답은 “아니오”에 가깝다. 데이터 분석 결과 예상과는 다르게 공교육 투자가 지식재산 확보에 긍정적 영향을 주는 것은 사실이지만, 그 영향이 크지 않다는 결론을 얻었다. 또한 국가 지식재산 확보에 정말 중요한 부분은 바로 지식재산 생산의 주체인 연구원 인력의 확보라는 것을 확인하였다.

국가 차원에서 공교육에 투자하는 것은 단순 학업 성취도 향상이나 지식재산 확보 등의 성과보다 훨씬 고차원적인 의미와 목적이 있으나, 적어도 **지식재산 확보의 부분에서는 공교육 투자보다는 연구원 인력의 양성과 지원에 투자하는 것이 훨씬 합리적**이라는 결론이다.

하지만 다음 결론에도 몇 가지 한계가 있다. 먼저 데이터 확보 기간이 10~20년 정도의 기간에 그쳐 공교육비 투자의 장기적인 효과를 분석하는 것에는 한계가 있었다. 공교육비 투자 대상 학생들이 지식 재산을 생산하는 주체가 되는데는 적어도 꽤 오랜 기간이 필요하기 때문에 더 긴 기간의 데이터가 있었다면 조금 더 정확한 분석을 수행할 수 있었을 것이다. 또한 지식재산의 확보를 단순 특허 등록(및 출원)수로 제한한 것도 한계점으로 볼 수 있다. 특허권이 가장 대표적인 지식 재산권이긴 하지만 상표권이나 저작권 등 여러 권리들에 대해서도 분석을 진행했다면 더욱 다채로운 분석과 결론을 얻을 수 있었을 거라는 아쉬움이 있다.