

00000000 원인 분석

4조 : 홍길동 홍순동 이영자 이순자

목차

1. 개요

2. 데이터 수집

3. 데이터 추출 및 정제

4. 데이터 분석

5. 결론

1. 분석배경

저출산 여파에 줄줄이 폐교...문닫은 초중고 3834곳, 대부분 이 '수도권 밖'

 A5면 1단 | 기사입력 2021.01.15. 오후 5:25 | 최종수정 2021.01.15. 오후 8:15 | 기사원문 | 스크랩 |  본문듣기 · 설정

'지방 소멸' 이미 현실이 됐다...전남 828개교·경북 729개 교 '폐교'

"취학대상아동 3~4명 불과...전교생 1명인 곳도"

높은 전출인구까지 '악재'

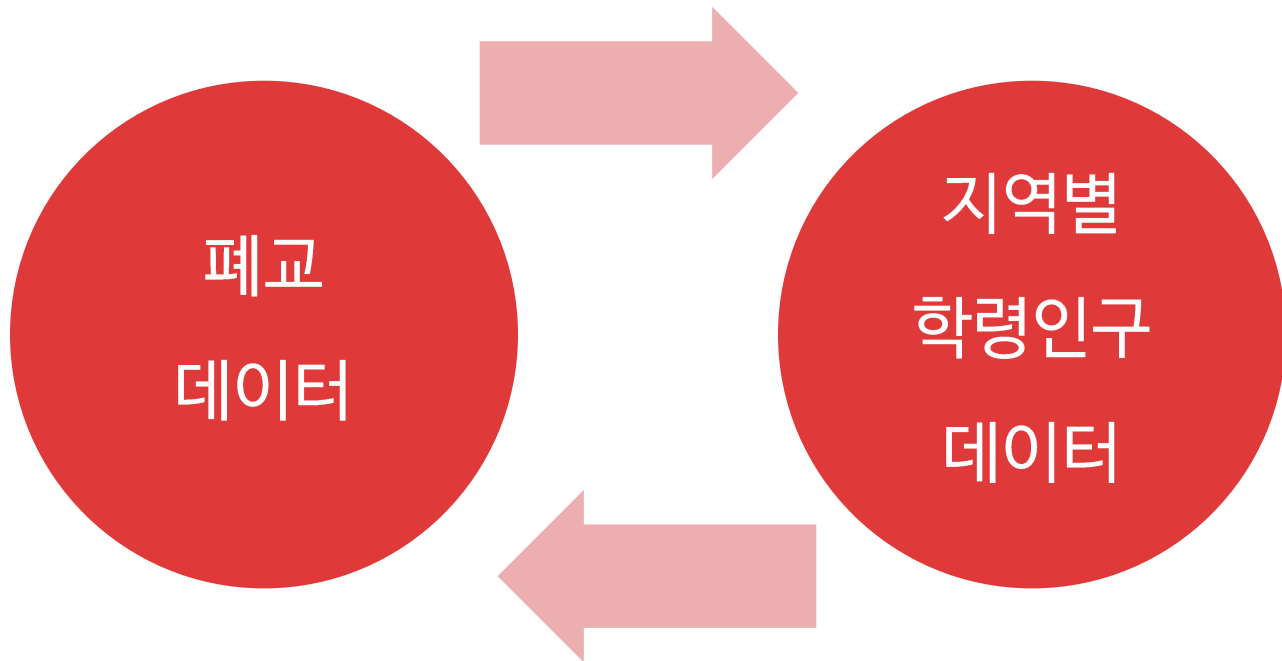
(서울=뉴스1) 최현만 기자 | 2021-01-10 07:28 송고 | 2021-01-10 14:03 최종수정

학생 감소에 40년간 초중고 3855곳 폐교...전체 학교의 32%

기사입력 2021.07.27. 오전 11:47 | 최종수정 2021.07.27. 오후 12:36 | 기사원문 | 스크랩 |  본문듣기 · 설정

- 우리나라는 현재 시간이 지나갈 수록 폐교가 점점 많아진다는 것을 자주 미디어로 접할 수 있음
- 특히나 청소년 인구가 점점 줄어들어 초·중·고등학교 폐교가 점차 늘어날 것으로 예상됨
- 그래서 '학령인구의 감소때문에 폐교가 감소한다'라는 가정을 통해 실제로 초 중 고 인구감소와 폐교 증가가 연관이 있는지 프로젝트를 통해 확인하려는 계획

1. 분석배경



2. 데이터수집_폐교데이터

1. 수집



- ‘지방교육재정알리미’ 있는 데이터 사용
- 지역별 폐교 데이터
(<https://eduinfo.go.kr/portal/main.do#anchor2>)
- 각 지역별로 자세한 폐교 데이터 정보 : 지역별로 초·중·고등학교 폐교 추세 등등 다양한 시계열 데이터를 확인 할 수 있을 것으로 예상

2. 파일 다운로드

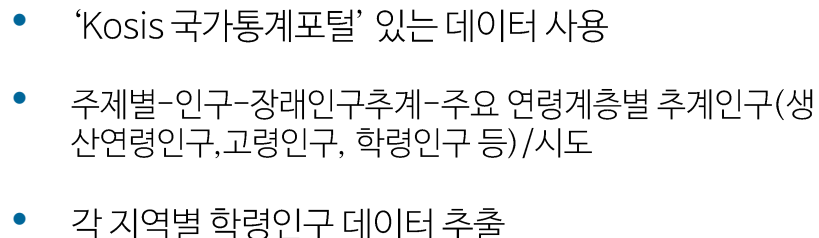
시도교육청 폐교재산현황 (자료일: 2021-03-01)

시·도교육청	폐교학교수	폐교학교	현행학교			폐교학교수(단위: 천명)	폐교학교
			대안(일반)	지역별	개교		
서울	3	1	-	-	-	2	2
부산	45	16	1	22	23	6	29
대구	36	18	5	13	18	-	18
인천	57	40	4	8	12	5	17
광주	14	8	4	2	6	-	6
대전	8	5	2	1	3	-	3
울산	27	10	3	11	14	3	17
세종	2	1	-	-	-	1	1
경기	175	186	59	42	101	16	117
강원	464	256	145	18	163	45	208
충북	254	120	80	21	101	33	134
충남	265	210	13	15	28	27	55
전북	326	272	9	38	47	7	54
전남	833	651	49	44	93	89	182
광주	732	483	123	63	186	63	249
경남	582	338	97	56	153	91	244
제주	32	4	23	-	23	5	28
계	3,805	2,498	617	354	971	393	1,364

이름	유형	업로드 크기
Busan_폐교목록.xlsx	Microsoft Excel 워크시트	10KB
Chungbuk_폐교목록.xlsx	Microsoft Excel 워크시트	18KB
Chungnam_폐교목록.xlsx	Microsoft Excel 워크시트	12KB
Daegu_폐교목록.xlsx	Microsoft Excel 워크시트	10KB
Daejeon_폐교목록.xlsx	Microsoft Excel 워크시트	9KB
Gwangju_폐교목록.xlsx	Microsoft Excel 워크시트	9KB
Gyeongbuk_폐교목록.xlsx	Microsoft Excel 워크시트	25KB
Gyeonggi_폐교목록.xlsx	Microsoft Excel 워크시트	16KB
Gyeongnam_폐교목록.xlsx	Microsoft Excel 워크시트	24KB
Incheon_폐교목록.xlsx	Microsoft Excel 워크시트	10KB
Jeju_폐교목록.xlsx	Microsoft Excel 워크시트	11KB
Jeonbuk_폐교목록.xlsx	Microsoft Excel 워크시트	12KB
Jeonnam_폐교목록.xlsx	Microsoft Excel 워크시트	21KB
Kangwon_폐교목록.xlsx	Microsoft Excel 워크시트	22KB
Sejong_폐교목록.xlsx	Microsoft Excel 워크시트	9KB
Seoul_폐교목록.xlsx	Microsoft Excel 워크시트	9KB
Ulsan_폐교목록.xlsx	Microsoft Excel 워크시트	10KB

- 각 지역마다 폐교목록에 대한 데이터 다운로드
- xlsx 파일 이용

1. 수집



2. 파일 다운로드

- 데이터 사용 기간 : 2001년 ~ 2030년
- xls/xlsx 파일 이용

3. 데이터 추출 및 정제_폐교데이터

1. 파일 한번에 불러 온 후 필요한 컬럼만 추출해 한번에 취합

#각 지역마다 필요한 열만 추출

```
path = './files/'
files = os.listdir(path)

cs = pd.DataFrame()

for file in files:
    data = pd.read_excel(path + file, usecols = 'A,C,E')
    cs = pd.concat([cs,data])

cs_total = cs.reset_index(drop = True)
```

2. 결측값 확인 후, 해당 컬럼에 결측값 확인

#결측값 확인

```
cs_total.isna().sum()
```

#필요한 컬럼 결측치 확인

```
cs_total['급별'].unique()
```

2-1. 결측값에 알맞은 전처리

```
cs_total['급별'] = cs_total['급별'].replace('중','중')
```

2-2. 결측값에 알맞은 전처리

#nan 결측치 대체 (~초등 / ~중등 / ~고등)

```
for i in range(len(cs_total)):
    if cs_total['급별'].isna()[i] == True:
        if "초등" in cs_total['폐교명'][i]:
            cs_total['급별'].values[i] = '초'
        elif "중등" in cs_total['폐교명'][i] == True:
            cs_total['급별'].values[i] = '중'
        elif "고등" in cs_total['폐교명'][i] == True:
            cs_total['급별'].values[i] = '고'
```

#nan 결측치 대체 (~초 / ~중 / ~고)

```
for i in range(len(cs_total)):
    if cs_total['급별'].isna()[i] == True:
        str = cs_total["폐교명"][i]
        print(str)
        if str[-1] == "초":
            cs_total['급별'].values[i] = '초'
        elif str[-1] == "중":
            cs_total['급별'].values[i] = '중'
        elif str[-1] == "고":
            cs_total['급별'].values[i] = '고'
```

3. 필요하지 않는 데이터 제거 및 타입 변경

#필요 데이터만 추출

```
cs_close = (cs_total["폐교년도"] >= 2001) & (cs_total["폐교년도"] <= 2020)
cs_totals = cs_total[cs_close]
```

#폐교년도 타입 변경

```
cs_totals["폐교년도"] = cs_totals["폐교년도"].astype("str")
```

#인덱스 리셋

```
cs_totals = cs_totals.reset_index(drop = True)
```



	시도교육청	폐교년도	급별
0	부산	2009	중
1	부산	2007	초
2	부산	2010	고
3	부산	2016	초
4	부산	2020	중

3. 데이터 추출 및 정제_지역별 학령인구

1. 파일 불러온 후 필요 없는 컬럼 제거

```
region_youth_1 = pd.read_excel('./region_youth_1.xlsx', usecols = "B:BY", header = 1, index_col = 0)
```

2. 가져온 데이터 확인 후 전체 컬럼 이름 확인

```
for col in region_youth_1:  
    print(col)
```

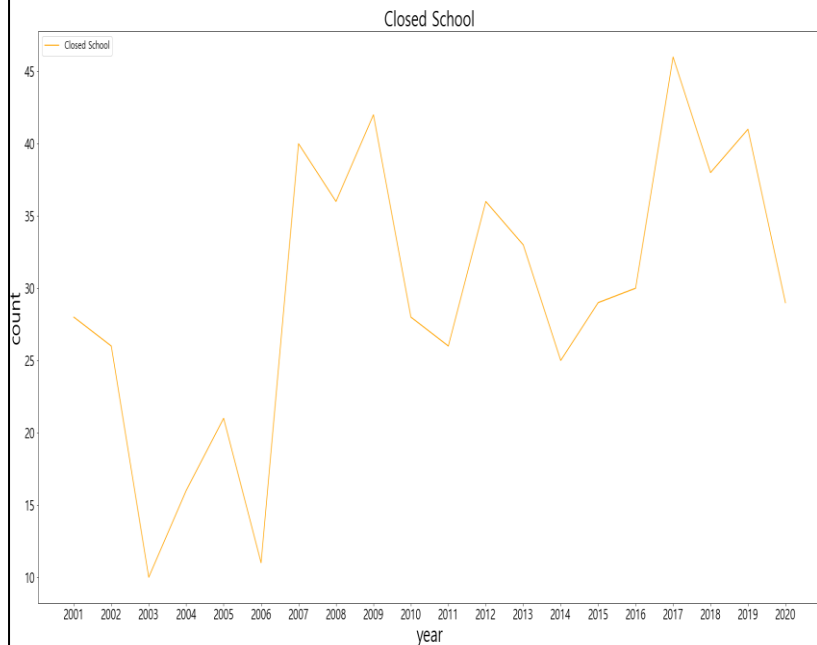
3. 필요한 컬럼만 따로 추출 및 새로운 파생변수 추가

```
#전국  
region_all = region_youth_1[['전국', '전국.1', '전국.2']]  
region_all.columns = ['ele', 'mid', 'hig']  
region_all['all'] = region_all['ele'] + region_all['mid'] + region_all['hig']  
region_all.reset_index(level=0, inplace=True)
```



	시점	ele	mid	hig	all
0	2001	4150	1847	2014	8011
1	2002	4192	1863	1902	7957
2	2003	4186	1913	1846	7945
3	2004	4118	1993	1824	7935
4	2005	4018	2064	1841	7923

4. 데이터 분석_전체 폐교데이터



- 대한민국 전체에서 폐교 수는 2007년부터 꾸준하게 30 ~ 45개씩 폐교 수가 존재한 것을 확인

#시계열그래프 시각화

#그래프 크기

```
plt.figure(figsize = (30, 15))
```

#그래프 x 및 y행에 대한 데이터 지정

```
plt.plot(cs_all['폐교년도'], cs_all['폐교 수'], color = 'orange', label='Closed School')
```

#x 눈금레이블 크기

```
plt.xticks(fontsize = 20)
```

#y 눈금레이블 크기

```
plt.yticks(fontsize = 20)
```

#x레이블 제목 크기

```
plt.xlabel('year', fontsize = 30)
```

#y레이블 제목 크기

```
plt.ylabel('count', fontsize = 30)
```

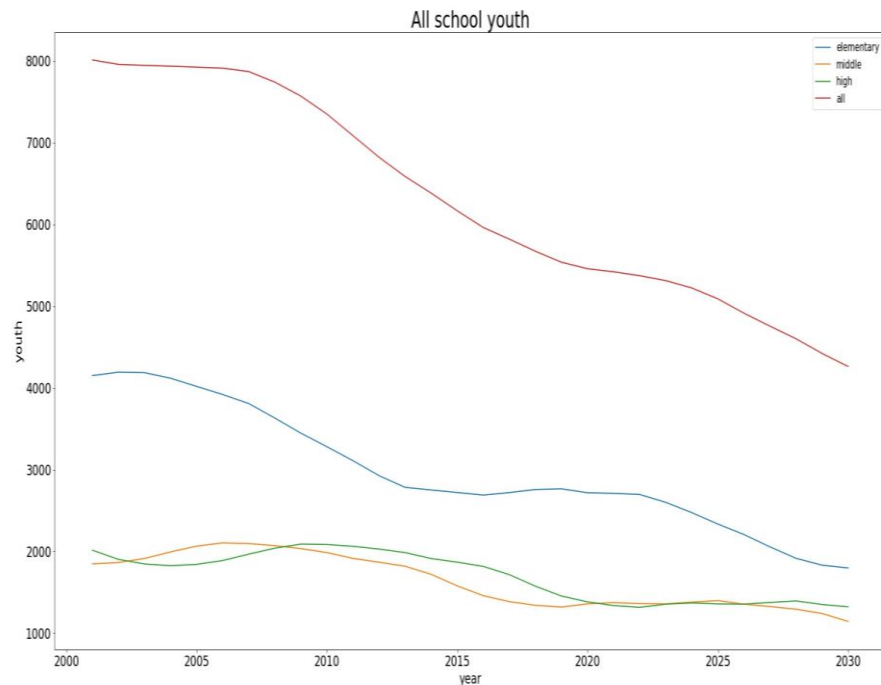
#그림 제목 크기

```
plt.title('Closed School', fontsize = 30)
```

#범례위치 및 크기 지정

```
plt.legend(loc = 'upper left', fontsize = 15)
```

4. 데이터 분석_전체 학령인구 데이터



- 전체적인 학령인구는 2010년대부터 점차적으로 줄어드는 것을 확인

#전국 학령인구 시계열 그래프

#그래프 크기

```
plt.figure(figsize=(30, 15))
```

#그래프 x축 y축 내용 추가

```
plt.plot(region_all['시점'], region_all['ele'], label='elementary')
```

```
plt.plot(region_all['시점'], region_all['mid'], label='middle')
```

```
plt.plot(region_all['시점'], region_all['high'], label='high')
```

```
plt.plot(region_all['시점'], region_all['all'], label='all')
```

#그래프 제목

```
plt.title('All school youth', fontsize = 30)
```

#x축 제목 크기

```
plt.xticks(fontsize = 20)
```

#y축 제목 크기

```
plt.yticks(fontsize = 20)
```

#x축

```
plt.xlabel('year', fontsize = 20)
```

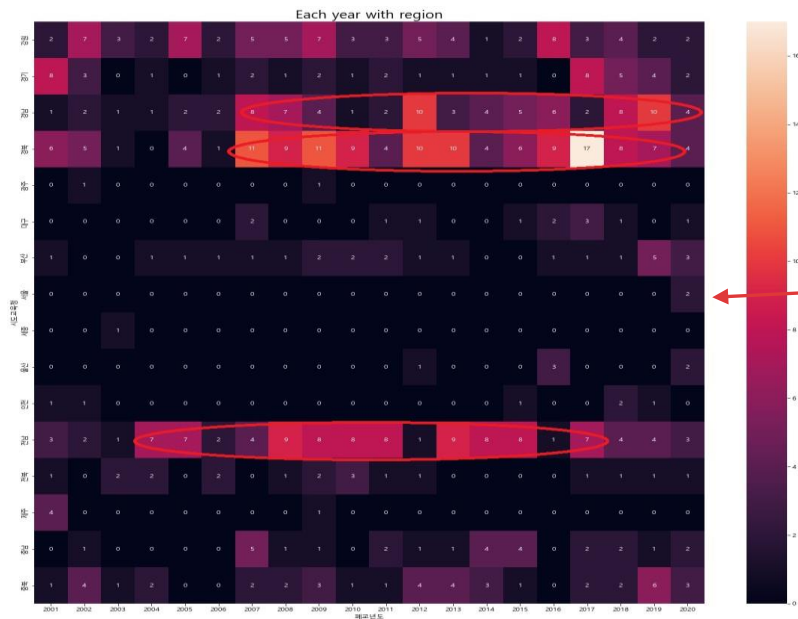
#y축

```
plt.ylabel('youth', fontsize = 20)
```

#범례

```
plt.legend(loc = 'upper right', fontsize = 15)
```

4. 데이터분석_전체 폐교데이터



- 2001 ~ 2020년 사이 각 지역별 폐교데이터를 확인
- 경북, 경남, 전남 지역에서 2000년대 중반부터 폐교가 많이 발생한 것을 확인
- 서울지역에서 2020년 2건 확인

#히트맵 그래프 시각화

```
plt.figure(figsize = (20, 20))
```

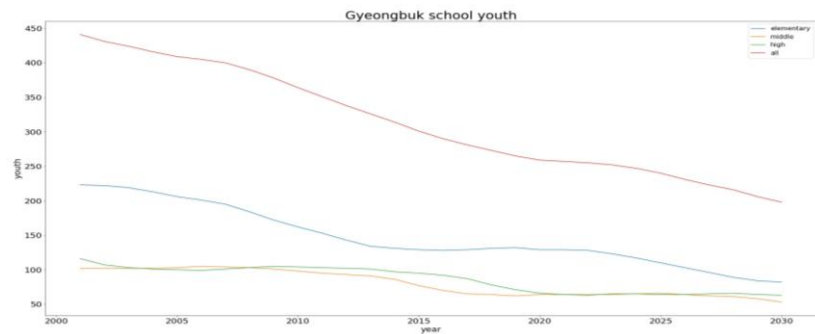
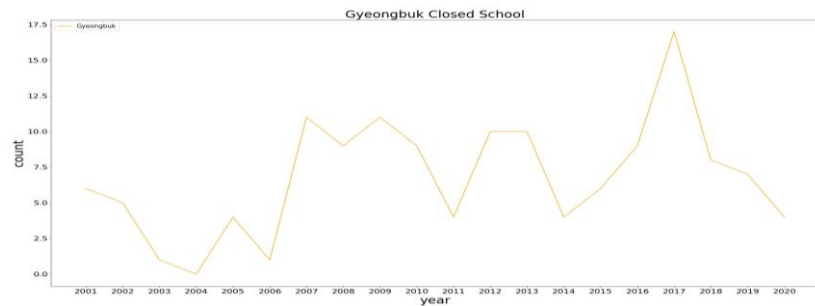
```
sns.heatmap(df, annot= True)
```

```
plt.title('Each year with region', fontsize = 20)
```

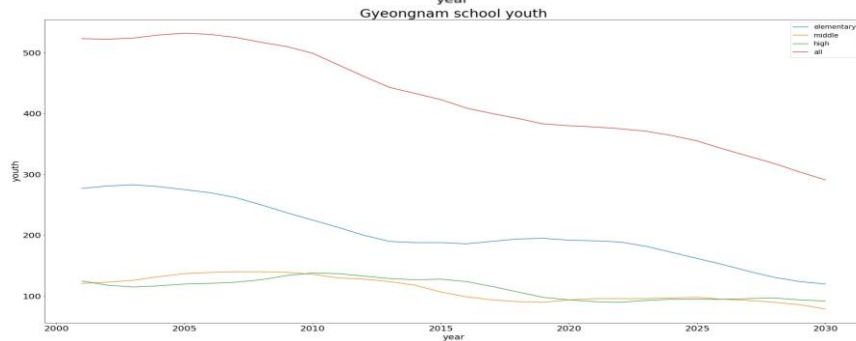
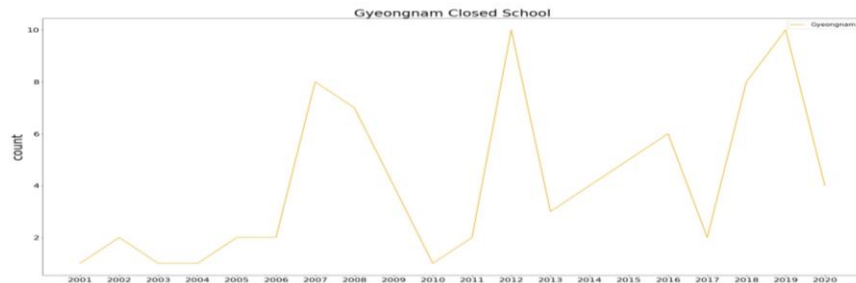
- 세종은 2003년 당시 현재 세종에 포함된 지역에서 있었던 폐교 데이터 1건 존재

4. 데이터 분석_특정지역 별 학령인구 데이터

1) 경북

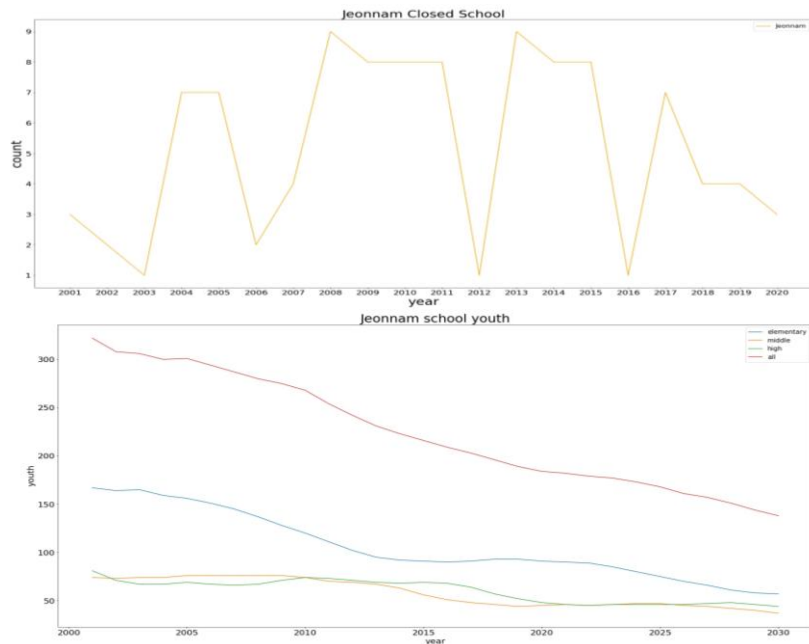


2) 경남



3. 데이터 추출 및 정제_특정지역 별 폐교데이터

3) 전남



- 수도권에 거리가 먼 지역들이고 이에 2030 세대들은 **수도권으로 인구유입, 학령인구 감소** 등으로 폐교가 더 늘어날 것으로 예상
- 폐교 증가한 시기 및 인구 감소시기에 관련된 뉴스 기사 확인 (2007년 ~ 2013년 사이 기사)

전남 학령인구 10년 뒤 '반토막'

기사입력 2009.10.13. 오전 10:38 [스크랩](#) [본문듣기](#) · [설정](#)

경북교육청 초·중학 학급당 정원 줄여

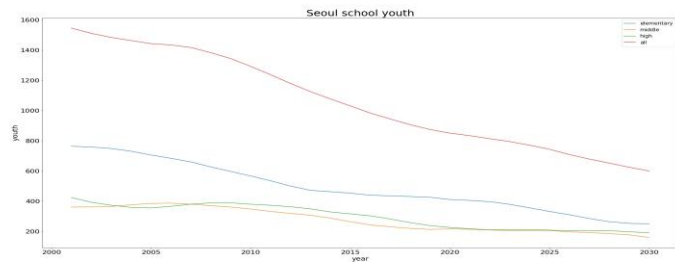
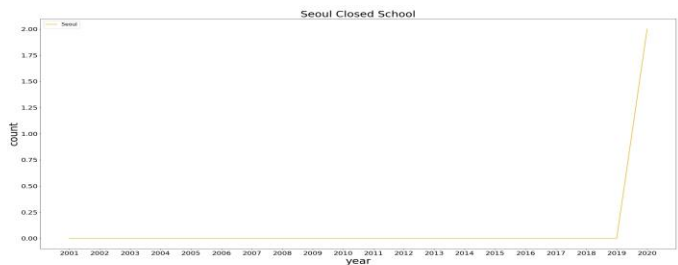
기사입력 2008.01.25. 오후 4:12 [스크랩](#) [본문듣기](#) · [설정](#)

경남 초·중·고 5곳 중 1곳 학생 수 부족 통폐합 대상

기사입력 2011.03.11. 오전 10:33 [기사원문](#) [스크랩](#) [본문듣기](#) · [설정](#)

3. 데이터 추출 및 정제_특정지역 별 폐교데이터

4) 서울



- 2020년 서울에 폐교 2건 발생
- 2010년부터 급격한 학령인구 감소로 폐교 발생 (2010년 129.2만명 → 2020년 **85만명**)

→ 더 이상 서울도 폐교에 대한 걱정을 안 할 수가 없음

[사설]서울마저 '저출산 폐교'... 예고된 재앙 눈뜨고 당할 건가

📖 A35면 1단 | 기사입력 2019.03.26. 오전 3:04 | 기사원문 | 스크랩 | 본문듣기 · 설정

서울 학령인구 2000년 153만 명에서 2045년 58만 명까지 감소 전망

기사입력 2021.07.29. 오후 3:43 | 기사원문 | 스크랩 | 본문듣기 · 설정

4. 데이터분석_상관분석 : 폐교수와 학령인구 상관관계

1. 통합 파일 생성

(연도기준 폐교 및 학령인구 통합)

#일부 데이터 결합

```
corr_20012020 = pd.concat([region_all, cs_all], axis =1)
```



	Elementary	Middle	High	All	Closed School
0	4150	1847	2014	8011	28
1	4192	1863	1902	7957	26
2	4186	1913	1846	7945	10
3	4118	1993	1824	7935	16
4	4018	2064	1841	7923	21

2. 변수간 공분산 해석

(pandas cov()사용)

#공분산 분석 (cov)

```
corr_20012020.cov()
```

	Elementary	Middle	High	All	Closed School
Elementary	365868.450000	123361.344737	43807.455263	533037.250000	-3270.823684
Middle	123361.344737	77132.028847	43421.350000	243914.723684	-1112.086842
High	43807.455263	43421.350000	39953.313158	127182.18421	-203.576316
All	533037.250000	243914.723684	127182.18421	904134.92105	-4586.486842
Closed School	-3270.823684	-1112.086842	-203.576316	-4586.486842	98.260526

- 초등학생 수와 폐교 수가 변수간 관계가 큼
- 초등학생 수가 감소할 수록 폐교 수는 증가함

3. 변수간 상관관계 분석

(pandas corr()사용)

#상관분석 실행 (corr)

```
cor_reg_clo = corr_20012020.corr(method='pearson')
```

```
cor_reg_clo.corr()
```

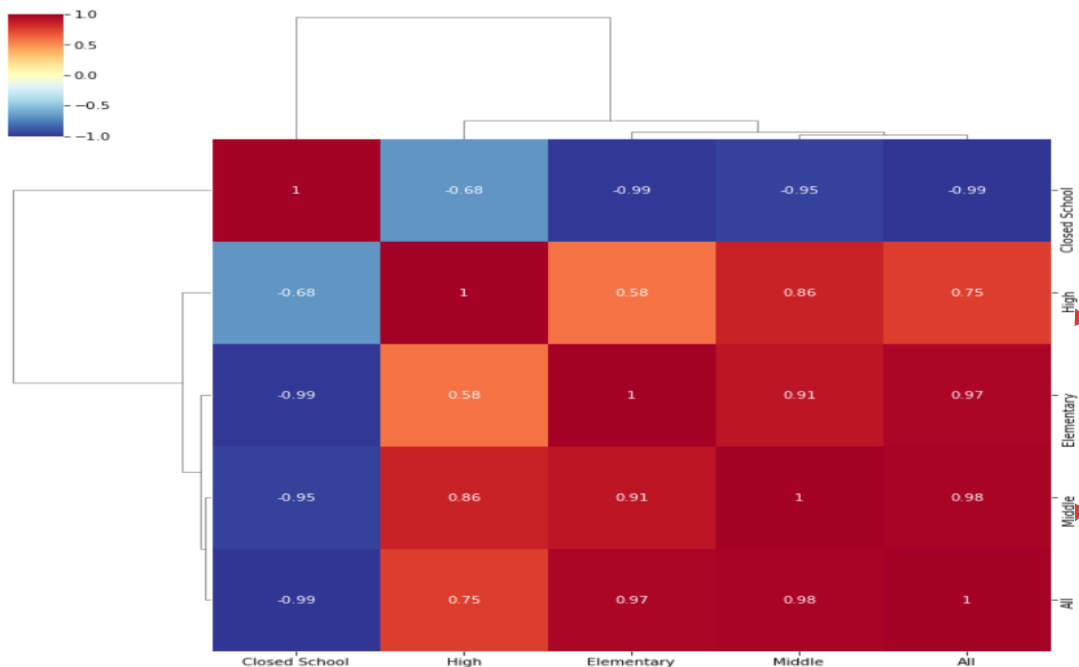
	Elementary	Middle	High	All	Closed School
Elementary	1.000000	0.909908	0.584110	0.973927	-0.990466
Middle	0.909908	1.000000	0.857488	0.979504	-0.954843
High	0.584110	0.857488	1.000000	0.751618	-0.684978
All	0.973927	0.979504	0.751618	1.000000	-0.994418
Closed School	-0.990466	-0.954843	-0.684978	-0.994418	1.000000

- 폐교수와 전체 학령인구와는 강한 음의상관관계를 확인

4. 데이터분석_상관분석 : 폐교수와 학령인구 상관관계

#상관분석 시각화(clustermap)

```
sns.clustermap(corr, annot = True, cmap = 'RdYlBu_r', vmin = -1, vmax = 1)
```



- 전 연령별로 폐교 수 데이터와 **강한 음의 상관관계**를 파악
 - 초등학교, 중학교, 고등학교 순으로 폐교 데이터와 강한 상관관계를 갖고 있음
 - 전체 학령인구 감소와 비슷한 영향 있는 연령은 중학교 학령인구를 확인 (**0.98**)
- > 고등학교 학령인구와는 비교적 낮은 상관관계 (**0.75**)

4. 데이터분석_회귀분석모델 도출 : 전처리

1. 데이터 그룹별 집계 (groupby, count 사용)

```
cs_emh = cs_totals.groupby(['폐교년도', '급별'], as_index=False).count()
```



```
cs_emh.head()
```

	폐교년도	급별	시도교육청
0	2001	고	2
1	2001	중	2
2	2001	초	24
3	2002	중	3
4	2002	초	23

2. 각 그룹별로 전처리

```
ele = (cs_emh['급별'] == '초')  
cs_ele = cs_emh[ele]
```

3. 2001년 ~ 2020년 DataFrame 생성

```
time = ['2001', '2002', '2003', '2004', '2005',  
        '2006', '2007', '2008', '2009', '2010',  
        '2011', '2012', '2013', '2014', '2015',  
        '2016', '2017', '2018', '2019', '2020']
```

```
time_dic = {'폐교년도': time}
```

```
#DataFrame형식 변환  
time_df = pd.DataFrame(time_dic)
```

4. Left join 후 불필요한 컬럼 제거

```
#left join  
cs_ele_1 = pd.merge(time_df, cs_ele, how='left')  
cs_ele_1 = cs_ele_1.fillna(0)  
  
cs_ele_2 = cs_ele_1.drop(['폐교년도', '급별'], axis=1)
```

5. 각 그룹별 항목들 병합 및 파생변수 추가

```
cs_hig_2 = cs_hig_1.drop(['폐교년도', '급별'], axis=1)
```

```
cs_all_emh = pd.concat([time_df, cs_ele_2, cs_mid_2, cs_hig_2], axis=1)
```

```
cs_all_emh.columns = ['year', 'elementary', 'middle', 'high']
```

```
cs_all_emh['all'] = cs_all_emh['elementary'] + cs_all_emh['middle'] + cs_all_emh['high']
```

6. 폐교데이터, 학령인구 데이터 통합

```
cs_regre = pd.concat([cs_all_emh, corr_20012020], axis=1)
```

```
cs_regre.rename(columns = {'year':'년도', 'elementary':'초등학교', 'middle':'중학교', 'high':'고등학교', 'all':'폐교',  
                           'Elementary':'초등학생', 'Middle':'중학생', 'High':'고등학생', 'All':'학령인구'}, inplace=True)
```

	년도	초등학교	중학교	고등학교	폐교	초등학생	중학생	고등학생	학령인구
0	2001	24	2	2	28	4150	1847	2014	8011
1	2002	23	3	0	26	4192	1863	1902	7957
2	2003	9	0	1	10	4186	1913	1846	7945
3	2004	13	2	1	16	4118	1993	1824	7935
4	2005	16	4	1	21	4018	2064	1841	7923

4. 데이터분석_회귀분석모델 도출 : 학령인구수와 폐교수 회귀분석

2-1. 초등학교~초등학생수

(파이썬 statsmodels의 OLS Mode사용, 최소제곱법 사용 단순회귀분석 실시)

```
#초등학생 / 초등학교
```

```
model = smf.ols(formula = '초등학교 ~ 초등학생', data =cs_regre)
result_ole = model.fit()
result_ole.summary()
```

OLS Regression Results

Dep. Variable:	초등학교	R-squared:	0.026
Model:	OLS	Adj. R-squared:	-0.028
Method:	Least Squares	F-statistic:	0.4874
Date:	Sat, 11 Dec 2021	Prob (F-statistic):	0.494
Time:	20:06:17	Log-Likelihood:	-68.262
No. Observations:	20	AIC:	140.5
Df Residuals:	18	BIC:	142.5
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	28.6864	9.944	2.885	0.010	7.794	49.578
초등학생	-0.0021	0.003	-0.698	0.494	-0.008	0.004

Omnibus:	1.126	Durbin-Watson:	1.640
Prob(Omnibus):	0.569	Jarque-Bera (JB):	0.815
Skew:	0.472	Prob(JB):	0.665
Kurtosis:	2.704	Cond. No.	1.94e+04

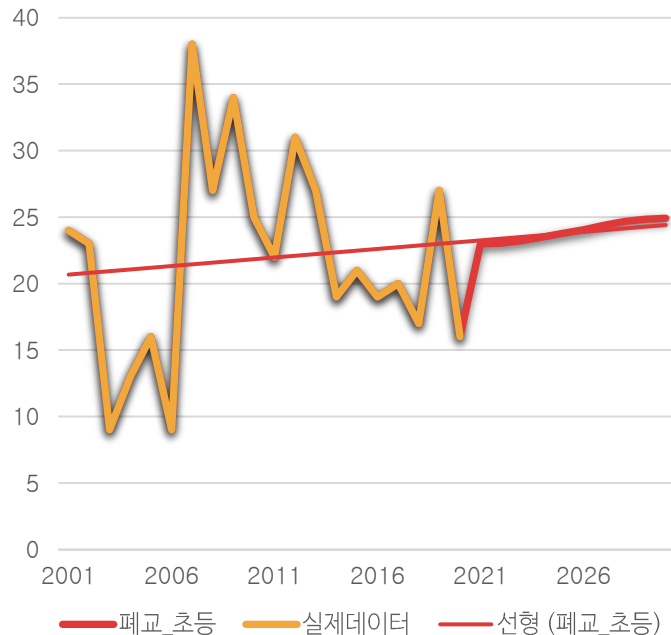
- R-squared: 0.026 (회귀모형 설명력 2.6%)



회귀모형의식

$$Y = 28.6864 + (-0.0021) \times \text{초등학생 수}$$

초등학교 폐교 데이터



4. 데이터분석_회귀분석모델 도출 : 학령인구수와 폐교수 회귀분석

2-2. 중학교~중학생수 (초등학생수와 동일하게 단순회귀분석 실시)

#중학생 / 중학교

```
model = smf.ols(formula = '중학교 ~ 중학생', data = cs_regre)
result = model.fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	중학교	R-squared:	0.614
Model:	OLS	Adj. R-squared:	0.592
Method:	Least Squares	F-statistic:	28.62
Date:	Sat, 11 Dec 2021	Prob (F-statistic):	4.38e-05
Time:	20:35:06	Log-Likelihood:	-53.720
No. Observations:	20	AIC:	111.4
Df Residuals:	18	BIC:	113.4
Df Model:	1		
Covariance Type:	nonrobust		

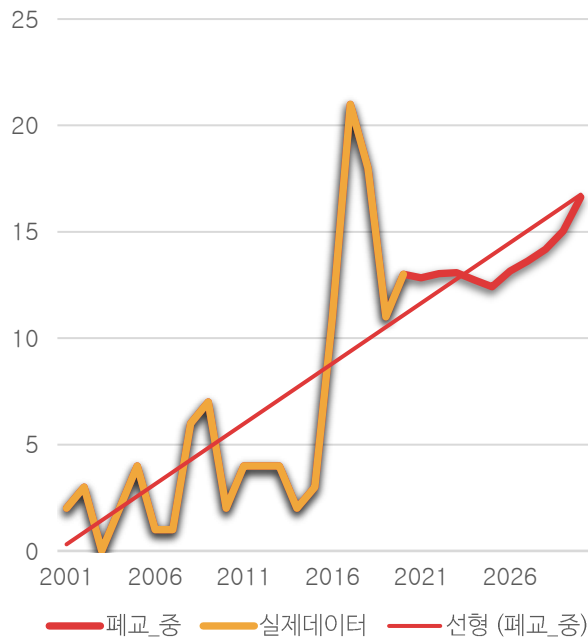
	coef	std err	t	P> t	[0.025	0.975]
Intercept	35.4897	5.585	6.355	0.000	23.756	47.223
중학생	-0.0165	0.003	-5.350	0.000	-0.023	-0.010

Omnibus:	1.586	Durbin-Watson:	1.165
Prob(Omnibus):	0.453	Jarque-Bera (JB):	0.875
Skew:	0.512	Prob(JB):	0.646
Kurtosis:	2.969	Cond. No.	1.21e+04

- R-squared: 0.614 (회귀모형 설명력 61.4%)

→ 회귀모형의 식
$$Y = 35.4897 + (-0.0165) \times \text{중학생수}$$

중학교 폐교 데이터



4. 데이터분석_회귀분석모델 도출 : 학령인구수와 폐교수 회귀분석

3. 폐교 수~학령인구 수 (초, 중, 고) - 다중회귀분석

(파이썬 statsmodels의 OLS Mode사용, 다중회귀분석 실시)

#전체연령 / 다중회귀

```
model = smf.ols(formula = '폐교 ~ 초등학생 + 중학생 + 고등학생', data = cs_regre)
result = model.fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	폐교	R-squared:	0.331
Model:	OLS	Adj. R-squared:	0.206
Method:	Least Squares	F-statistic:	2.640
Date:	Sat, 11 Dec 2021	Prob (F-statistic):	0.0849
Time:	19:06:30	Log-Likelihood:	-69.721
No. Observations:	20	AIC:	147.4
Df Residuals:	16	BIC:	151.4
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	43.8414	22.055	1.988	0.064	-2.913	90.595
초등학생	-0.0063	0.006	-1.103	0.287	-0.018	0.006
중학생	-0.0138	0.019	-0.744	0.467	-0.053	0.026
고등학생	0.0168	0.019	0.895	0.384	-0.023	0.057

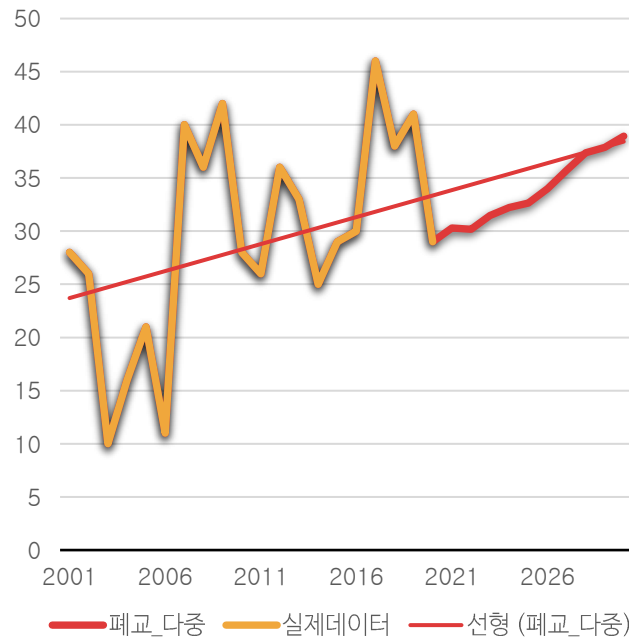
Omnibus:	1.147	Durbin-Watson:	1.650
Prob(Omnibus):	0.563	Jarque-Bera (JB):	0.972
Skew:	0.326	Prob(JB):	0.615
Kurtosis:	2.140	Cond. No.	4.76e+04

• Adj. R-squared: 0.206 (회귀모형 설명력 20.6%)

회귀모형의 식

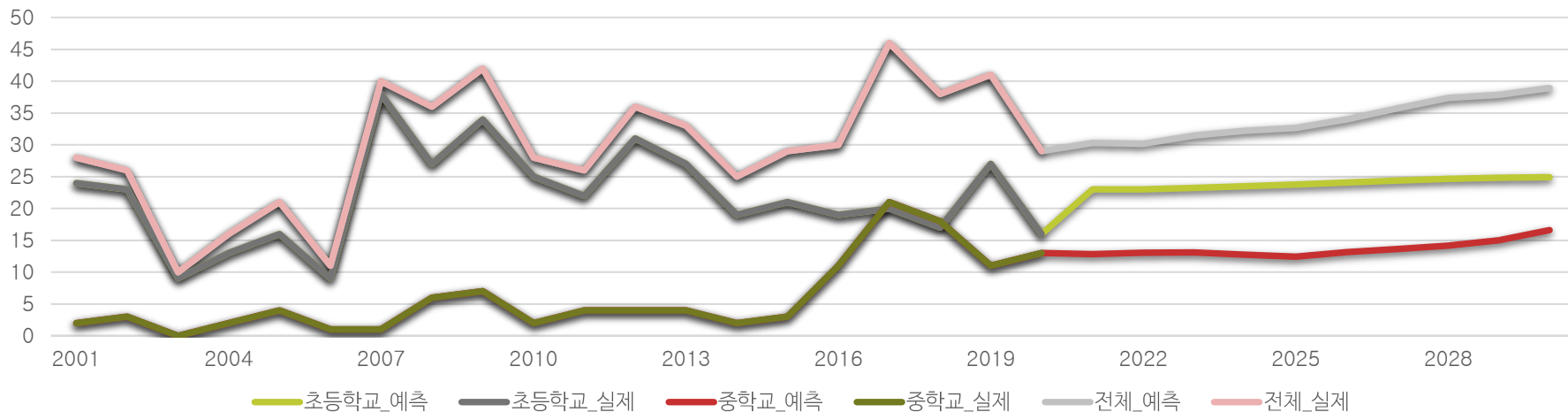
➔
$$Y = 43.8414 + (-0.0063) \times \text{초등학생 수} + (-0.0138) \times \text{중학생 수} + (0.0168) \times \text{고등학생 수}$$

학령인구 폐교데이터



4. 데이터분석_회귀모델 기반 예측 시각화

학령인구 폐교데이터 (전체)



- 전체 폐교 수 중 초등학교 폐교수가 대부분 을 차지
- 회귀식을 통해 예측 결과 점차 늘어 날 것 예상

5. 결론

- 폐교 수 증가는 학령 인구감소와 밀접한 관련이 있어 보임
- 특히, 초등학교 중학교 학령인구 감소와 폐교수의 데이터는 매우 높은 상관관계
- 학령인구 회귀모형 설명력은 20.6% 기록
→ 독립변수 데이터가 회귀분석에 적합한 데이터 적용 X / 데이터 양이 적음
- 단순회귀 및 다중회귀를 통해 폐교수를 예측한 결과 앞으로 늘어날 것으로 예상
→ 정확한 예측을 하지 못한 점 한계 (아무런 변수 없이 회귀식만을 통해 예측)

5. 결론_출산율

- 현재 한국 출산율은 1인당 0.84명으로 학령인구 감소에 큰 영향을 끼칠 것으로 예상
- 출산율이 상승함으로써 학령인구 증가를 통해 폐교를 줄일 수 있음

5-1. 대책

1. 국가 및 각 지역 지자체에서 출산장려정책을 통해 출산 장려
2. 지역세분화 정책을 통해 세종이나 각 지역별 신도시 건설 및 꾸준한 재정비를 통해 경제상승으로 출산율 상승 기대



별첨_각 지역별 폐교데이터

1. 2001 ~ 2020년에 대한 DataFrame 형성

#시점데이터 사전형식으로 정리

```
time = ['2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010',  
        '2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020']
```

```
time_dic = {'폐교년도': time}
```

#DataFrame형식 변환

```
time_df = pd.DataFrame(time_dic)
```

폐교년도

0	2001
1	2002
2	2003
3	2004
4	2005



2. 각 지역에 대한 데이터 추출 및 전처리

#부산

```
busan = (cs_totals["시도교육청"] == '부산')  
busan_if = cs_totals[busan]  
cs_bu = busan_if.groupby(['폐교년도'], as_index=False).count()  
cs_bu = cs_bu.drop(['급별'], axis='columns')  
cs_bu = cs_bu.rename(columns={'시도교육청': '폐교 수'})  
cs_bu['폐교년도'] = cs_bu['폐교년도'].astype('str')
```



#left join

```
cs_busan = pd.merge(time_df, cs_bu, how='left')
```

#결측치들을 '0'으로 변환

```
cs_busan = cs_busan.fillna(0)
```

3. 각 지역에 대해 시각화

#시계열그래프 시각화_부산

#그래프 크기

```
plt.figure(figsize = (30, 15))
```

#그래프 x 및 y행에 대한 데이터 지정

```
plt.plot(cs_busan['폐교년도'], cs_busan['폐교 수'], color = 'orange', label='Busan')
```

#x 눈금레이블 크기

```
plt.xticks(fontsize = 20)
```

#y 눈금레이블 크기

```
plt.yticks(fontsize = 20)
```

#x레이블 제목 크기

```
plt.xlabel('year', fontsize = 30)
```

#y레이블 제목 크기

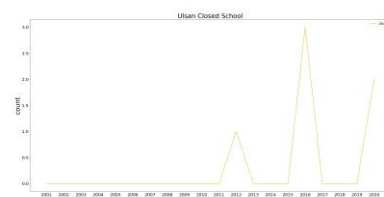
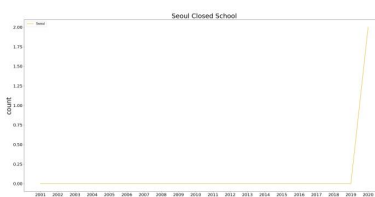
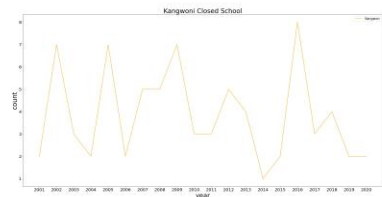
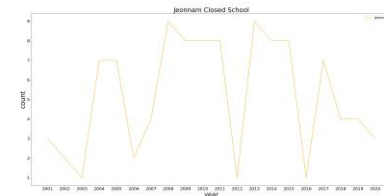
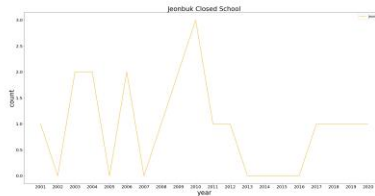
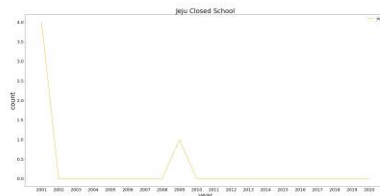
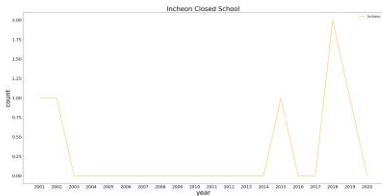
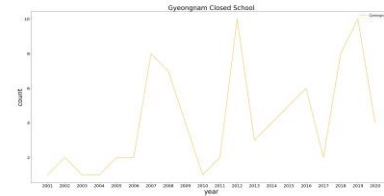
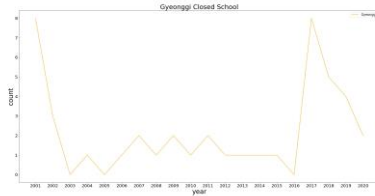
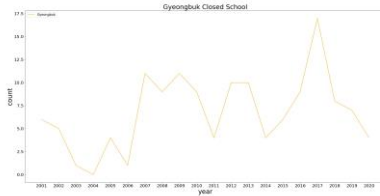
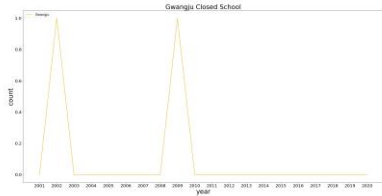
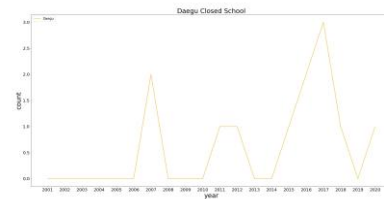
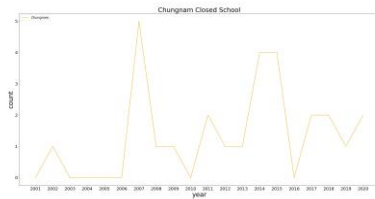
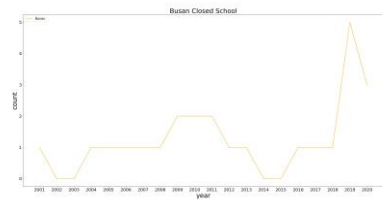
```
plt.ylabel('count', fontsize = 30)
```

#그림 제목 크기

```
plt.title('Busan Closed School', fontsize = 30)
```

#범례위치 및 크기 지정

```
plt.legend(loc = 'upper left', fontsize = 15)
```

- 세종은 세종에 편입된 지역 중 2003년에 폐교된 학교 포함
- 대전은 2001년 ~ 2020년도까지 폐교 데이터가 없음

별첨_각 지역별 학령인구

1. 각 지역별로 필요한 컬럼 추출 및 파생변수 추가

#서울

```
region_seo = region_youth_1[['서울특별시', '서울특별시.1', '서울특별시.2']]
region_seo.columns = ['ele', 'mid', 'hig']
region_seo['all'] = region_seo['ele'] + region_seo['mid'] + region_seo['hig']
region_seo.reset_index(level= ['시점'], inplace= True)
```

#부산

```
region_bus = region_youth_1[['부산광역시', '부산광역시.1', '부산광역시.2']]
region_bus.columns = ['ele', 'mid', 'hig']
region_bus['all'] = region_bus['ele'] + region_bus['mid'] + region_bus['hig']
region_bus.reset_index(level= ['시점'], inplace= True)
```

#대구

```
region_daeg = region_youth_1[['대구광역시', '대구광역시.1', '대구광역시.2']]
region_daeg.columns = ['ele', 'mid', 'hig']
region_daeg['all'] = region_daeg['ele'] + region_daeg['mid'] + region_daeg['hig']
region_daeg.reset_index(level= ['시점'], inplace= True)
```

#인천

```
region_inc = region_youth_1[['인천광역시', '인천광역시.1', '인천광역시.2']]
region_inc.columns = ['ele', 'mid', 'hig']
region_inc['all'] = region_inc['ele'] + region_inc['mid'] + region_inc['hig']
region_inc.reset_index(level= ['시점'], inplace= True)
```

#광주

```
region_gwa = region_youth_1[['광주광역시', '광주광역시.1', '광주광역시.2']]
region_gwa.columns = ['ele', 'mid', 'hig']
region_gwa['all'] = region_gwa['ele'] + region_gwa['mid'] + region_gwa['hig']
region_gwa.reset_index(level= ['시점'], inplace= True)
```

2. 각 지역에 대해 시각화

#그래프 크기

```
plt.figure(figsize=(30, 15))
```

#그래프 x축 y축 내용 추가

```
plt.plot(region_seo['시점'], region_seo['ele'], label = 'elementary')
plt.plot(region_seo['시점'], region_seo['mid'], label = 'middle')
plt.plot(region_seo['시점'], region_seo['hig'], label = 'high')
plt.plot(region_seo['시점'], region_seo['all'], label = 'all')
```

#그래프 제목

```
plt.title('Seoul school youth', fontsize = 30)
```

#x축 제목 크기

```
plt.xticks(fontsize = 20)
```

#y축 제목 크기

```
plt.yticks(fontsize = 20)
```

#x축

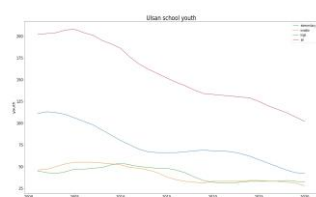
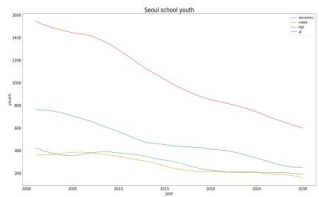
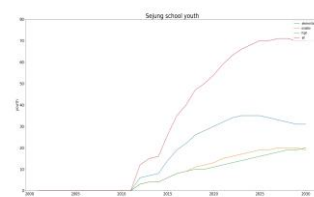
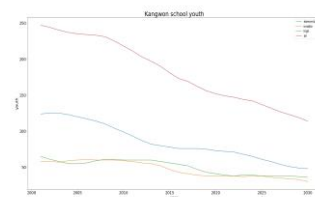
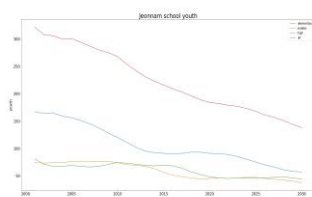
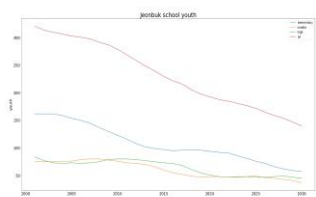
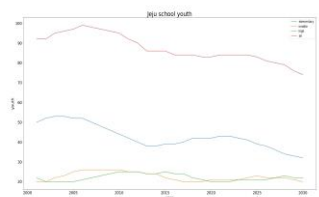
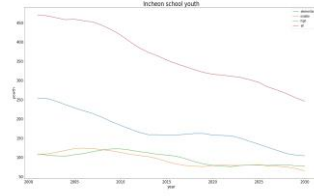
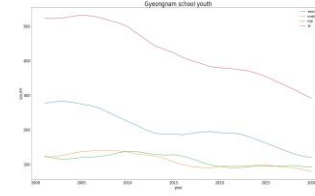
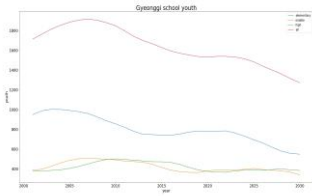
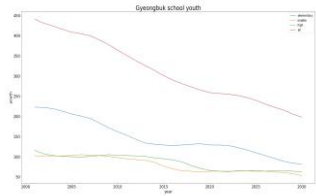
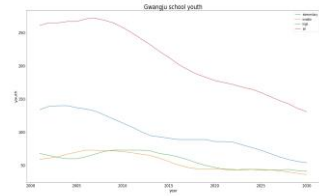
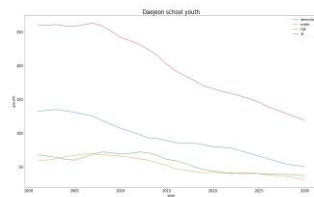
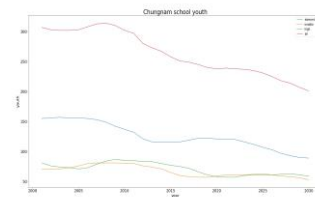
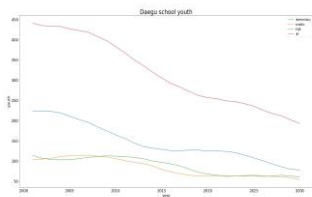
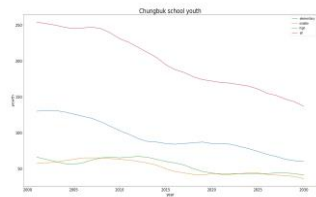
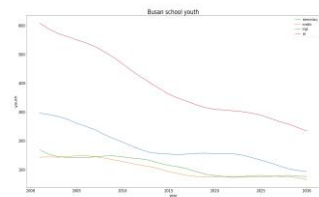
```
plt.xlabel('year', fontsize = 20)
```

#y축

```
plt.ylabel('youth', fontsize = 20)
```

#범례

```
plt.legend(loc = 'upper right', fontsize = 15)
```



- OO은 2012년부터 데이터 있어서 다른 그래프들과 상이