인구통계(출생) 데이터 및 시설물 데이터 EDA

학습 목표

- 데이터에 대한 탐색을 수행한다.

- [경제-건축분야. 인구통계]
- [데이터 출처]

- 총 출생아수 : https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=INH_1B8000F_01&vw_cd=MT_ZTITLE&list_id=A21&seqNo=&lang_mode=ko&language=k (https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=INH_1B8000F_01&vw_cd=MT_ZTITLE&list_id=A21&seqNo=&lang_mode=ko&language=k
- 고등교육 이수율 : http://www.index.go.kr/unify/idx-info.do?idxCd=8024 (http://www.index.go.kr/unify/idx-info.do?idxCd=8024)
- 2022_도로_교량 및 터널현황조서 https://bti.kict.re.kr/bti/publicMain/main.do (https://bti.kict.re.kr/bti/publicMain/main.do)
- 학교별, 학과별 고등교육기관 취업통계 https://kess.kedi.re.kr/contents/dataset?itemCode=04&menuId=m_02_04_03_02&tabId=m3 (https://kess.kedi.re.kr/contents/dataset?itemCode=04&menuId=m_02_04_03_02&tabId=m3)

- 데이터 분석 코드
  - HTML코드 (https://ldjwj.github.io/dataAnalysis/01_12_population_analysis.html)

학습 내용

- 데이터 처리 및 탐색을 수행해 봅니다.

목차

01. 데이터 준비 및 라이브러리 임포트

목차로 이동하기

In [1]:

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

In [2]:

```
### 한글 폰트 설정
from matplotlib import font_manager, rc
import matplotlib.pyplot as plt
import platform
import matplotlib

path = "C:/Windows/Fonts/malgun.ttf"
if platform.system() == "Windows":
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
elif platform.system()=="Darwin":
    rc('font', family='AppleGothic')
else:
    print("Unknown System")
```

In [3]:

```
con_birth_dat = pd.read_csv("./data/arch/1926_2021.csv")  # 건축, 출생 관련 데이터 셋
birth_dat  = pd.read_csv("./data/arch/birth_data_1970_2021.csv") # 인구 통계 - 출생
con_birth_dat.shape, birth_dat.shape
```

Out[3]:

```
((97, 11), (52, 7))
```

02. 데이터 탐색 및 시각화

목차로 이동하기

In [4]:

```
con_birth_dat
```

Out[4]:

	준공년도	교량	터널	지하차도	인원	출생아수(명)	자연증가건수(명)	조출생률(천명당)	자연증가율(천명당)	합계출산율(명)	출생성비(명)
0	1926.0	1.0	1.0	0.0	8.0	NaN	NaN	NaN	NaN	NaN	NaN
1	1927.0	2.0	0.0	0.0	8.0	NaN	NaN	NaN	NaN	NaN	NaN
2	1928.0	1.0	0.0	0.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN
3	1929.0	0.0	0.0	0.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN
4	1930.0	0.0	0.0	0.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
92	2018.0	613.0	93.0	16.0	2824.0	326822.0	28002.0	6.4	0.5	0.977	105.4
93	2019.0	283.0	54.0	10.0	1348.0	302676.0	7566.0	5.9	0.1	0.918	105.5
94	2020.0	215.0	83.0	8.0	1192.0	272337.0	-32611.0	5.3	-0.6	0.837	104.8
95	2021.0	187.0	40.0	12.0	908.0	NaN	NaN	NaN	NaN	NaN	NaN
96	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

97 rows × 11 columns

In [5]:

```
birth_dat
```

Out[5]:

	기본항목별	출생아수(명)	자연증가건수(명)	조출생률(천명당)	자연증가율(천명당)	합계출산율(명)	출생성비(명)
0	1970	1006645	748056	31.2	23.2	4.530	109.5
1	1971	1024773	787245	31.2	23.9	4.540	109.0
2	1972	952780	742709	28.4	22.2	4.120	109.5
3	1973	965521	698061	28.3	20.5	4.070	104.6
4	1974	922823	674016	26.6	19.4	3.770	109.4
5	1975	874030	603373	24.8	17.1	3.430	112.4
6	1976	796331	529474	22.2	14.8	3.000	110.7
7	1977	825339	576085	22.7	15.8	2.990	104.2
8	1978	750728	498430	20.3	13.5	2.640	111.3
9	1979	862669	622683	23.0	16.6	2.900	106.4
10	1980	862835	585551	22.6	15.4	2.820	105.3
11	1981	867409	629928	22.4	16.3	2.570	107.1
12	1982	848312	602545	21.6	15.3	2.390	106.8
13	1983	769155	514592	19.3	12.9	2.060	107.3
14	1984	674793	438348	16.7	10.8	1.740	108.3
15	1985	655489	415071	16.1	10.2	1.660	109.4
16	1986	636019	396763	15.4	9.6	1.580	111.7
17	1987	623831	380327	15.0	9.1	1.530	108.8
18	1988	633092	397313	15.1	9.5	1.550	113.2
19	1989	639431	402613	15.1	9.5	1.560	111.8
20	1990	649738	408122	15.2	9.5	1.570	116.5
21	1991	709275	467005	16.4	10.8	1.710	112.4
22	1992	730678	494516	16.7	11.3	1.760	113.6
23	1993	715826	481569	16.0	10.8	1.654	115.3
24	1994	721185	478746	16.0	10.6	1.656	115.2
25	1995	715020	472182	15.7	10.3	1.634	113.2
26	1996	691226	450077	15.0	9.8	1.574	111.5
27	1997	675394	430701	14.5	9.3	1.537	108.2
28	1998	641594	395769	13.7	8.4	1.464	110.1
29	1999	620668	372934	13.2	7.9	1.425	109.5
30	2000	640089	391349	13.5	8.2	1.480	110.1
31	2001	559934	316121	11.7	6.6	1.309	109.0
32	2002	496911	249387	10.3	5.2	1.178	109.9
33	2003	495036	248573	10.2	5.1	1.191	108.6

	기본항목별	출생아수(명)	자연증가건수(명)	조출생률(천명당)	자연증가율(천명당)	합계출산율(명)	출생성비(명)
34	2004	476958	230738	9.8	4.8	1.164	108.2
35	2005	438707	192833	9.0	4.0	1.085	107.8
36	2006	451759	207597	9.2	4.2	1.132	107.6
37	2007	496822	250340	10.1	5.1	1.259	106.2
38	2008	465892	219779	9.4	4.4	1.192	106.4
39	2009	444849	197907	9.0	4.0	1.149	106.4
40	2010	470171	214766	9.4	4.3	1.226	106.9
41	2011	471265	213869	9.4	4.3	1.244	105.7
42	2012	484550	217329	9.6	4.3	1.297	105.7
43	2013	436455	170198	8.6	3.4	1.187	105.3
44	2014	435435	167743	8.6	3.3	1.205	105.3
45	2015	438420	162525	8.6	3.2	1.239	105.3
46	2016	406243	125416	7.9	2.5	1.172	105.0
47	2017	357771	72237	7.0	1.4	1.052	106.3
48	2018	326822	28002	6.4	0.5	0.977	105.4
49	2019	302676	7566	5.9	0.1	0.918	105.5
50	2020	272337	-32611	5.3	-0.6	0.837	104.8
51	2021	260562	-57200	5.1	-1.1	0.808	105.1

년도별 출생아수 시각화

In [6]:

```
birth_dat.columns
```

Out[6]:

```
Index(['기본항목별', '출생아수(명)', '자연증가건수(명)', '조출생률(천명당)', '자연증
가율(천명당)', '합계출산율(명)',
       '출생성비(명)'],
      dtype='object')
```

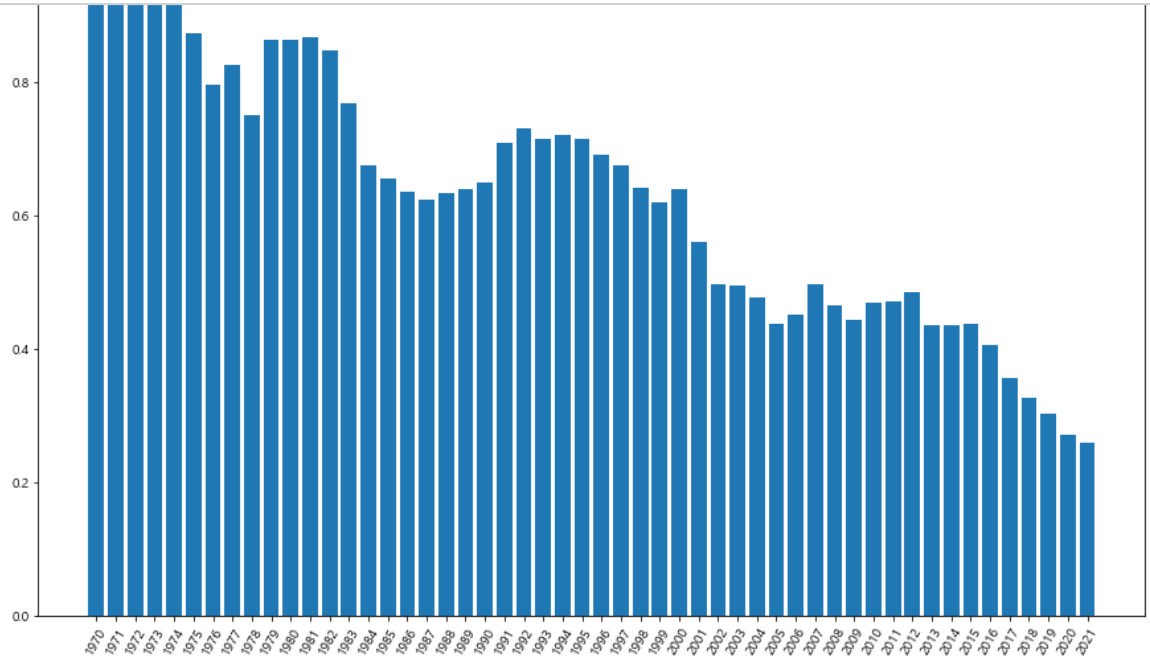
In [7]:

```
birth_dat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   기본항목별        52 non-null     int64
 1   출생아수(명)      52 non-null     int64
 2   자연증가건수(명)   52 non-null     int64
 3   조출생률(천명당)   52 non-null     float64
 4   자연증가율(천명당)  52 non-null     float64
 5   합계출산율(명)     52 non-null     float64
 6   출생성비(명)      52 non-null     float64
dtypes: float64(4), int64(3)
memory usage: 3.0 KB
```

In [8]:

```
plt.figure(figsize=(15,10))
plt.bar(birth_dat['기본항목별'], birth_dat['출생아수(명)'] )
plt.xticks(birth_dat['기본항목별'], rotation =60)
```



- 1971년의 출생아수가 가장 많다.
- 시간이 지나면 지날수록 점점 출생아수가 감소하고 있다.

In [9]:

```
plt.figure(figsize=(15,10))
ticks = birth_dat['기본항목별'].values
plt.bar(birth_dat['기본항목별'], birth_dat['자연증가율(천명당)'] )
plt.xticks(ticks, rotation =60)
```

```
  <matplotlib.axis.XTick at 0x203861dcfa0>,
  <matplotlib.axis.XTick at 0x203861f50a0>,
  <matplotlib.axis.XTick at 0x203861f5820>,
  <matplotlib.axis.XTick at 0x203861f9040>,
  <matplotlib.axis.XTick at 0x203861f9700>,
  <matplotlib.axis.XTick at 0x203861f5cd0>,
  <matplotlib.axis.XTick at 0x203861dc4f0>,
  <matplotlib.axis.XTick at 0x203861f9ac0>,
  <matplotlib.axis.XTick at 0x20386202310>],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
```

In [ ]:

In [10]:

```
con_birth_dat.tail()
```

Out[10]:

	준공년도	교량	터널	지하차도	인원	출생아수(명)	자연증가건수(명)	조출생률(천명당)	자연증가율(천명당)	합계출산율(명)	출생성비(명)
92	2018.0	613.0	93.0	16.0	2824.0	326822.0	28002.0	6.4	0.5	0.977	105.4
93	2019.0	283.0	54.0	10.0	1348.0	302676.0	7566.0	5.9	0.1	0.918	105.5
94	2020.0	215.0	83.0	8.0	1192.0	272337.0	-32611.0	5.3	-0.6	0.837	104.8
95	2021.0	187.0	40.0	12.0	908.0	NaN	NaN	NaN	NaN	NaN	NaN
96	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [11]:

```
con_birth_dat1 = con_birth_dat.iloc[ 0:-1 , :]
con_birth_dat1.tail()
```

Out[11]:

	준공년도	교량	터널	지하차도	인원	출생아수(명)	자연증가건수(명)	조출생률(천명당)	자연증가율(천명당)	합계출산율(명)	출생성비(명)
91	2017.0	1236.0	214.0	36.0	5800.0	357771.0	72237.0	7.0	1.4	1.052	106.3
92	2018.0	613.0	93.0	16.0	2824.0	326822.0	28002.0	6.4	0.5	0.977	105.4
93	2019.0	283.0	54.0	10.0	1348.0	302676.0	7566.0	5.9	0.1	0.918	105.5
94	2020.0	215.0	83.0	8.0	1192.0	272337.0	-32611.0	5.3	-0.6	0.837	104.8
95	2021.0	187.0	40.0	12.0	908.0	NaN	NaN	NaN	NaN	NaN	NaN

In [12]:

```
con_birth_dat1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96 entries, 0 to 95
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   준공년도        96 non-null     float64
 1   교량          96 non-null     float64
 2   터널          96 non-null     float64
 3   지하차도        96 non-null     float64
 4   인원          96 non-null     float64
 5   출생아수(명)     51 non-null     float64
 6   자연증가건수(명)   51 non-null     float64
 7   조출생률(천명당)   51 non-null     float64
 8   자연증가율(천명당)  51 non-null     float64
 9   합계출산율(명)    51 non-null     float64
 10  출생성비(명)     51 non-null     float64
dtypes: float64(11)
memory usage: 8.4 KB
```

데이터 자료형 변환

In [13]:

```
con_birth_dat1['준공년도'] = con_birth_dat1['준공년도'].astype(int)
con_birth_dat1['교량'] = con_birth_dat1['교량'].astype(int)
con_birth_dat1['터널'] = con_birth_dat1['터널'].astype(int)
con_birth_dat1['지하차도'] = con_birth_dat1['지하차도'].astype(int)
con_birth_dat1['인원'] = con_birth_dat1['인원'].astype(int)
con_birth_dat1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96 entries, 0 to 95
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   준공년도        96 non-null     int32
 1   교량          96 non-null     int32
 2   터널          96 non-null     int32
 3   지하차도        96 non-null     int32
 4   인원          96 non-null     int32
 5   출생아수(명)     51 non-null     float64
 6   자연증가건수(명)   51 non-null     float64
 7   조출생률(천명당)   51 non-null     float64
 8   자연증가율(천명당)  51 non-null     float64
 9   합계출산율(명)    51 non-null     float64
 10  출생성비(명)     51 non-null     float64
dtypes: float64(6), int32(5)
memory usage: 6.5 KB


C:₩Users₩totofriend₩AppData₩Local₩Temp₩ipykernel_20548₩3776137475.py:1: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  con_birth_dat1['준공년도'] = con_birth_dat1['준공년도'].astype(int)
C:₩Users₩totofriend₩AppData₩Local₩Temp₩ipykernel_20548₩3776137475.py:2: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  con_birth_dat1['교량'] = con_birth_dat1['교량'].astype(int)
C:₩Users₩totofriend₩AppData₩Local₩Temp₩ipykernel_20548₩3776137475.py:3: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  con_birth_dat1['터널'] = con_birth_dat1['터널'].astype(int)
C:₩Users₩totofriend₩AppData₩Local₩Temp₩ipykernel_20548₩3776137475.py:4: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  con_birth_dat1['지하차도'] = con_birth_dat1['지하차도'].astype(int)
C:₩Users₩totofriend₩AppData₩Local₩Temp₩ipykernel_20548₩3776137475.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  con_birth_dat1['인원'] = con_birth_dat1['인원'].astype(int)
```

In [14]:

```
con_birth_dat1.head()
```

Out[14]:

	준공년도	교량	터널	지하차도	인원	출생아수(명)	자연증가건수(명)	조출생률(천명당)	자연증가율(천명당)	합계출산율(명)	출생성비(명)
0	1926	1	1	0	8	NaN	NaN	NaN	NaN	NaN	NaN
1	1927	2	0	0	8	NaN	NaN	NaN	NaN	NaN	NaN
2	1928	1	0	0	4	NaN	NaN	NaN	NaN	NaN	NaN
3	1929	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN
4	1930	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN

연도별 교량, 터널, 지하차도, 인원을 확인해 보자.

In [15]:

```
con_birth_dat1.columns
```

Out[15]:

```
Index(['준공년도', '교량', '터널', '지하차도', '인원', '출생아수(명)', '자연증가건수(명)', '조출생률(천명당)',
       '자연증가율(천명당)', '합계출산율(명)', '출생성비(명)'],
      dtype='object')
```

In [16]:

```
sel = ['준공년도', '교량', '터널', '지하차도', '인원']
con_dat = con_birth_dat1[sel]
con_dat.head(15)
```

Out[16]:

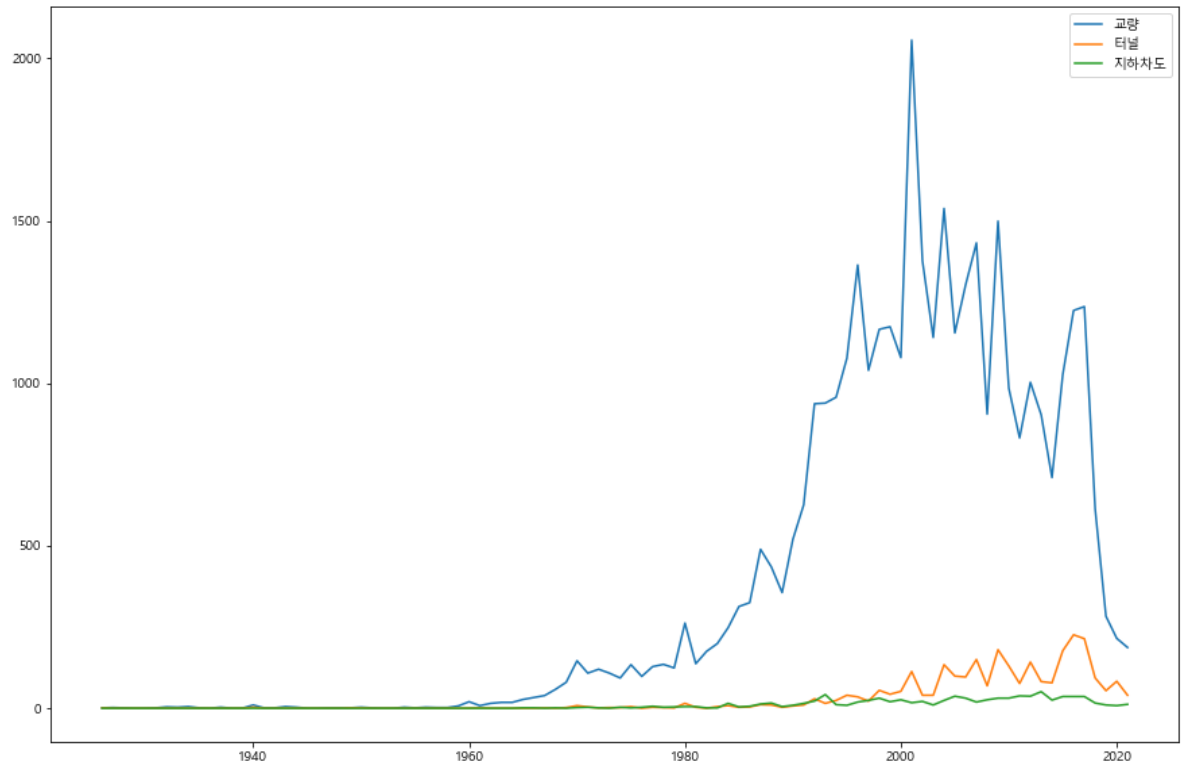
	준공년도	교량	터널	지하차도	인원
0	1926	1	1	0	8
1	1927	2	0	0	8
2	1928	1	0	0	4
3	1929	0	0	0	0
4	1930	0	0	0	0
5	1931	1	0	0	4
6	1932	4	0	0	16
7	1933	3	0	0	12
8	1934	5	0	0	20
9	1935	1	0	0	4
10	1936	0	0	0	0
11	1937	3	0	0	12
12	1938	0	0	0	0
13	1939	0	0	0	0
14	1940	10	3	0	52

In [17]:

```
plt.figure(figsize=(15,10))
plt.plot(con_dat['준공년도'], con_dat['교량'], label="교량")
plt.plot(con_dat['준공년도'], con_dat['터널'], label="터널")
plt.plot(con_dat['준공년도'], con_dat['지하차도'], label="지하차도")
plt.legend()
```

Out[17]:

<matplotlib.legend.Legend at 0x20386ae8490>



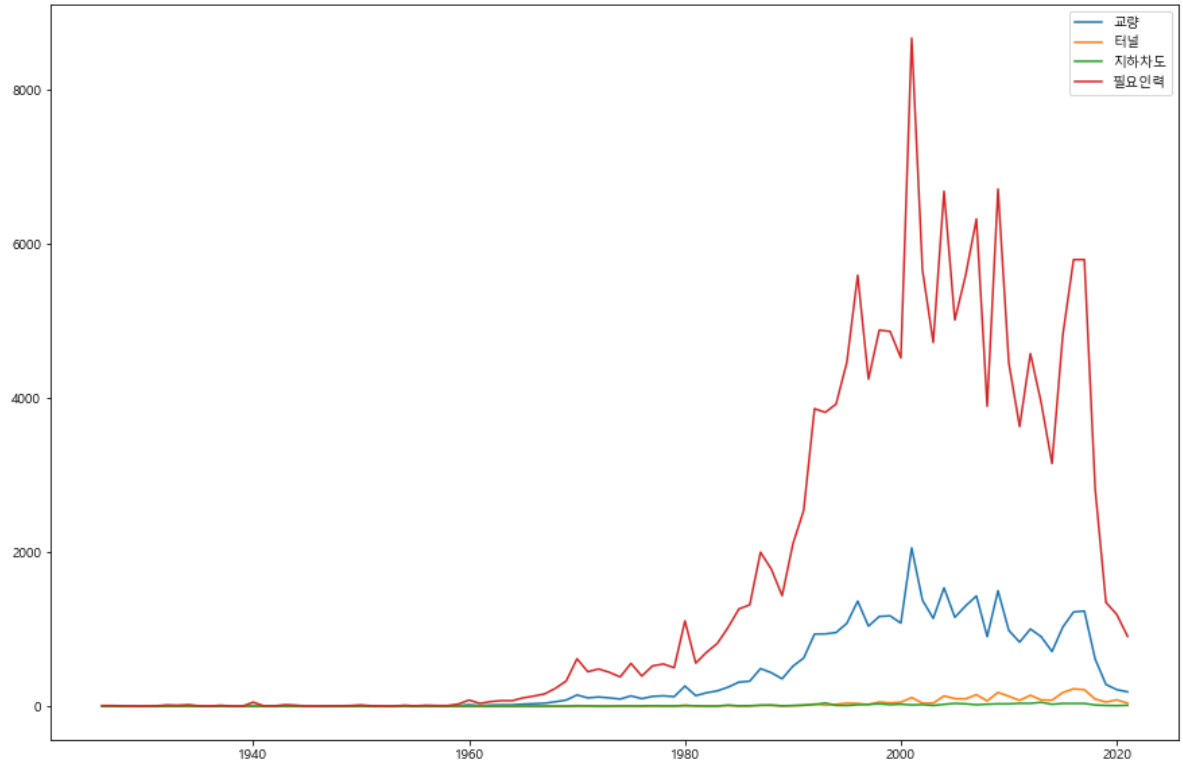
- 년도가 지나면 지날수록 교량이 급격하게 증가하고 있다.

In [18]:

```
plt.figure(figsize=(15,10))
plt.plot(con_dat['준공년도'], con_dat['교량'], label="교량")
plt.plot(con_dat['준공년도'], con_dat['터널'], label="터널")
plt.plot(con_dat['준공년도'], con_dat['지하차도'], label="지하차도")
plt.plot(con_dat['준공년도'], con_dat['인원'], label="필요인력")
plt.legend()
```

Out[18]:

```
<matplotlib.legend.Legend at 0x20386d2e6d0>
```



- 필요인력은 더 많게 필요한 상황이다.