

```
In [183... # 데이터 가져오기
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

# 한글 폰트 지정
import matplotlib
from matplotlib import font_manager, rc
font_loc = "C:/Windows/Fonts/malgunbd.ttf"
font_name = font_manager.FontProperties(fname=font_loc).get_name()
matplotlib.rc('font', family=font_name)
matplotlib.rcParams['axes.unicode_minus'] = False
```

```
In [184... # 데이터 불러오기
birth_dat = pd.read_csv("./DATA(V4).csv")
birth_dat.shape,
```

Out[184]: ((51, 17),)

```
In [185... # 데이터 확인하기(전체를 확인하자.)
birth_dat
```

Out[185]:

	준공 년도	교량	교량_ 누적	터널	터널_ 누적	지하 차도	지하 차도_ 누적	대상 시설 물수	총_ 누적	설계 수명 도래	필요 인원	출생아 수(명)	자연증 가건수 (명)	조출 생률 (천명 당)
0	1970	146	146	8	8	2	2	156	156	NaN	52	1006645	748056	31.2
1	1971	108	254	4	12	3	5	115	271	NaN	90	1024773	787245	31.2
2	1972	120	374	1	13	1	6	122	393	NaN	131	952780	742709	28.4
3	1973	108	482	2	15	0	6	110	503	NaN	168	965521	698061	28.3
4	1974	93	575	2	17	3	9	98	601	NaN	200	922823	674016	26.6
5	1975	134	709	5	22	1	10	140	741	NaN	247	874030	603373	24.8
6	1976	98	807	0	22	3	13	101	842	NaN	281	796331	529474	22.2
7	1977	128	935	3	25	6	19	137	979	NaN	326	825339	576085	22.7
8	1978	135	1070	2	27	3	22	140	1119	NaN	373	750728	498430	20.3
9	1979	124	1194	1	28	4	26	129	1248	NaN	416	862669	622683	23.0
10	1980	262	1456	15	43	5	31	282	1530	NaN	510	862835	585551	22.6
11	1981	137	1593	3	46	5	36	145	1675	NaN	558	867409	629928	22.4
12	1982	175	1768	0	46	1	37	176	1851	NaN	617	848312	602545	21.6
13	1983	199	1967	5	51	1	38	205	2056	NaN	685	769155	514592	19.3
14	1984	248	2215	8	59	15	53	271	2327	NaN	776	674793	438348	16.7
15	1985	313	2528	3	62	4	57	320	2647	NaN	882	655489	415071	16.1
16	1986	325	2853	4	66	6	63	335	2982	NaN	994	636019	396763	15.4
17	1987	489	3342	11	77	13	76	513	3495	NaN	1165	623831	380327	15.0
18	1988	435	3777	10	87	16	92	461	3956	NaN	1319	633092	397313	15.1
19	1989	356	4133	3	90	5	97	364	4320	NaN	1440	639431	402613	15.1
20	1990	520	4653	7	97	9	106	536	4856	NaN	1619	649738	408122	15.2
21	1991	626	5279	10	107	15	121	651	5507	NaN	1836	709275	467005	16.4
22	1992	937	6216	29	136	22	143	988	6495	NaN	2165	730678	494516	16.7
23	1993	939	7155	15	151	42	185	996	7491	NaN	2497	715826	481569	16.0
24	1994	957	8112	24	175	11	196	992	8483	NaN	2828	721185	478746	16.0
25	1995	1077	9189	40	215	9	205	1126	9609	NaN	3203	715020	472182	15.7
26	1996	1364	10553	35	250	19	224	1418	11027	NaN	3676	691226	450077	15.0
27	1997	1040	11593	22	272	24	248	1086	12113	NaN	4038	675394	430701	14.5
28	1998	1166	12759	55	327	31	279	1252	13365	NaN	4455	641594	395769	13.7
29	1999	1174	13933	43	370	20	299	1237	14602	NaN	4867	620668	372934	13.2
30	2000	1079	15012	52	422	26	325	1157	15759	156.0	5253	640089	391349	13.5
31	2001	2056	17068	113	535	17	342	2186	17945	271.0	5982	559934	316121	11.7

	준공 년도	교량	교량_ 누적	터 널	터널_ 누적	지 하 차 도	지 하 차 도_ 누적	대상 시설 물수	총_ 누적	설계 수명 도래	필요 인원	출생아 수(명)	자연증 가건수 (명)	조 출 생 률 (천명 당)
32	2002	1377	18445	40	575	21	363	1438	19383	393.0	6461	496911	249387	10.3
33	2003	1141	19586	40	615	10	373	1191	20574	503.0	6858	495036	248573	10.2
34	2004	1538	21124	134	749	24	397	1696	22270	601.0	7423	476958	230738	9.8
35	2005	1155	22279	99	848	37	434	1291	23561	741.0	7854	438707	192833	9.0
36	2006	1303	23582	96	944	31	465	1430	24991	842.0	8330	451759	207597	9.2
37	2007	1432	25014	150	1094	19	484	1601	26592	979.0	8864	496822	250340	10.1
38	2008	905	25919	69	1163	26	510	1000	27592	1119.0	9197	465892	219779	9.4
39	2009	1499	27418	180	1343	31	541	1710	29302	1248.0	9767	444849	197907	9.0
40	2010	985	28403	130	1473	31	572	1146	30448	1530.0	10149	470171	214766	9.4
41	2011	832	29235	76	1549	38	610	946	31394	1675.0	10465	471265	213869	9.4
42	2012	1003	30238	142	1691	37	647	1182	32576	1851.0	10859	484550	217329	9.6
43	2013	903	31141	82	1773	51	698	1036	33612	2056.0	11204	436455	170198	8.6
44	2014	710	31851	78	1851	25	723	813	34425	2327.0	11475	435435	167743	8.6
45	2015	1027	32878	177	2028	36	759	1240	35665	2647.0	11888	438420	162525	8.6
46	2016	1224	34102	226	2254	36	795	1486	37151	2982.0	12384	406243	125416	7.9
47	2017	1236	35338	214	2468	36	831	1486	38637	3495.0	12879	357771	72237	7.0
48	2018	613	35951	93	2561	16	847	722	39359	3956.0	13120	326822	28002	6.4
49	2019	283	36234	54	2615	10	857	347	39706	4320.0	13235	302676	7566	5.9
50	2020	215	36449	83	2698	8	865	306	40012	4856.0	13337	272337	-32611	5.3



In [186... # 데이터 확인하기(주요부분만 확인하자.)
birth_dat.head()

Out[186]:

	준공 년도	교 량	교 량_누 적	터 널	터 널_누 적	지 하 차 도	지 하 차 도_누 적	대 상 시 설 물 수	총_누 적	설 계 수 명 래	필 요 인 원	출 생 아 수(명)	자 연 증 가 건수 (명)	조 출 생 률 (천명 당)	자 연 증 가 율 (천명 당)	합 계 출 산 율 (명)	출 생 비 (명)
0	1970	146	146	8	8	2	2	156	156	NaN	52	1006645	748056	31.2	23.2	4.53	109.
1	1971	108	254	4	12	3	5	115	271	NaN	90	1024773	787245	31.2	23.9	4.54	109.
2	1972	120	374	1	13	1	6	122	393	NaN	131	952780	742709	28.4	22.2	4.12	109.
3	1973	108	482	2	15	0	6	110	503	NaN	168	965521	698061	28.3	20.5	4.07	104.
4	1974	93	575	2	17	3	9	98	601	NaN	200	922823	674016	26.6	19.4	3.77	109.

2030년도 출생자수는 어떨까?

In [187... train.columns

```
-----  
--  
NameError                                Traceback (most recent call last)  
Input In [187], in <cell line: 1>()  
----> 1 train.columns  
  
NameError: name 'train' is not defined
```

In [188... birth_dat.corr() # 데이터를 확인해 보자.

Out[188]:

	준공년도	교량	교량_누적	터널	터널_누적	지하차도	지하차도_누적	대상시설 물수	총_누적
준공년도	1.000000	0.658309	0.964388	0.794646	0.884320	0.737899	0.958096	0.699909	0.961987
교량	0.658309	1.000000	0.567084	0.681915	0.337146	0.697873	0.514309	0.996119	0.553582
교량_누적	0.964388	0.567084	1.000000	0.840631	0.954728	0.699112	0.993692	0.621370	0.999802
터널	0.794646	0.681915	0.840631	1.000000	0.779643	0.699386	0.820904	0.742645	0.838779
터널_누적	0.884320	0.337146	0.954728	0.779643	1.000000	0.560371	0.974912	0.403237	0.960440
지하차도	0.737899	0.697873	0.699112	0.699386	0.560371	1.000000	0.676909	0.731652	0.692118
지하차도_누적	0.958096	0.514309	0.993692	0.820904	0.974912	0.676909	1.000000	0.570913	0.995342
대상시설물수	0.699909	0.996119	0.621370	0.742645	0.403237	0.731652	0.570913	1.000000	0.608776
총_누적	0.961987	0.553582	0.999802	0.838779	0.960440	0.692118	0.995342	0.608776	1.000000
설계수명도래	0.963743	-0.736312	0.930235	0.208036	0.977511	-0.148658	0.962984	-0.673490	0.937057
필요인원	0.961990	0.553591	0.999802	0.838780	0.960439	0.692126	0.995343	0.608784	1.000000
출생아수(명)	-0.954220	-0.581251	-0.909169	-0.731181	-0.841213	-0.653733	-0.900029	-0.621241	-0.907057

	준공년도	교량	교량_누적	터널	터널_누적	지하차도	지하차도_누적	대상시설물수	총_누적
자연증가건수(명)	-0.955111	-0.549352	-0.921243	-0.736597	-0.869249	-0.636238	-0.916820	-0.592526	-0.9204
조출생률(천명당)	-0.954660	-0.663523	-0.866766	-0.703709	-0.763806	-0.696091	-0.853642	-0.693767	-0.8621
자연증가율(천명당)	-0.956211	-0.628762	-0.879298	-0.708610	-0.791932	-0.675676	-0.869765	-0.662336	-0.8761
합계출산율(명)	-0.859902	-0.659257	-0.709293	-0.564165	-0.584710	-0.639381	-0.695813	-0.673411	-0.7031
출생성비(명)	-0.372187	0.006385	-0.522547	-0.463235	-0.582619	-0.194449	-0.515214	-0.049117	-0.5274

- 년도와 출생자수 데이터를 이용한 선형 모델을 구축해서 2030년도 출생자수를 예측해 보자.

```

In [189]: birth_dat.columns
Out[189]: Index(['준공년도', '교량', '교량_누적', '터널', '터널_누적', '지하차도', '지하차도_누적', '대상시설물수', '총_누적', '설계수명도래', '필요인원', '출생아수(명)', '자연증가건수(명)', '조출생률(천명당)', '자연증가율(천명당)', '합계출산율(명)', '출생성비(명)'], dtype='object')

In [190]: ## 학습모델 import
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

In [191]: birth_dat.shape
Out[191]: (51, 17)

```

```
In [192... ## 데이터 나누기
sel = ['준공년도']
X = birth_dat[sel]
y = birth_dat['출생아수(명)']

X_train = X[0:40] # ~2010년도 까지 데이터를 학습시킬 년도데이터를 확정.
X_test = X[41:] #2011년도 에서부터는 학습된 데이터와 대조해볼 년도데이터 확정.
y_train = y[0:40] # ~2010년도 까지 데이터를 학습시킨다.
y_test = y[41:] # 2011년도 부터 자료는 학습된 데이터와 비교해 본다.
```

```
In [193... X_test
```

```
Out[193]:
```

	준공년도
41	2011
42	2012
43	2013
44	2014
45	2015
46	2016
47	2017
48	2018
49	2019
50	2020

```
In [194... # 모델 생성 및 학습
model = LinearRegression() #모델 생성
model.fit(X_train, y_train) #모델 학습
```

```
Out[194]: LinearRegression()
```

2011~2021년까지 출생자 수, 예측 후 평가해 보자.

```
In [195... pred = model.predict(X_test)
pred
```

```
Out[195]: array([424196.94896811, 411334.52101313, 398472.09305816, 385609.66510319,
        372747.23714822, 359884.80919325, 347022.38123827, 334159.9532833 ,
        321297.52532833, 308435.09737336])
```

```
In [196... from sklearn.metrics import mean_absolute_error
mean_absolute_error(pred, y_test)
```

```
Out[196]: 39292.892026266454
```

*전체오차는 약 38903명정도 차이가 난다.

```
In [197... X_test
```

Out[197]:

	준공년도
41	2011
42	2012
43	2013
44	2014
45	2015
46	2016
47	2017
48	2018
49	2019
50	2020

* 2030년도 출생자수 예측

```
In [198... ### 2030년도 예측수행해 보기
pred_2030 = model.predict([[2030]])
pred_2030
```

C:\Users\Whyun5\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

Out[198]: array([179810.81782364])

결과 : 2030년도의 출생자수는 17980명으로 예상된다.

* 시설물의 수 예측

시설물의 수는 교량, 터널, 지하철도를 한꺼번에 예측하기에 어려우니 교량만 해보자.

년도와 교량 시설물수 데이터를 이용한 선형모델을 구축해서 2030년 시설물 수를 구하자.

```
In [199... # 데이터 나누기
sel = ['준공년도']
X = birth_dat[sel]
y = birth_dat['교량']

X_train = X[0:40]
X_test = X[41:]
y_train = y[0:40]
ytrain = y[41:]
```

```
In [200... ## 모델 생성 및 학습
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[200]: LinearRegression()


```
In [201... pred = model.predict(X_test)
pred
```

```
Out[201]: array([1593.0565666 , 1635.27082552, 1677.48508443, 1719.69934334,
        1761.91360225, 1804.12786116, 1846.34212008, 1888.55637899,
        1930.7706379 , 1972.98489681])
```

```
In [202... ## 전체오차 검색
mean_absolute_error(pred, y_test)
```

```
Out[202]: 391414.3792682927
```

*약 전체오차가 39292 개소 정도 오차가 난다. 직선형으로 분석하면 안된다는것을 알았다.

```
In [203... ## 그럼에도 불구하고 2030년 교량의 갯수 예측
pred_2030 = model.predict([[2030]])
pred_2030
```

```
C:\Users\Whyun5\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(
```

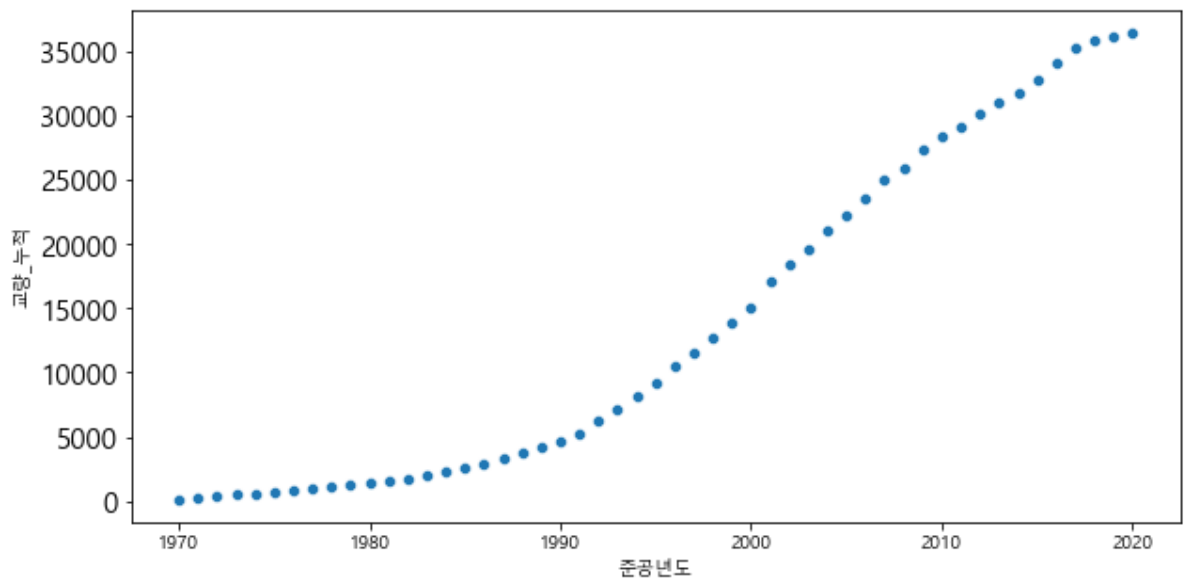
```
Out[203]: array([2395.12748593])
```

- 신설교량의 갯수는 2395개로 예측되었다.

2021년 ~ 2030년까지 매년 예측 교량의 갯수 파악하기

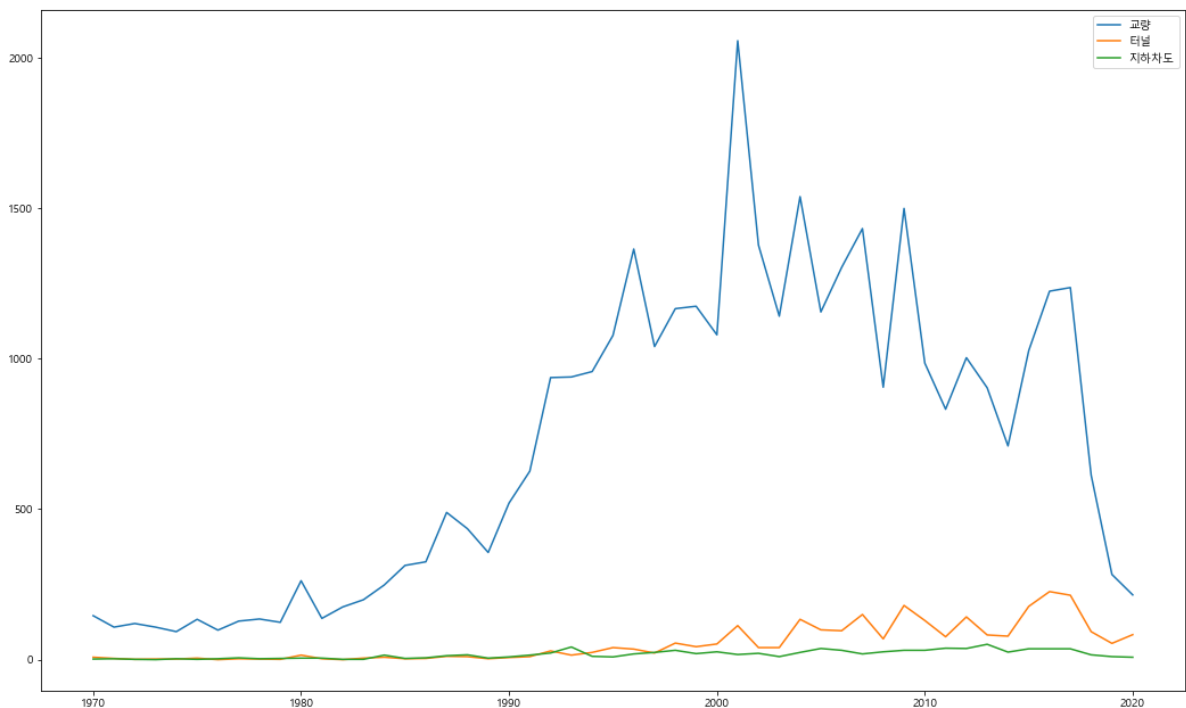
```
In [204... for i in range(2021,2030+1,1) :
    pred_i = model.predict([[i]])
    print(i, pred_i)
```

```
2021 [2015.19915572]
2022 [2057.41341463]
2023 [2099.62767355]
2024 [2141.84193246]
2025 [2184.05619137]
2026 [2226.27045028]
2027 [2268.48470919]
2028 [2310.69896811]
2029 [2352.91322702]
2030 [2395.12748593]
```

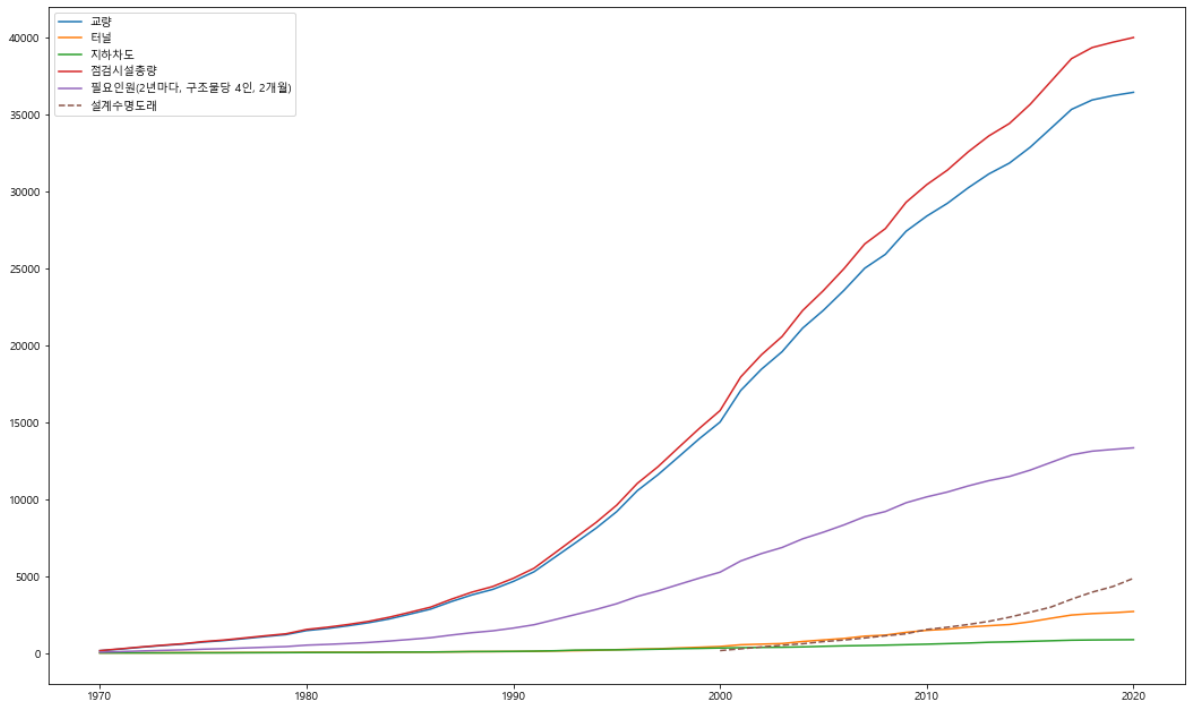
```
In [219... plt.figure(figsize=(18,11))
plt.plot(birth_dat['준공년도'], birth_dat['교량'], label="교량")
plt.plot(birth_dat['준공년도'], birth_dat['터널'], label="터널")
plt.plot(birth_dat['준공년도'], birth_dat['지하차도'], label="지하차도")
plt.legend()
```

Out[219]: <matplotlib.legend.Legend at 0x289751ccc10>



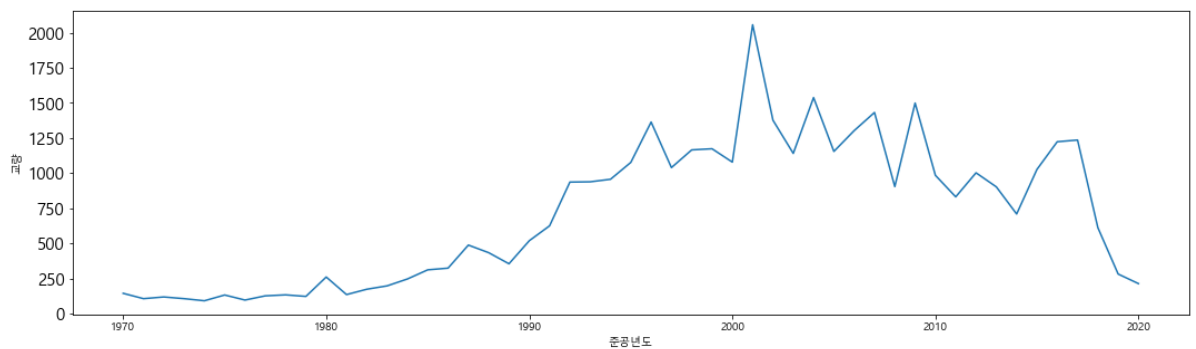
```
In [220... plt.figure(figsize=(18,11))
plt.plot(birth_dat['준공년도'], birth_dat['교량_누적'], label="교량")
plt.plot(birth_dat['준공년도'], birth_dat['터널_누적'], label="터널")
plt.plot(birth_dat['준공년도'], birth_dat['지하차도_누적'], label="지하차도")
plt.plot(birth_dat['준공년도'], birth_dat['총_누적'], label="점검시설총량")
plt.plot(birth_dat['준공년도'], birth_dat['필요인원'], label="필요인원(2년마다, 구조")
plt.plot(birth_dat['준공년도'], birth_dat['설계수명도래'], linestyle="--", label="")
plt.legend()
```

Out[220]: <matplotlib.legend.Legend at 0x28975b38640>



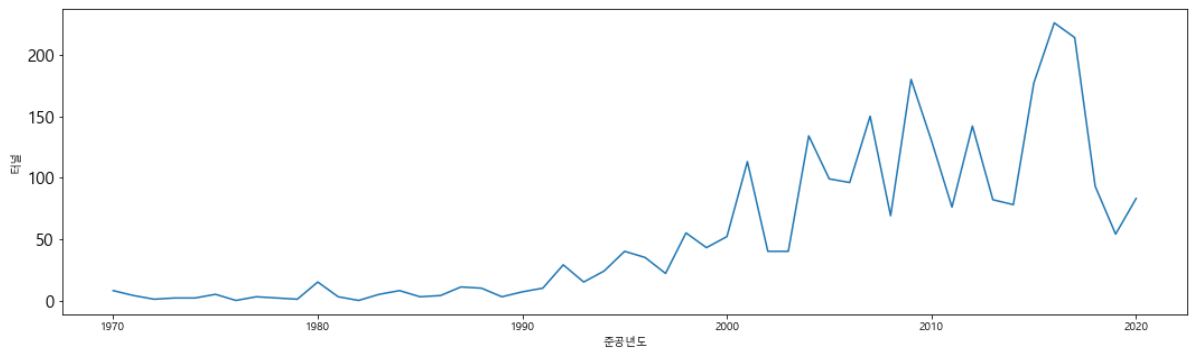
```
In [214]: plt.figure(figsize=(18,5))
sns.lineplot(x=birth_dat['준공년도'], y=birth_dat['교량'])
plt.yticks(fontsize=15)
```

```
Out[214]: (array([-250.,   0.,  250.,  500.,  750., 1000., 1250., 1500., 1750.,
        2000., 2250.]),
 [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, ''),
  Text(0, 0, '']))
```



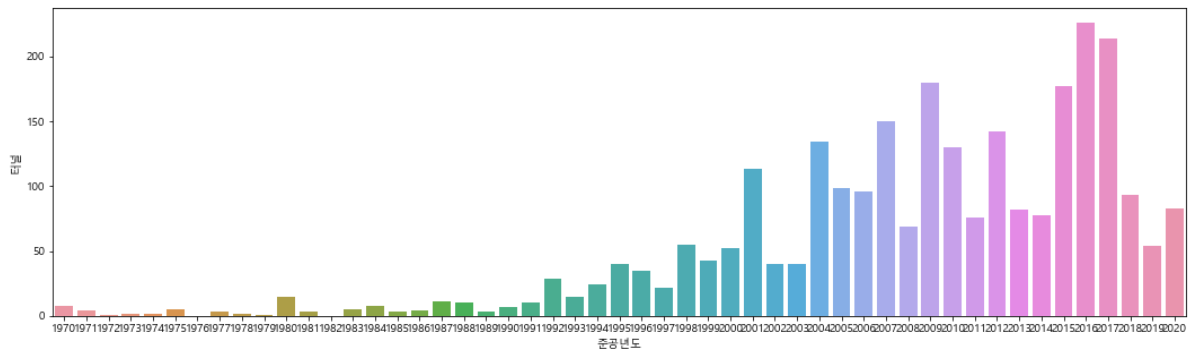
```
In [152]: plt.figure(figsize=(18,5))
sns.lineplot(x=birth_dat['준공년도'], y=birth_dat['터널'])
plt.yticks(fontsize=15)
```

```
Out[152]: (array([-50.,   0.,  50., 100., 150., 200., 250.]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '']))
```



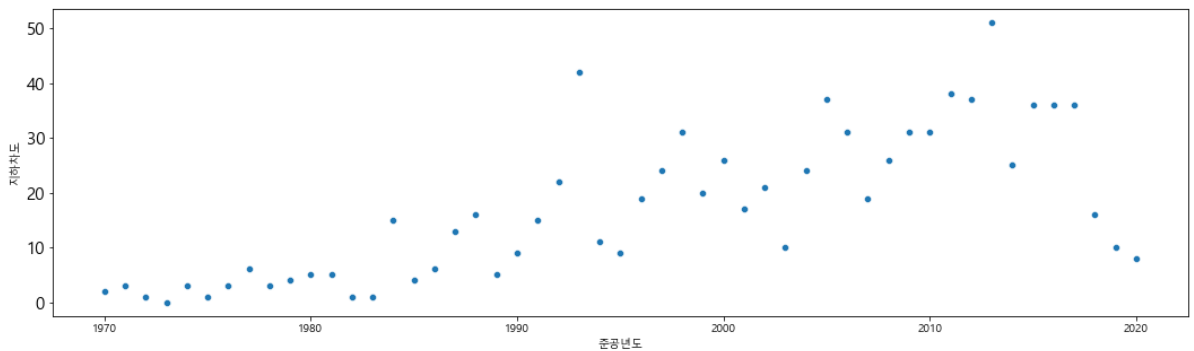
```
In [49]: plt.figure(figsize=(18,5))
sns.barplot(x=birth_dat['준공년도'], y=birth_dat['터널'])
```

Out[49]: <AxesSubplot:xlabel='준공년도', ylabel='터널'>



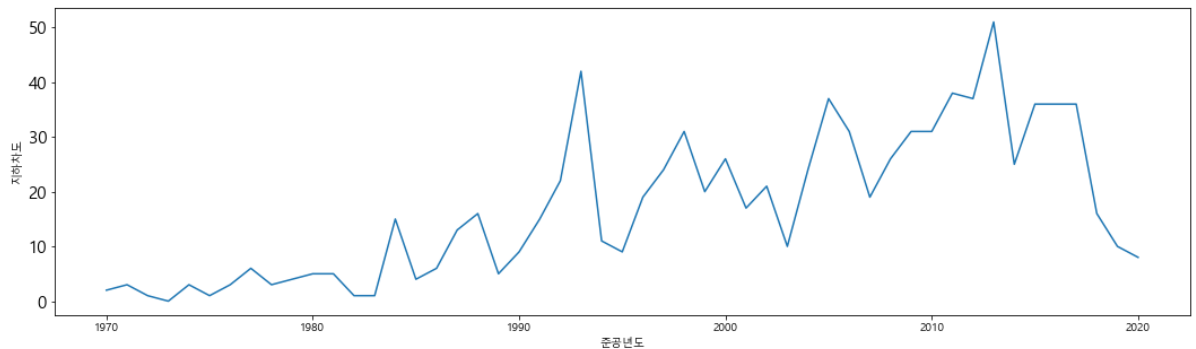
```
In [155... plt.figure(figsize=(18,5))
sns.scatterplot(x=birth_dat['준공년도'], y=birth_dat['지하차도'])
plt.yticks(fontsize=15)
```

Out[155]: (array([-10., 0., 10., 20., 30., 40., 50., 60.]),
[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, '')])



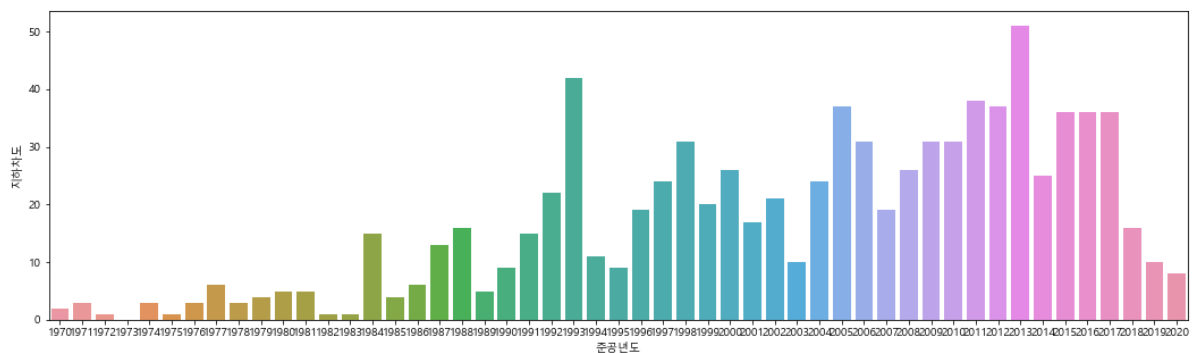
```
In [154... plt.figure(figsize=(18,5))
sns.lineplot(x=birth_dat['준공년도'], y=birth_dat['지하차도'])
plt.yticks(fontsize=15)
```

```
Out[154]: (array([-10.,  0., 10., 20., 30., 40., 50., 60.]),
 [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 [47]: plt.figure(figsize=(18,5))
sns.barplot(x=birth_dat['준공년도'], y=birth_dat['지하차도'])
```

```
Out[47]: <AxesSubplot:xlabel='준공년도', ylabel='지하차도'>
```



```
In [ ]:
```