

출산율 관련 데이터 분석 및 시각화 보고서

2020444101
민찬기

GitHub

<https://github.com/MCK-OOTS/DataVisualization/blob/main/DataVisualization.ipynb>

주제 선정 이유

- 최근 전례가 없는 출산율을 보이고 있는 대한민국의 출산율에 대한 원인과 관련 지표에 대한 궁금증을 해결하고 선택
- 흔히 말하는 여성의 사회진출, 청년들의 늦은 사회진출 같은 지표가 출산율과 연관이 있을까? 하는 궁금증을 해결하고자 이 주제를 선정

개발 환경 및 라이브러리, API

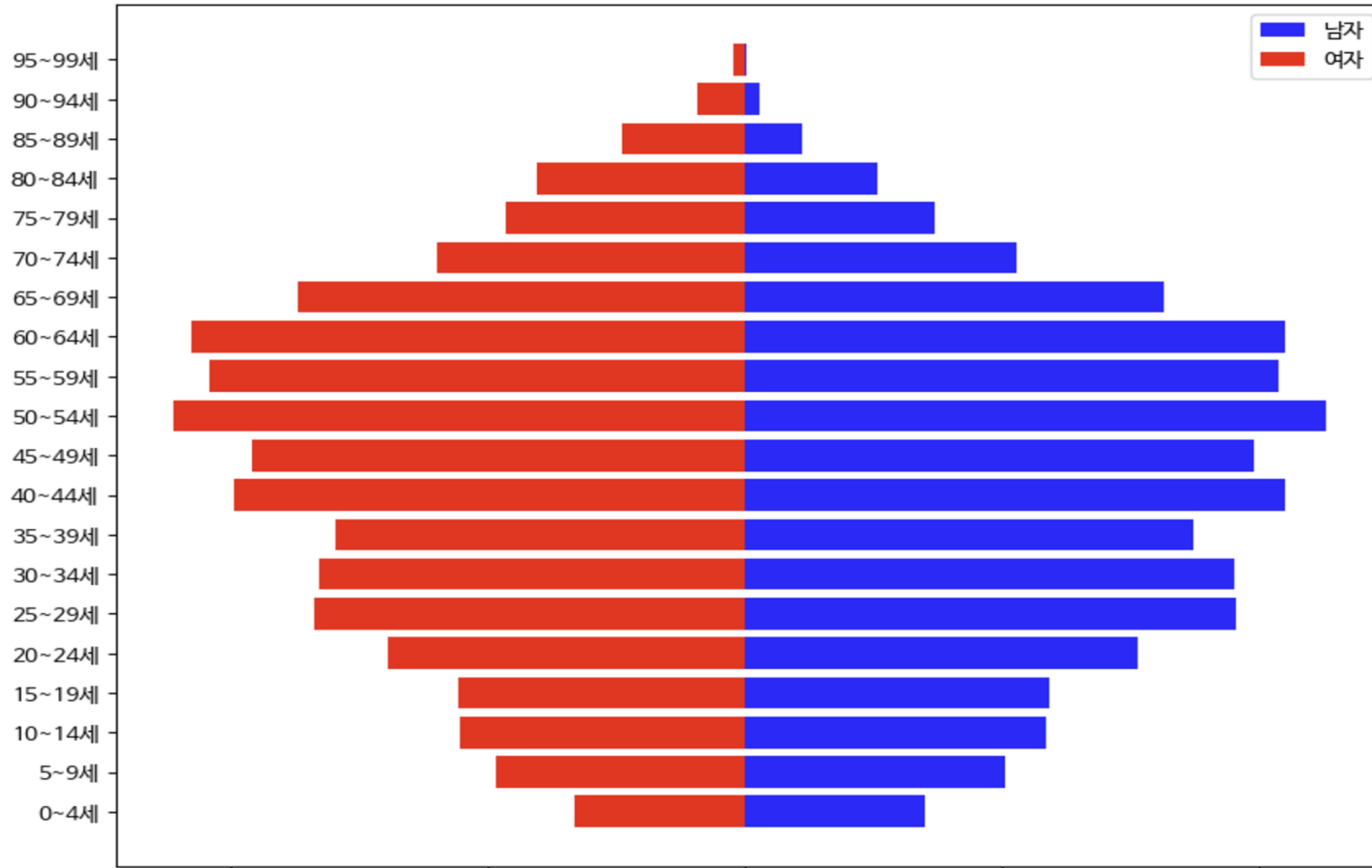


Google Colaboratory



seaborn

대한민국의 저출산이 심각한 문제가 되는 이유



1. 저출산 자체의 문제
2. 젊은 층의 인구 감소
3. 고령화 사회
4. 노동력 부족

이게 지속될 경우 젊은 층이 2~3배 되는 윗 세대를 부양해야하는 엄청난 부담을 짊어지게 된다.

데이터 수집(Kosis 국가통계포털의 API)

- 요청 URL : <https://kosis.kr/openapi/statisticsList.do>

- 요청 파라미터

- apiKey : 발급 ApiKey

- itmId : 데이터 분류 코드

- objL1 : 데이터 분류 코드

- objL2

-

- objL8

- Format : 응답형식(JSON)

- jsonVD : JSON응답형식 (Y)

- PreSe : 기간별 데이터 요청(Y, M)

- newEstPrdCnt : 최신 데이터 연도

- orgId : 데이터 제공 조직 코드

- tblId : 테이블ID

```
# API URL (KOSIS 국가 통계 포털)
```

```
url = "https://kosis.kr/openapi/Param/statisticsParameterData.do"
```

```
# 요청 파라미터
```

```
params = {
```

```
    "apiKey": "OWVmNjBiOTgzZjhiNzYyNjYwNjU0YWUxYTlkYTZMM=", # API 키
```

```
    "itmId": "T01+T02+", # 데이터 분류 코드
```

```
    "objL1": "00+", # 데이터 분류 코드
```

```
    "objL2": "005+010+015+020+025+030+035+040+045+050+055+060+065+070+075+080+085+090+095+100+", # 데이터 분류 코드
```

```
    "format": "json", # 응답 형식
```

```
    "jsonVD": "Y", # JSON 형식
```

```
    "prdSe": "Y", # 기간별 데이터 요청
```

```
    "newEstPrdCnt": "1", # 최신 3년 데이터
```

```
    "orgId": "101", # 데이터 제공 조직 ID (통계청)
```

```
    "tblId": "DT_11N1503", # 테이블 ID (인구 구조 관련 테이블)
```

```
}
```

데이터 수집(OECD Data Exploer API)

- API 데이터 쿼리 구문 :
- {호스트 URL}/{에이전시 식별자},{데이터세트 식별자},{데이터세트 버전}/
{데이터선택}?{기타 선택 매개변수}

API URL 설정

```
url = (  
    "https://sdmx.oecd.org/public/rest/data/" #HOST URL  
    "OECD.ECO.MPD,DSD_AN_HOUSE_PRICES@DF_HOUSE_PRICES," #Agency Identifier  
    "1.0/OECD+USA+GBR+CHE+TUR+SWE+ESP+SVN+SVK+PRT+POL+NZL+NOR+NLD+MEX+LVA+LTU+LUX+KOR+JPN+HUN+ISR+ITA+IRL+ISL+GRC+DEU+FRA+EST+FIN+I  
    "startPeriod=2023&endPeriod=2023&dimensionAtObservation=AllDimensions&format=genericdata" #Other Option  
)
```

데이터 수집(OECD .xlsx)

- OECD API에서 제공하지 않는 데이터는 OECD dataset의 .xlsx파일을 GoogleDrive에 저장 후 불러내어 사용

SF1.2 Children in families

[PDF](#) [XLSx](#)

SF1.3 Further information on living arrangements of children

[PDF](#) [XLSx](#)

SF1.4 Population by age of children and youth dependency ratio

[PDF](#) [XLSx](#)

SF1.5 Living conditions of children

[PDF](#) [XLSx](#)

Fertility indicators

SF2.1 Fertility rates

[PDF](#) [XLSx](#) [@OECD_Social](#)

SF2.2 Ideal and actual number of children

(under development)

SF2.3 Age of mothers at childbirth and age-specific fertility

[PDF](#) [XLSx](#)

SF2.4 Share of births outside of marriage

[PDF](#) [XLSx](#)

SF2.5 Childlessness

[PDF](#) [XLSx](#)

Marital and partnership status

SF3.1 Marriage and divorce rate

[PDF](#) [XLSx](#)

SF3.2 Family dissolution and children

[PDF](#) [XLSx](#)

SF3.3 Cohabitation rate and prevalence of other forms of partnership

[PDF](#) [XLSx](#)

내 드라이브 > 3-2BigData ▾

유형 ▾

사람 ▾

수정 날짜 ▾

이름 ↑

소유자



가계지출.xlsx

나



남성육아휴가.xlsx

나



여성육아휴가.xlsx

나



조혼인율.xlsx

나



첫째 출산나이.xlsx

나



초혼나이.xlsx

나



출산나이.xlsx

나

대한민국 인구 수 데이터 수집

1. KOSIS 국가 통계 포털의 API를 통해 데이터 수집
2. itmlId : 총인구_남자(명), 총인구_여자(명)
3. objL1 : 전국
4. objL2 : 원하는 데이터의 나이대(0~99세)
5. newEstPrdCnt : 최근 1년의 데이터
6. orgId : 통계청의 제공 데이터
7. tblId : 데이터 가져올 테이블ID

```
# API URL (KOSIS 국가 통계 포털)
url = "https://kosis.kr/openapi/Param/statisticsParameterData.do"

# 요청 파라미터
params = {
    "apiKey": "0WVmNjBiOTgzZjhiNzYyNjYwNjU0YWUxYTlkYTZlMmM=", # API 키
    "itmlId": "T01+T02+", # 데이터 분류 코드
    "objL1": "00+", # 데이터 분류 코드
    "objL2": "005+010+015+020+025+030+035+040+045+050+055+060+065+070+075+080+085+090+095+100+", # 데이터 분류 코드
    "format": "json", # 응답 형식
    "jsonVD": "Y", # JSON 형식
    "prdSe": "Y", # 기간별 데이터 요청
    "newEstPrdCnt": "1", # 최신 3년 데이터
    "orgId": "101", # 데이터 제공 조직 ID (통계청)
    "tblId": "DT_1IN1503", # 테이블 ID (인구 구조 관련 테이블)
}

# API 요청 보내기
response = requests.get(url, params=params)

# JSON 데이터 파싱 및 데이터 확인
population_data = response.json()
#print(population_data)
```


OECD 주택 가격 지수 데이터 수집

- OECD 데이터 요청 파라미터에 맞춰서 작성

```
# API URL 설정
url = (
    "https://sdmx.oecd.org/public/rest/data/" #HOST URL
    "OECD.ECO.MPD.DSD_AN_HOUSE_PRICES@F_HOUSE_PRICES," #Agency Identifier
    "1.0/OECD+USA+GBR+CHE+TUR+SWE+ESP+SIN+SVK+PRT+POL+NZL+NOR+NLD+MEX+LVA+LTU+LUX+KOR+JPN+HUN+ISR+ITA+IRL+ISL+GRC+DEU+FRA+EST+FIN+DNK+CZE+CRI+CHL+COL+CAN+BEL+AUT+AUS.A.HPI_YDH.?" #Data S
    "startPeriod=2023&endPeriod=2023&dimensionAtObservation=AllDimensions&format=genericdata" #Other Option
)

# 데이터 요청
response = requests.get(url)

# XML 데이터 파싱
root = ET.fromstring(response.text)
```

OECD평균 초혼 연령 데이터 수집 및 전처리

- API로 제공하지 않는 데이터 중 **xlsx**파일로 제공하는 데이터는 구글 드라이브에 저장한 후에 사용
- 파일을 오픈시 국가를 인덱스로 잡고 DF에 저장

	Male mean age at first marriage			Female mean age at first marriage		
	1990	2000	2019	1990	2000	2019
Spain	27.8	30.2	36.1	25.6	28.1	33.9
Sweden	30.3	33	36.7	27.7	30.4	34.1
Norway	29	30.9	35.7	26.4	28.4	33.1
Italy	28.9	30.9	35.5	25.9	27.8	32.7
France	..	30.7	35.2	..	28.4	33.1
Denmark	30.5	32.5	35.1	27.8	29.9	32.8
Iceland	29.4	33.3	34.4	26.9	30.6	32.4
Portugal	26.6	27.4	33.2	24.6	25.2	31.5
Luxembou	27.7	30.3	34.8	25.6	27.4	32.1
United Kin	27.2	30.5	33.7	25.2	28.2	31.8
Austria	27.7	30	34.7	25.2	27.4	32
Belgium	26.5	29.1	33.5	24.4	26.9	31.2
Ireland	28.7	..	33.8	26.6	..	31.9
Finland	28.4	30.5	34.2	26.3	28.3	31.9
Netherland	28.5	30.7	34.4	26.1	28	31.9
Chile	..	27.7	32.7	..	25.6	31.5
OECD-26 ave	27.7	29.8	33.5	25.2	27.3	31.2
Germany	28.2	30.5	34	25.5	27.7	31.2
Slovenia	26.9	29.9	33.9	23.9	27	31.2
EU-26 aver	-	-	33.2	-	-	30.7
Switzerland	29.5	30.8	33.1	27	28.2	30.7

```
#OECD에서 제공하는 .xlsx 파일 이용
#GoogleDrive에 저장 후 사용
#Google Drive 마운트
from google.colab import drive
drive.mount('/content/drive', force_remount=True)

#파일 경로
file_path = '/content/drive/My Drive/3-2BigData/'

#Excel 파일 읽기
try:
    df_OECD_avg_age = pd.read_excel(file_path+'초혼나이.xlsx', engine='openpyxl', header=1, index_col=0) #index_col = 국가 이름을 인덱스로 지정
    df_OECD_avg_age = df_OECD_avg_age.loc[:, [2019, '2019.1']] #.xlsx의 파일을 불러오면서 header=1 옵션으로 인해 같은 연도 구분을 위해 여성의 데이터 쪽의 연도는 '.1'이 붙으면서 문자 취급
    df_OECD_avg_age.columns = ['OECD 남성평균 초혼연령', 'OECD 여성평균 초혼연령'] #컬럼 이름 변경
    df_OECD_avg_age['전체 평균 초혼연령'] = df_OECD_avg_age[['OECD 남성평균 초혼연령', 'OECD 여성평균 초혼연령']].mean(axis=1) #남녀의 연령 데이터를 바탕으로 평균 값 구해서 컬럼추가
    df_OECD_avg_age = df_OECD_avg_age.sort_values(by='전체 평균 초혼연령', ascending=True) #내림차순 정렬

except Exception as e:
    print("파일을 읽는 중 오류 발생:", e)
```

대한민국 출산율 데이터 전처리

df_fertility – 비교 국가들의 출산율 데이터

1. DF로 저장
2. 원하는 데이터 컬럼만 추출 후 저장
3. 특정국가의 데이터만 따로 추출해서 저장
 - 대표적인 출산율 개선국가(독일), 높은 출산율국가(프랑스)
 - 대표적인 저출산국가(일본,이탈리아)
4. 컬럼 이름 변경
5. 저출산 국가와 높은 출산율 국가를 다시 분리

Df_all_fertility – 출산율 하위 10개국가의 출산율 데이터

1. DF로 저장
2. 원하는 컬럼 데이터만 추출 후 저장
3. 가장 최근 연도인 2021년도 데이터만 저장
4. 컬럼명 변경
5. 기본적으로 API값이 수치 데이터가 아닌 문자 데이터이기 때문에 float형으로 형변환
6. 오름차순으로 정렬 후 head()를 통해 10개국가의 데이터만 저장

```
#전처리
df_fertility = pd.DataFrame(fertility_data) # list > DataFrame
df_fertility = df_fertility[['PRD_DE', 'C1_NM', 'DT']] #원하는 데이터 키 값만 저장
#대표적인 저출산국가(일본,이탈리아)와 높은 출산율(프랑스), 출산율 개선국가(독일)의 출산율 변화 비교
df_fertility = df_fertility[df_fertility['C1_NM'].isin(['대한민국', '일본', '프랑스', '독일', '이탈리아'])] #원하는 국가의 데이터만 저장
df_fertility.columns = ['연도', '국가', '출산율(명)'] # 데이터 컬럼 이름 변경
#저출산국가와 높은 출산율 국가 데이터를 다시 분리
df_high_fertility = df_fertility[df_fertility['국가'].isin(['프랑스', '독일', '대한민국'])]
df_low_fertility = df_fertility[df_fertility['국가'].isin(['일본', '이탈리아', '대한민국'])]

#전 세계 데이터
df_all_fertility = pd.DataFrame(fertility_data) # list > DataFrame
df_all_fertility = df_all_fertility[['PRD_DE', 'C1_NM', 'DT']] #원하는 데이터 키 값만 저장
df_all_fertility = df_all_fertility[df_all_fertility['PRD_DE'] == '2021'] # 2021년도 데이터만 저장
df_all_fertility.columns = ['연도', '국가', '출산율(명)'] # 데이터 컬럼 이름 변경

#하위 10개국 데이터만 저장
df_all_fertility['출산율(명)'] = df_all_fertility['출산율(명)'].astype(float) #하위 50개 국가 필터링을 위해 데이터를 float로 변경
df_all_fertility = df_all_fertility.sort_values(by="출산율(명)", ascending=True).head(10) #오름차순 정렬 후 10개의 국가 데이터만 저장

print(df_fertility)
print(df_all_fertility)
```

대한민국 청년(19~24세)실업률 데이터 전처리

1. DF로 바로 저장시 스칼라 값 에러로 인해 리스트로 데이터 가공 후 DF로 저장
2. 해당 데이터가 29년도 까지 받아지므로 데이터 기간을 선택해서 데이터 가공

```
# 단일 값(스칼라 값) 에러로인해 리스트로 저장 후 DF로 저장
```

```
data_list = []  
for item in unemploy_data:  
    data_list.append({  
        "연도": item.get("PRD_DE"),  
        "국가": item.get("C2_NM"),  
        "청년 실업률(%)": item.get("DT")  
    })
```

```
df_unemploy = pd.DataFrame(data_list)
```

```
#2000~ 2021까지의 데이터만 저장 (2029년도 까지 데이터 제공)
```

```
df_unemploy = df_unemploy[df_unemploy['연도'].isin([str(year) for year in range(2000, 2022)])]
```

```
# 컬럼이름 변경
```

```
df_unemploy.columns = ['연도', '국가', '청년 실업률(%)']
```

```
print(df_unemploy)
```

OECD 주택 가격 지수 데이터 전처리

- 개발자가이드에 CSV, XML, JSON을 제공한다고 되어 있지만 XMI 밖에 사용되지 않는다.
- 1. XML에서 관찰자(Obs)를 노드 기준으로 탐색
- 2. 각 관찰자 노드에서 관측값의 세부정보 추출
(ObsKey: REF_AREA(국가), TIME_PERIOD(기간))
(ObsValue) : 실제 관측된 데이터 값(주택가격지수)
- 3. 데이터를 기준으로 정렬하기 위해 수치 데이터로 형변환
- 4. 시각화했을 때 높은게 위에서부터 나오도록 다시 정렬

```
# 데이터 저장 리스트
countries = [] #나라
years = [] #연도
values = [] #값

# 각 'Obs' 항목에서 데이터 추출
for obs in root.findall('.//generic:Obs', namespaces={'generic': 'http://www.sdmx.org/resources/sdmx/schemas/v2.1/data/generic'}): #관측값 obs를 namespaces에서 찾을
# 기간, 국가, 값 추출
for obs_key in obs.findall('.//generic:ObsKey', namespaces={'generic': 'http://www.sdmx.org/resources/sdmx/schemas/v2.1/data/generic'}): #obs를 바탕으로 국가와 기간 값 추출
for value_elem in obs_key.findall('.//generic:Value', namespaces={'generic': 'http://www.sdmx.org/resources/sdmx/schemas/v2.1/data/generic'}):
if value_elem.attrib.get('id') == 'REF_AREA': #국가
country = value_elem.attrib.get('value')
if value_elem.attrib.get('id') == 'TIME_PERIOD': #기간
year = value_elem.attrib.get('value')

# 값 추출
obs_value = obs.find('.//generic:ObsValue', namespaces={'generic': 'http://www.sdmx.org/resources/sdmx/schemas/v2.1/data/generic'})
if obs_value is not None:
value = obs_value.attrib.get('value') #주택가격

# 데이터 리스트에 추가
if country and year and value:
countries.append(country)
years.append(year)
values.append(value)

# DataFrame으로 변환
df_hous_price = pd.DataFrame({
'Country': countries,
'Year': years,
'Value': values
})

df_hous_price['Value'] = df_hous_price['Value'].astype(float) #데이터 정렬을 위해 수치 데이터로 형변환
df_hous_price = df_hous_price.sort_values(by='Value', ascending=True)

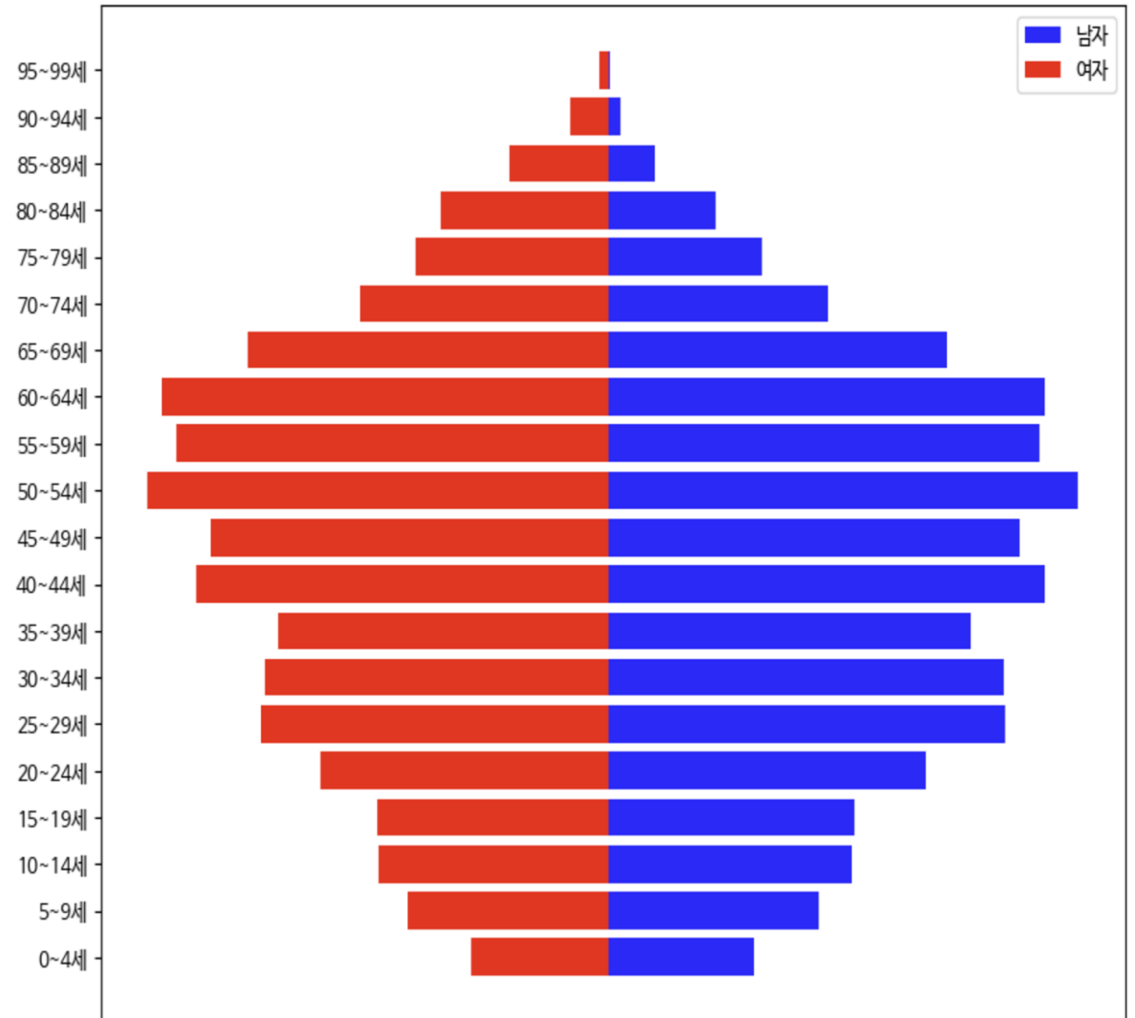
# 데이터 출력
print(df_hous_price.head())
```

대한민국 인구 수 데이터 시각화

- 1. 수평 막대 그래프를 활용하여 인구피라미드 그래프로 시각화
- 2. 여자의 데이터 값은 음수로 처리해서 왼쪽으로 그래프가 형성

In [56]:

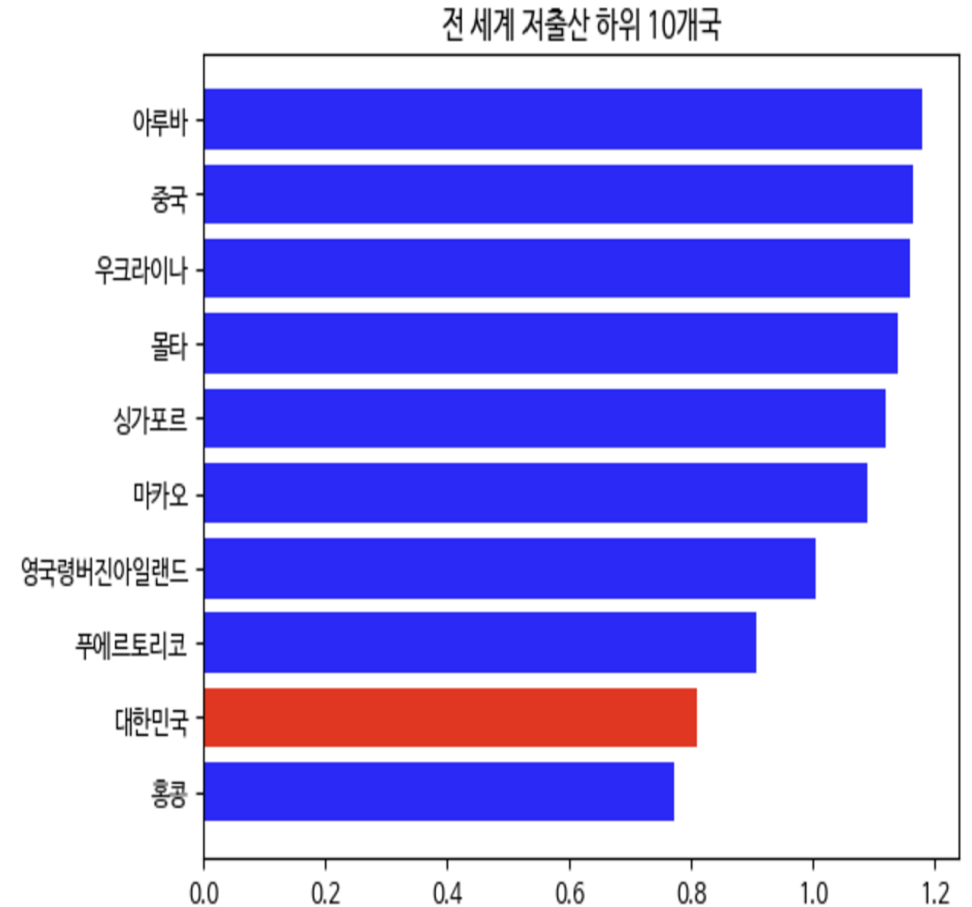
```
# 피라미드 형식으로 시각화 (여성 데이터를 음수로 바꿔서 왼쪽에 표시)
plt.figure(figsize=(10, 8))
plt.barh(male_population.index, male_population, color='blue', label='남자')
plt.barh(female_population.index, -female_population, color='red', label='여자')
#범례 추가
plt.legend()
```



대한민국 출산율 데이터 시각화

1. 수평 그래프를 활용해서 시각화
2. 대한민국 구분이 쉽도록 다른 색으로 설정

```
#2021 저출산 10개국  
#그래프 시각화  
plt.title('전 세계 저출산 하위 10개국')  
plt.barh(  
    df_all_fertility['국가'],  
    df_all_fertility['출산율(명)'],  
    color=['blue' if country != '대한민국' else 'red' for country in df_all_fertility['국가']]  
)
```



대한민국 출산율 데이터 시각화

1. 시각화를 위해 수치 데이터로 변경
2. 연도별 데이터를 lineplot을 활용하여 시각화

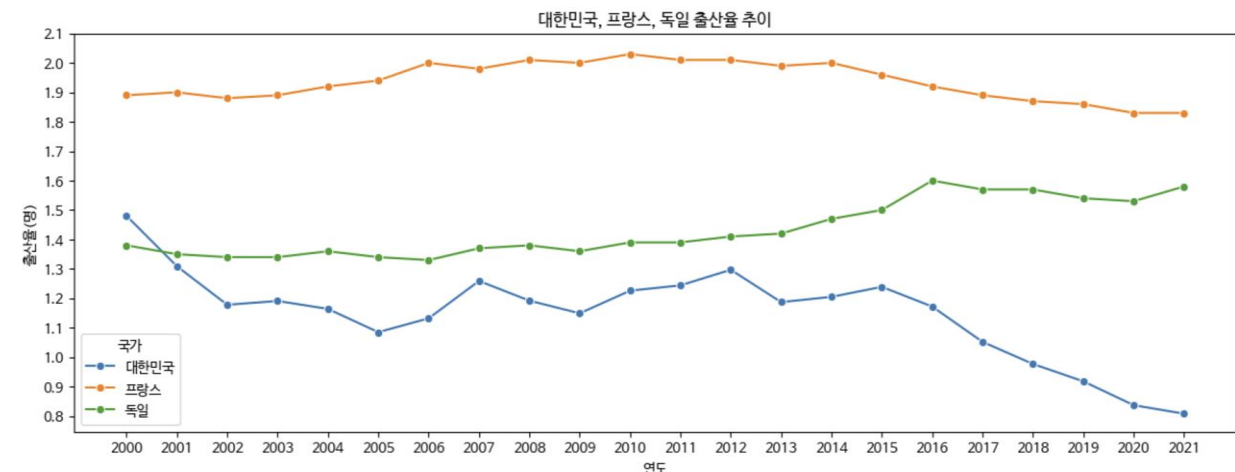
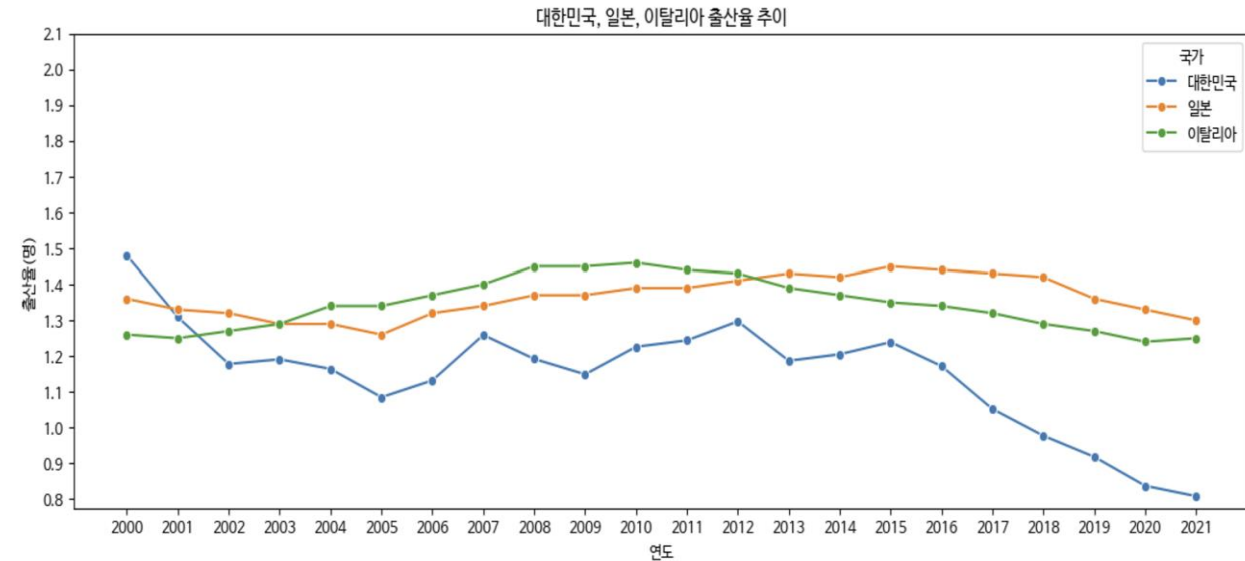
#수치데이터 변경

```
df_high_fertility['출산율(명)'] = pd.to_numeric(df_high_fertility['출산율(명)'])  
df_low_fertility['출산율(명)'] = pd.to_numeric(df_low_fertility['출산율(명)'])
```

#lineplot으로 시각화

```
plt.figure(figsize=(15, 5))  
sns.lineplot(x='연도', y='출산율(명)', data=df_low_fertility, marker='o', hue='국가')  
plt.yticks(np.arange(0.8, 2.2, 0.1))  
plt.title("대한민국, 일본, 이탈리아 출산율 추이")
```

```
plt.figure(figsize=(15, 5))  
sns.lineplot(x='연도', y='출산율(명)', data=df_high_fertility, marker='o', hue='국가')  
plt.yticks(np.arange(0.8, 2.2, 0.1))  
plt.title("대한민국, 프랑스, 독일 출산율 추이")
```



대한민국 고용률 데이터 시각화

1. 서브 플롯을 사용해서 데이터 한눈에 대략적으로 들어올 수 있게 시각화
2. 연도별 추이는 lineplot, 특정 연도의 데이터는 barplot을 활용해 시각화

#고용률 비교

```
hired_sub_plot = plt.figure(figsize=(40, 20))#서브플롯으로 보기 쉽게 시각화
```

```
ax1=hired_sub_plot.add_subplot(2,3,1)
ax2=hired_sub_plot.add_subplot(2,3,2)
ax3=hired_sub_plot.add_subplot(2,3,3)
ax4=hired_sub_plot.add_subplot(2,3,4)
ax5=hired_sub_plot.add_subplot(2,3,5)
ax6=hired_sub_plot.add_subplot(2,3,6)
```

```
ax1.set_title('2000년대 대한민국 고용률 추이')
sns.lineplot(x='연도', y='고용률(%)', data=df_hired[df_hired['국가'] == '대한민국'], marker='o', hue="성별구분", ax=ax1)
```

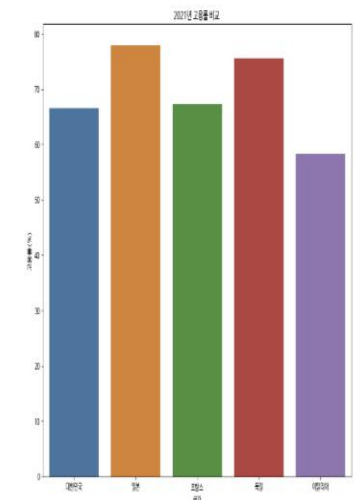
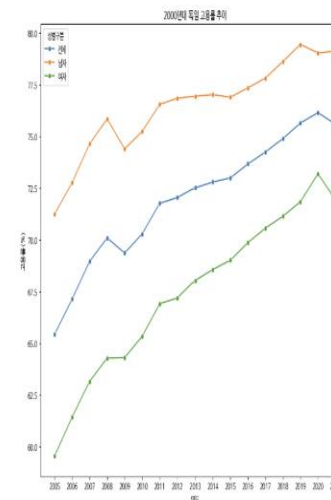
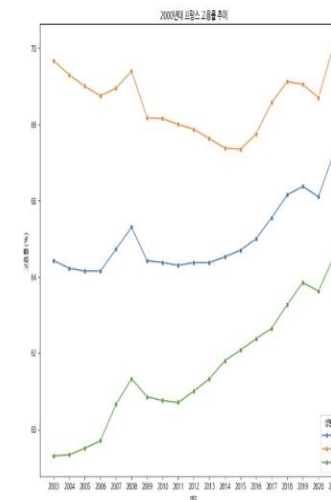
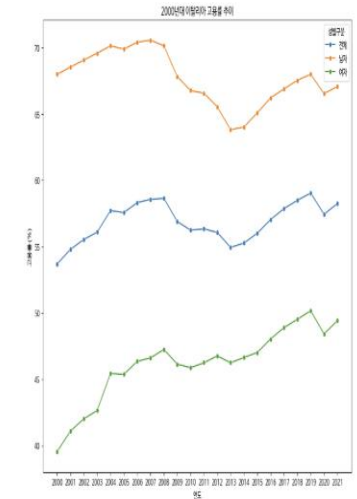
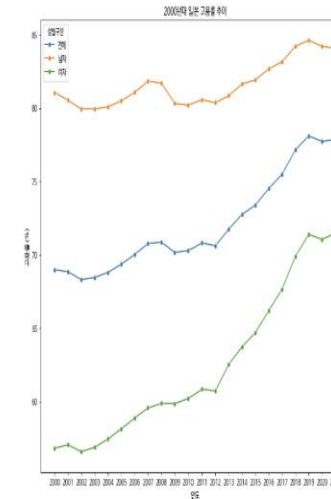
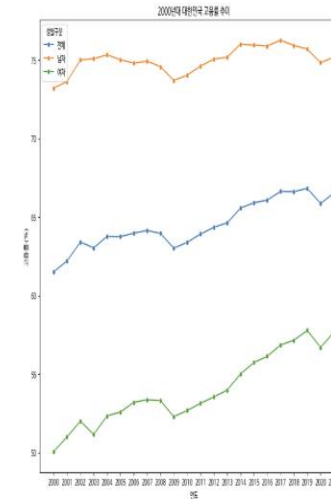
```
ax2.set_title('2000년대 일본 고용률 추이')
sns.lineplot(x='연도', y='고용률(%)', data=df_hired[df_hired['국가'] == '일본'], marker='o', hue="성별구분", ax=ax2)
```

```
ax3.set_title('2000년대 이탈리아 고용률 추이')
sns.lineplot(x='연도', y='고용률(%)', data=df_hired[df_hired['국가'] == '이탈리아'], marker='o', hue="성별구분", ax=ax3)
```

```
ax4.set_title('2000년대 프랑스 고용률 추이')
sns.lineplot(x='연도', y='고용률(%)', data=df_hired[df_hired['국가'] == '프랑스'], marker='o', hue="성별구분", ax=ax4)
```

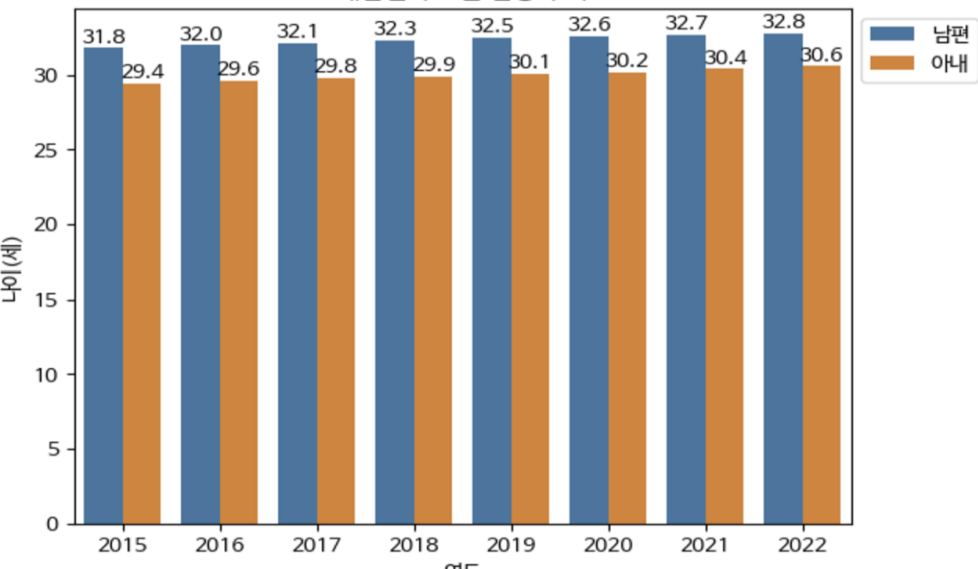
```
ax5.set_title('2000년대 독일 고용률 추이')
sns.lineplot(x='연도', y='고용률(%)', data=df_hired[df_hired['국가'] == '독일'], marker='o', hue="성별구분", ax=ax5)
```

```
ax6.set_title('2023년 고용률 비교')
sns.barplot(x='국가', y='고용률(%)', data=df_hired_total, hue='국가', ax=ax6)
```

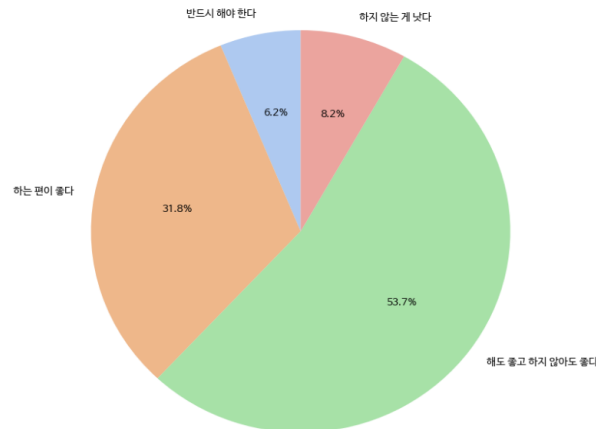


다양한 시각화 그래프

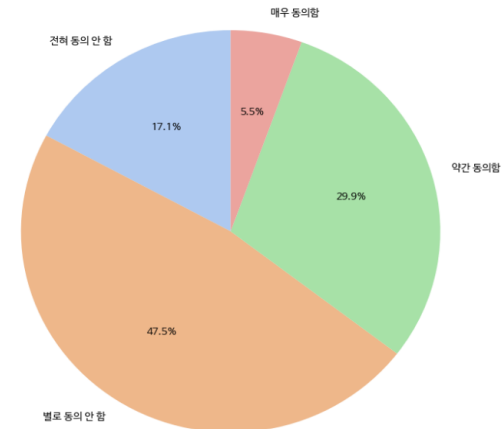
대한민국 초혼 연령 추이



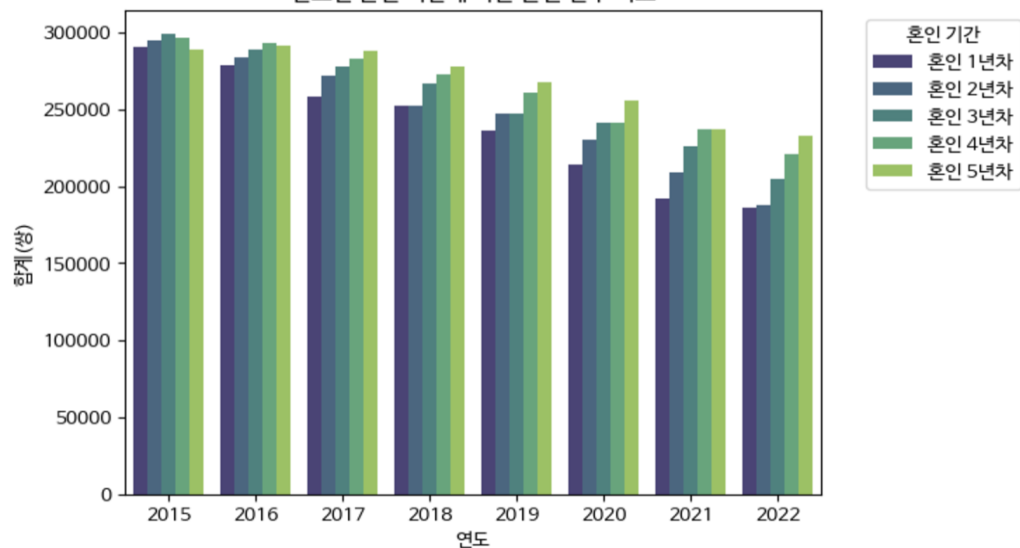
2022 비혼남녀(19~49) 결혼 발표영 건의 조사



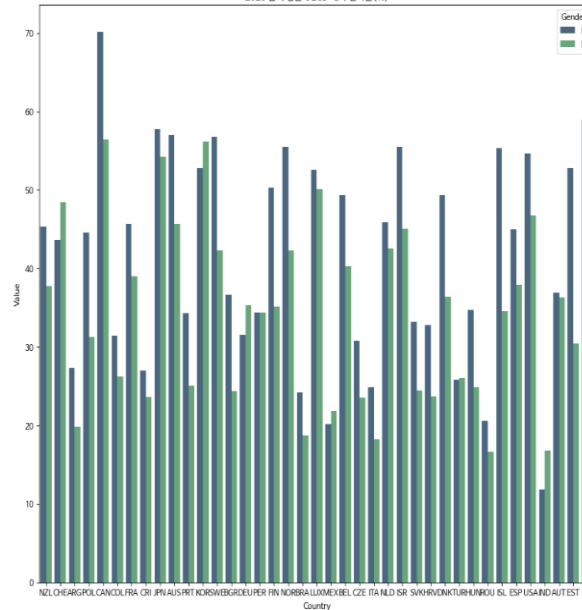
2021 비혼남녀(19~49) 결혼안 사법비 결혼아서 깊은 사법보너 땀쪽안기 건의 조사



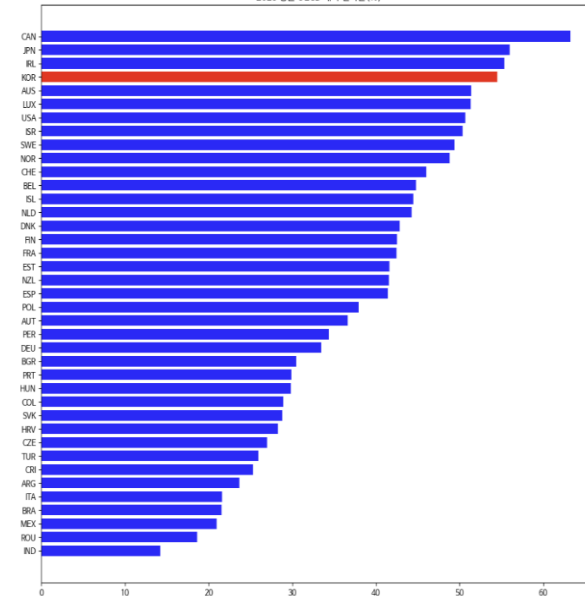
연도별 혼인 기간에 따른 혼인 건수 비교



2023 남녀 평균 OECD 대학 진학률(%)



2023 평균 OECD 대학 진학률(%)



데이터 분석

- 고용률 : 우리나라를 비롯해 고용률이 지속적으로 오르는 추세이지만 고용률과 출산율의 상관관계는 높아 보이지 않는다.
- 청년 실업률 : 프랑스의 경우만 보더라도 직접적인 이유라고 볼 수 없다
- GDP & 근무시간 : 전 세계적으로 봤을 높은 GDP를 가진 대한민국이지만 근무 시간당 GDP의 통계를 보면 높은 GDP가 의미가 있을까 싶다. 낮은 출산율을 기록하는 나라는 근무시간이 비교적 높다는 걸 알 수 있다. 꽤나 깊은 상관관계를 갖는 것으로 보인다. 대한민국은 2000년대에 지속적으로 감소하고 있긴 하지만 여전히 높은 근로시간 데이터를 보이고 있다
- 결혼 : 초혼연령의 경우 2015년부터 지속적으로 상승하여 2015년 대비 2022년 1살 더 증가했지만 OECD 국가들 중 중간 정도에 위치한다. 큰 문제라고 보이지 않는다. 조혼인율 또한 지속적으로 감소하지만 OECD 국가들 중 중간 정도에 위치한다. 이러한 이유는 결혼에 대한 인식조사 통계를 보면 알 수 있다. 결혼을 하기 싫다 보다 '굳이 결혼을 해야하나?' 라는 의견이 대다수임을 확인 할 수 있다. 또한 높은 대학 진학률 또한 여러 이유 중 하나가 된다고 본다.

데이터 분석

- 자녀 및 출산 : 기혼여성의 자녀인식 조사에 따르면 자녀가 있으면 좋겠다는 의견이 대다수임을 확인할 수 있다. 신혼부부 연차별 출생아 수를 보면 4년차 까지 0.9명으로 1명이 되지 않는다. 이 것이 의미하는 바는 자녀를 원하는 부부는 많지만 결혼 후 바로 아이를 갖는 것이 아니라 3~5년차에 아이를 갖는 것을 의미한다. 고로 늦은 나이에 아이를 갖다보니 둘째를 바라기 힘든 것이라 추정된다. 실제 OECD산 모 출산나이 통계를 보면 '평균출산나이', '평균 첫째 출산나이'의 부문 모두 대한민국이 월등히 가장 많은 나이를 보이고 있다.
- 가계지출 : 흔히 우리나라의 부동산 값을 큰 문제라고 지적한다. 하지만 OECD에서 제공하는 데이터를 보면 2015년 기준 주택가격지수를 보면 2023 대한민국의 주택가격 지수는 크지 않다. 물론 평균이다 보니 지방의 집 값까지 포함되므로 서울 집값의 높은 가격을 정확하게 반영할 수는 없을 것 같다. 그러나 OECD가계지출 데이터를 보면 우리나라의 가계지출은 다른나라에 비해 적은 비율을 차지한다. 오히려 교육에 관련한 비용이 압도적으로 높은 것을 확인할 수 있다.

결론

- 가장 큰 문제를 차지하는 것은 늦은 나이에 아이를 출산한다는 것 이다.
- 첫째 아이 자체를 늦게 출산하다 보니 둘째를 갖는 것에 나이에 대한 부담이 큰 것으로 보인다.
- 두번째로 아이를 낳기 위해서 결혼을 해야 하는데 높은 대학진학률, 사회적인식으로 인해 결혼하는 사람의 수도 줄고 결혼연령도 높아진다.
- 결혼 연령이 높아지니 앞서 말한 출산에 관해 나이에 대한 부담을 무시할 수 없을 것이다.
- 결혼을 하지 않는 이유를 볼 때 높은 근로시간이 큰 지분을 차지하는 것으로 생각된다.