

# 대한민국 인구 그래프

(2000 ~ 2022)



[kaggle.com/jeff125](https://kaggle.com/jeff125)

# 왜 인구 그래프를 그렸는가?

뉴스 또는 TV 방송 등에서 “**대한민국의 미래**”의 관한 이야기가 많이 나옵니다.

그 이야기의 내용의 핵심은 “**대한민국의 인구 자연 감소**” 및 “**전 세계 출산율 꼴찌**”라는 내용이 핵심이었습니다. 대한민국이 출산율이 꼴찌라는건 많이 들었지만, 어느 정도로 심각한지가 와 닿지 않았습니다. 그래서 **Kaggle**이라는 플랫폼에서 데이터를 찾아서 직접 그래프를 그려보며 그 위기를 직접 실감하게 되는 계기가 되었습니다.

# Kaggle이란?

Kaggle은 데이터 과학 및 머신 러닝 경진대회, 데이터 과학 문제, 데이터 정제 및 탐색 등을 위한 **온라인 플랫폼**입니다. 사용자들은 여기에서 데이터 과학자들과 경쟁하여 새로운 방법을 개발하거나, 오픈 소스 데이터세트를 사용하여 프로젝트를 수행할 수 있습니다.

# 데이터셋은 어디서?



Home



Competitions



Datasets




Code



Discussions



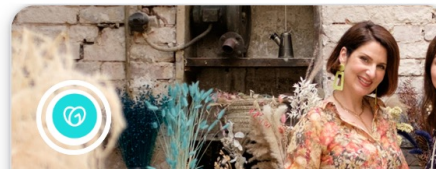
Learn



**1st and Future - Player Contact Detection**

Detect Player Contacts from Sensor and ...  
Featured  
Code Competition · 704 Teams

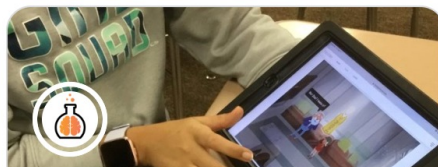
**\$100,000** 20d to go



**GoDaddy - Microbusiness Density Forecasting**

Forecast Next Month's Microbusiness Den...  
Featured  
2426 Teams

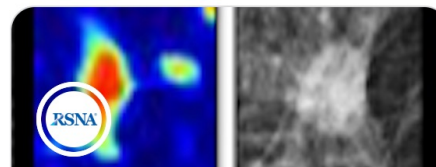
**\$60,000** 1mo to go



**Predict Student Performance from Game Play**

Trace student learning from Jo Wilder onli...  
Featured  
Code Competition · 174 Teams

**\$55,000** 3mo to go



**RSNA Screening Mammography Breast Canc...**

Find breast cancers in screening mammo...  
Featured  
Code Competition · 1506 Teams

**\$50,000** 18d to go

Competitions 또는 Datasets에서 데이터셋을 볼 수 있습니다.

Competitions에선 데이터의 이용약관 등 확인하고 하며 데이터를 활용하는 반면, Datasets에선 공개 데이터로 자신이 원하는 데이터가 있으면 찾아가 바로 데이터를 확인 할 수있는 차이가 있습니다.

# 데이터가공 전 짧게 확인

Korean\_demographics\_2000-2022.csv (305.13 kB)







Detail Compact Column

10 of 12 columns

## About this file

The second version include birth, death, natural growth, marriage, and divorce data in South Korea between January 2010 and June 2022.

The total numbers areThe second version include birth, death, natural growth, marriage, and divorce data in South Korea between January 2010 and June 2022.

Date	Region	# Birth	# Birth_rate	# Death
2010-2022	South Korea, Whole country and region	value	rate per 1000	# of death
 1Jan00 1Jun22	<b>18</b> unique values	 67 61.6k	 3.1 18.8	 52 44.5k
1/1/2000	Busan	3752	11.61	1875
1/1/2000	Chungcheongbuk-do	1903	15.06	924
1/1/2000	Chungcheongnam-do	2398	14.75	1466
1/1/2000	Daegu	3057	14.39	1117
1/1/2000	Daejeon	1859	16.08	565
1/1/2000	Gangwon-do	1966	14.91	1067
1/1/2000	Gwangju	2159	18.77	606

자신이 원하는 데이터를 찾으면 Excel  
파일의 데이터를 간단하게 (정렬, 검색 등)을  
이용해 확인 할 수 있습니다.

간단하게 확인을 했으면 우측 상단의  
New Notebook을 눌러 온라인으로 코딩 할  
수 있습니다.

New Notebook

Download (103 kB)



# 코드 리뷰

#데이터 정보 확인

```
print(data.shape) #행, 열 확인
print("-----")
print(data.dtypes) # 데이터들의 타입 확인
print("-----")
print(data.describe()) # 데이터들의 통계량 확인
```

가장 중요한 것이 전체적인 데이터 타입을 먼저 확인하는  
것 입니다. 그 이유는 데이터 가공시에 원하는 타입을  
만드려면 변환 과정이 있어야하는데 그 과정에서 데이터  
타입을 확인하는 과정이 오래걸리기 때문입니다.

(4860, 12) 열이 4860열, 행이 12행이라는 뜻 입니다.

```
-----
Date                object
Region              object
Birth               float64
Birth_rate          float64
Death               float64
Death_rate          float64
Divorce             float64
Divorce_rate        float64
Marriage            float64
Marriage_rate       float64
Natural_growth      float64
Natural_growth_rate float64
dtype: object
-----
```

데이터들의 타입을 확인 할 수 있습니다.

	Birth	Birth_rate	Death	Death_rate	Divorce \
count	4716.000000	4709.000000	4716.000000	4709.000000	4716.000000
mean	4138.169635	8.737872	2556.818066	5.857528	1130.374894
std	8450.112413	2.358128	5029.234791	1.589019	2287.594278
min	67.000000	3.100000	52.000000	3.100000	10.000000
25%	1004.750000	7.210000	692.500000	4.500000	278.000000
50%	1431.500000	8.830000	1149.000000	5.600000	392.000000
75%	2327.000000	10.070000	1721.000000	7.000000	651.250000
max	61644.000000	18.770000	44487.000000	15.700000	15517.000000

각 데이터들의 통계를 미리 확인 할 수 있습니다.

# 코드 리뷰

데이터의 실질적인 내용을 간단하게 확인 할 수 있습니다.

```
data.head()
```

**data.head()** 값에 원하는 숫자를 넣으면 그 만큼 행이 출력됩니다.

	Date	Region	Birth	Birth_rate	Death	Death_rate	Divorce	Divorce_rate	Marriage	Marriage_rate
0	1/1/2000	Busan	3752.0	11.61	1875.0	5.8	814.0	2.5	2435.0	7.5
1	1/1/2000	Chungcheongbuk-do	1903.0	15.06	924.0	7.3	220.0	1.7	828.0	6.6
2	1/1/2000	Chungcheongnam-do	2398.0	14.75	1466.0	9.0	321.0	2.0	1055.0	6.5
3	1/1/2000	Daegu	3057.0	14.39	1117.0	5.3	422.0	2.0	1577.0	7.4
4	1/1/2000	Daejeon	1859.0	16.08	565.0	4.9	280.0	2.4	868.0	7.5

# 코드 리뷰

```
data.Region.unique() Region의 행에서 나오는 모든 데이터들의 항목을 표시합니다. (중복 제거가 되서 나옴)
```

```
array(['Busan', 'Chungcheongbuk-do', 'Chungcheongnam-do', 'Daegu',  
      'Daejeon', 'Gangwon-do', 'Gwangju', 'Gyeonggi-do',  
      'Gyeongsangbuk-do', 'Gyeongsangnam-do', 'Incheon', 'Jeju',  
      'Jeollabuk-do', 'Jeollanam-do', 'Seoul', 'Ulsan', 'Whole country',  
      'Sejong'], dtype=object)
```

```
Whole_country = data[data.Region == 'Whole country']
```

**Region**의 행에서 나오는 모든 데이터들은 `Whole_country`라는 변수에 넣어졌습니다.

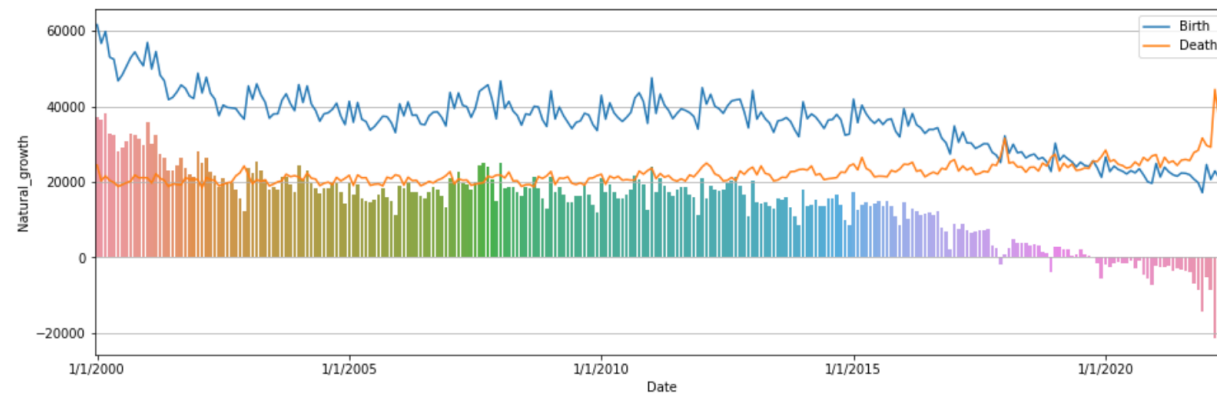


# 코드 리뷰

## Whole Country Birth and Death

전체적인 그래프는 출생아, 사망자를 기준으로 하였으며,  
대한민국의 실질적인 인구 성장을 막대 그래프로 표기했습니다.

```
plt.figure(figsize=(16,5))
plt.style.use('default')
plt.ticklabel_format(useOffset=False, style='plain')
sns.lineplot(data=Whole_country, x='Date', y='Birth', label='Birth')
sns.lineplot(data=Whole_country, x='Date', y='Death', label='Death')
sns.barplot(data=Whole_country, x='Date', y='Natural_growth')
plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005', '1/1/2010', '1/1/2015', '1/1/2020'])
plt.grid(True, axis='y')
plt.show()
```



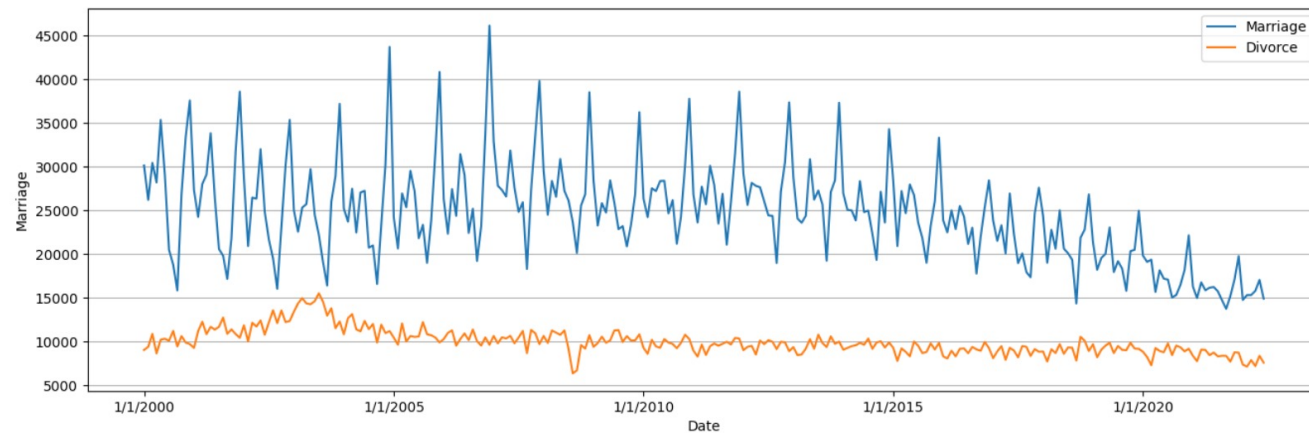
# 코드 리뷰

## Whole Country Marriage and Divorce

이 그래프는 결혼, 이혼의 관한 데이터를 그래프로

결혼과 이혼으로 인해 인구 변화가 있었는지 확인해보고 싶었습니다.

```
plt.figure(figsize=(16,5))
plt.ticklabel_format(useOffset=False, style='plain')
sns.lineplot(data=Whole_country, x='Date', y='Marriage', label='Marriage')
sns.lineplot(data=Whole_country, x='Date', y='Divorce', label='Divorce')
plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005', '1/1/2010', '1/1/2015', '1/1/2020'])
plt.grid(True, axis='y')
plt.show()
```



# 코드 리뷰

## Whole Country Birth\_rate

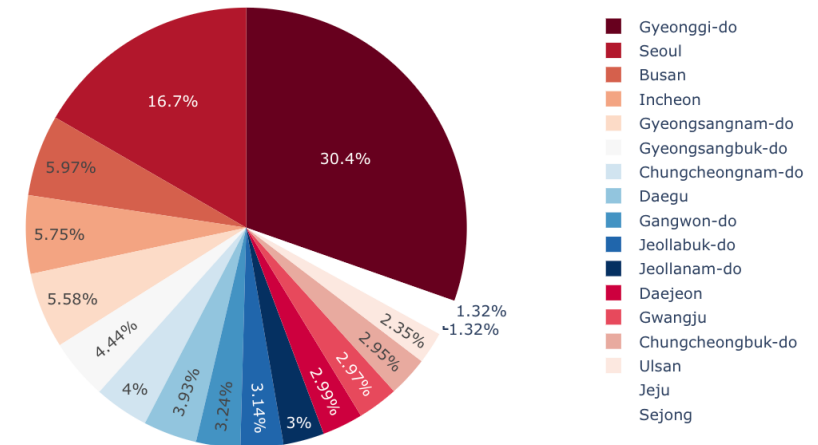
```
import plotly.express as px

filter_date = data.loc[data['Date'] == '6/1/2022']
filter_Birth = filter_date.Birth[: -1]
filter_region = filter_date.Region[: -1]

fig = px.pie(values=filter_Birth,
             names=filter_region,
             color_discrete_sequence=px.colors.sequential.RdBu,
             title= '6/1/2022 Regional Birth Rate Pie Chart ')

fig.show()
```

6/1/2022 Regional Birth Rate Pie Chart



이 그래프는 전국 총 출생아 수의 지역별 원형차트입니다.

지방 소멸이 어느 정도로 심각한지를 확인 할 수 있었습니다.

# 코드 리뷰

## Comparison Seoul and Whole Country

```
plt.figure(figsize=(16,7))

plt.subplot(2, 1, 1)
plt.ticklabel_format(useOffset=False, style='plain')

sns.lineplot(data=Whole_country, x='Date', y='Birth_rate', label='Whole_country_Birth_rate')
sns.lineplot(data=data[data.Region == 'Seoul'], x='Date', y='Birth_rate', label='Seoul_Birth_rate')
plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005', '1/1/2010', '1/1/2015', '1/1/2020'])
plt.grid(True, axis='y')

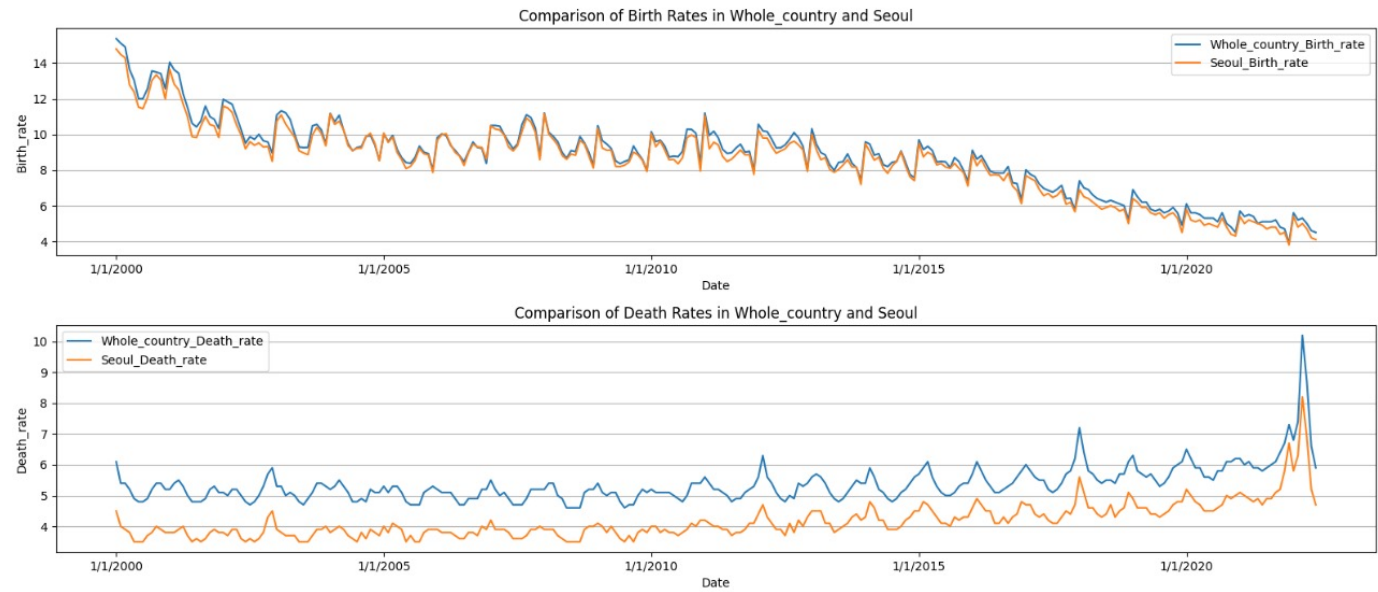
plt.title('Comparison of Birth Rates in Whole_country and Seoul')
plt.ylabel('Birth_rate')

plt.subplot(2, 1, 2)

sns.lineplot(data=Whole_country, x='Date', y='Death_rate', label='Whole_country_Death_rate')
sns.lineplot(data=data[data.Region == 'Seoul'], x='Date', y='Death_rate', label='Seoul_Death_rate')
plt.title('Comparison of Death Rates in Whole_country and Seoul')
plt.ylabel('Death_rate')

plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005', '1/1/2010', '1/1/2015', '1/1/2020'])
plt.grid(True, axis='y')

plt.tight_layout()
plt.show()
```



전국과 서울의 출산율, 사망률의 관한 비교그래프입니다.

서울에 인구 절반이 모여있어, 전국과 비교했을 때 큰 차이가 없었습니다.

# 코드 리뷰

## Comparison Jeollabuk Province and Whole Country

```
plt.figure(figsize=(16,7))

plt.subplot(2, 1, 1)
plt.ticklabel_format(useOffset=False, style='plain')

sns.lineplot(data=Whole_country, x='Date', y='Birth_rate', label='Whole_country_Birth_rate')
sns.lineplot(data=data[data.Region == 'Jeollabuk-do'], x='Date', y='Birth_rate', label='Jeollabuk Province_Birth_rate')
plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005', '1/1/2010', '1/1/2015', '1/1/2020'])
plt.grid(True, axis='y')

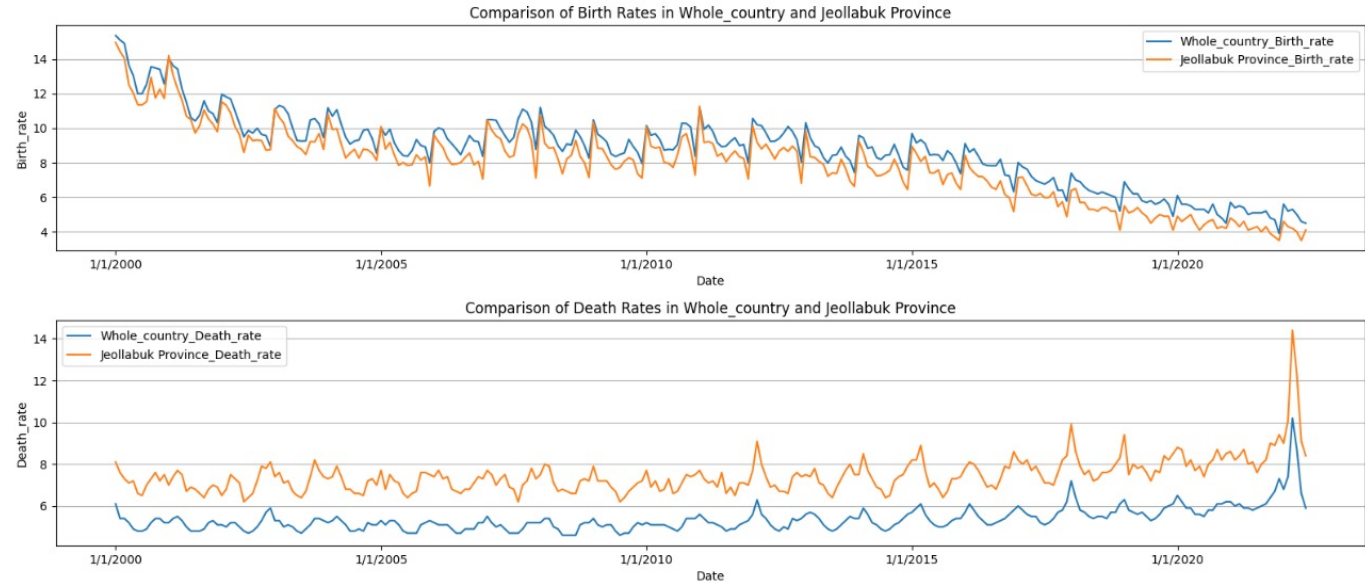
plt.title('Comparison of Birth Rates in Whole_country and Jeollabuk Province')
plt.ylabel('Birth_rate')

plt.subplot(2, 1, 2)

sns.lineplot(data=Whole_country, x='Date', y='Death_rate', label='Whole_country_Death_rate')
sns.lineplot(data=data[data.Region == 'Jeollabuk-do'], x='Date', y='Death_rate', label='Jeollabuk Province_Death_rate')
plt.title('Comparison of Death Rates in Whole_country and Jeollabuk Province')
plt.ylabel('Death_rate')

plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005', '1/1/2010', '1/1/2015', '1/1/2020'])
plt.grid(True, axis='y')

plt.tight_layout()
plt.show()
```



전국과 전북의 출산율, 사망률의 관한 비교그래프입니다.

가장 인구소멸이 심각한 지역이었습니다.

# 코드 리뷰

## Comparison Sejong and Whole Country

```
plt.figure(figsize=(16,7))

plt.subplot(2, 1, 1)
plt.ticklabel_format(useOffset=False, style='plain')

sns.lineplot(data=Whole_country, x='Date', y='Birth_rate' ,label='Whole_country_Birth_rate')
sns.lineplot(data=data[data.Region == 'Sejong'], x='Date', y='Birth_rate' ,label='Sejong_Birth_rate')
plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005','1/1/2010', '1/1/2015','1/1/2020'])
plt.grid(True, axis='y')

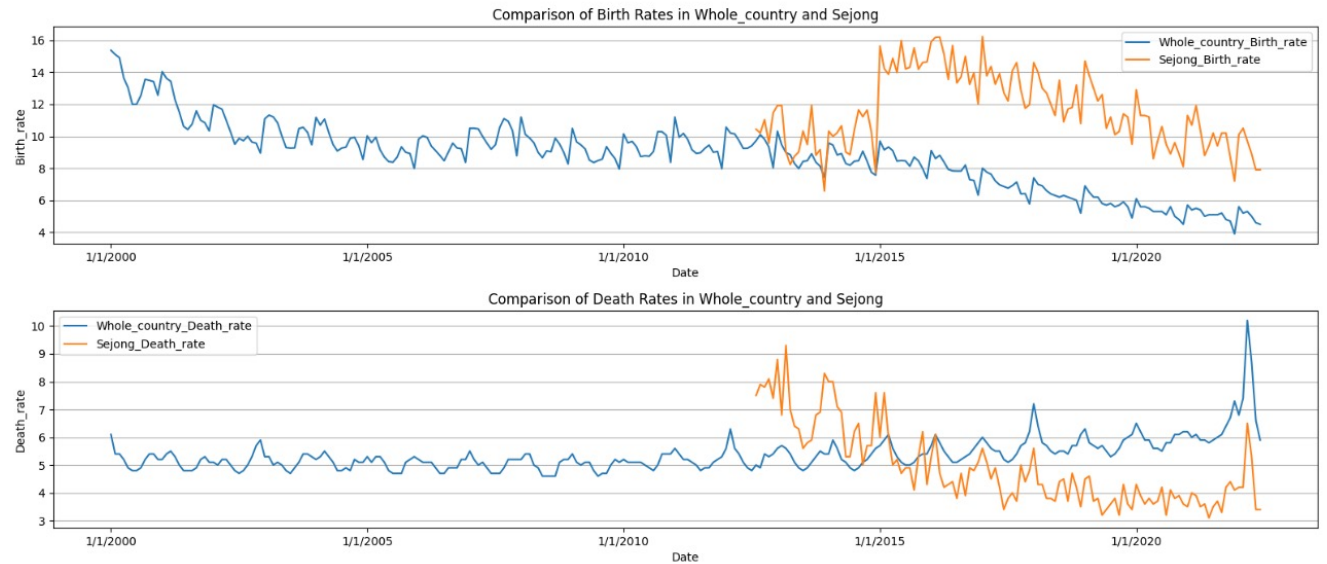
plt.title('Comparison of Birth Rates in Whole_country and Sejong')
plt.ylabel('Birth_rate')

plt.subplot(2, 1, 2)

sns.lineplot(data=Whole_country, x='Date', y='Death_rate', label='Whole_country_Death_rate')
sns.lineplot(data=data[data.Region == 'Sejong'], x='Date', y='Death_rate', label='Sejong_Death_rate')
plt.title('Comparison of Death Rates in Whole_country and Sejong')
plt.ylabel('Death_rate')

plt.xticks(rotation=0, size=10)
plt.xticks(['1/1/2000', '1/1/2005','1/1/2010', '1/1/2015','1/1/2020'])
plt.grid(True, axis='y')

plt.tight_layout()
plt.show()
```



전국과 세종의 출산율, 사망률의 관한 비교그래프입니다.

2012년 대한민국에서 새로 생긴 특별자치시 이기에 2012년 이전의 데이터가 없습니다.

그리고 현재 대한민국에서 가장 높은 출산율을 기록하고있으며, 사망률도 전국 최하위입니다.

# 코드 피드백



**Mayuresh Koli**

Posted 5 months ago · Posted on Version 3 of 4

Hello There @jeff125 ,

안녕 @jeff125

You have done a great job.

수고했어

But your comparison sections of the notebook can be better, If you use functions that will take one region and output those two plots.

하지만 한 지역을 선택하여 두개의 그래프를 출력하는 함수를 사용하면 비교 섹션이 더 나을 수도 있어

Here you have repeated the code for each region, so try to make a function so that it will be easier to write one line of code instead of many. 여기서 각 지역에 대한 코드를 많이 반복했지만, 대신에 한 줄의 코드를 쉽게 작성할 수 있도록 함수를 써서 해봐



**Alex**

Posted 5 months ago · Posted on Version 3 of 4

Great notebook @jeff125

잘했어 @jeff125

I like the pie chart to represent the share of the total birth for each region.

나는 원형차트의 각 지역의 전체 출생율을 나타내는 것이 마음에 들어.

# 느낀점

데이터가공하면서 정말 대한민국의 출산율의 참담함을 실감했습니다.

또한 데이터가공을 끝내고 피드백이 왔을 때 누군가가 내 코드를 보고 피드백을 해준다는 것이 정말 좋았습니다. 또한 “함수를 써서 해보는게 좋을것 같다.” 는 피드백을 보고 왜 이런 생각을 하지 못했는지에 아쉬운 마음도 들었습니다. 하지만 누군가가 내 코드를 보고 리뷰해준다는 것이 정말 그 기분은 말로 표현하기가 힘들정도로 좋았습니다.