올림픽 성적의 성공 요인

3팀 류지헌 지현우 고현욱 박소영 김민찬

목차

올림픽 성적은 무엇과 연관 있는가?

대한민국의 역대 올림픽 성적 순위

하계 올림픽			동계 올림픽		
연도	개최지	대한민국 순위	연도	개최지	대한민국 순위
1988	서울	4위	2010	벤쿠버	5위
2012	런던	5위	1994	릴레함메르	6위
2008	베이징	7위	2018	평창	7위
1992	바르셀로나	7위	2006	토리노	7위

- 개최국일때 높은 성적을 보임.
- 올림픽 성적에 영향을 미치는 것이 무엇일지 의문점 제기

연관성이 높은 요인

운동 재정 등　　날씨　　유전

영향력이 있는 많은 요소들 중에서
큰 영향력이 느껴지는

“ 개최국 ”　“ GDP ”　“ 인구수, GDP ”

단위로 채택하여 분석

분석 과정 | 데이터 수집 과정

단위	데이터 세트
“ 개최국 ”	✔ 120년의 올림픽 선수 데이터 세트 ⇒ 4개의 연속형 변수, 11개의 범주형 변수 ✔ 올림픽 개최국, 해당 개최국의 IOC 국가 코드 정보 수집
“ GDP ” “ 인구수, GDP ”	✔ 국가별 코드, 인구 수, GDP, 메달 수 ⇒ 3개의 연속형 변수, 10개의 범주형 변수

분석 과정 | 데이터 수집 과정

120년의 올림픽 선수 데이터 세트

연속형 변수	
변수 이름	**변수 설명**
Age	선수의 나이
Height	선수의 신장
Weight	선수의 체중
Year	올림픽 개최년도

범주형 변수	
변수 이름	**변수 설명**
ID	선수 식별자
Name	선수의 이름
Sex	선수의 성별
Games	개최년도 + 동하계 구분
Team	국가별 팀
NOC	국가 코드
Season	동하계 구분
City	개최지
Sport	종목
Event	세부종목
Medal	메달 종류+ 미달성

분석 과정 | 데이터 수집 과정

국가별 코드, 인구 수, **GDP**, 메달 수

연속형 변수	
변수 이름	**변수 설명**
Population	인구 수
GDP per Capita	1인당 GDP
Year	올림픽 개최년도

범주형 변수	
변수 이름	**변수 설명**
Country	국가 이름
Code	국가 코드
City	개최지
Sport	종목
Discipline	세부 정보
Country	동하계 데이터에서의 국가 코드
Gender	성별
Event	세부 종목
Medal	메달 종류
Athlete	선수 이름

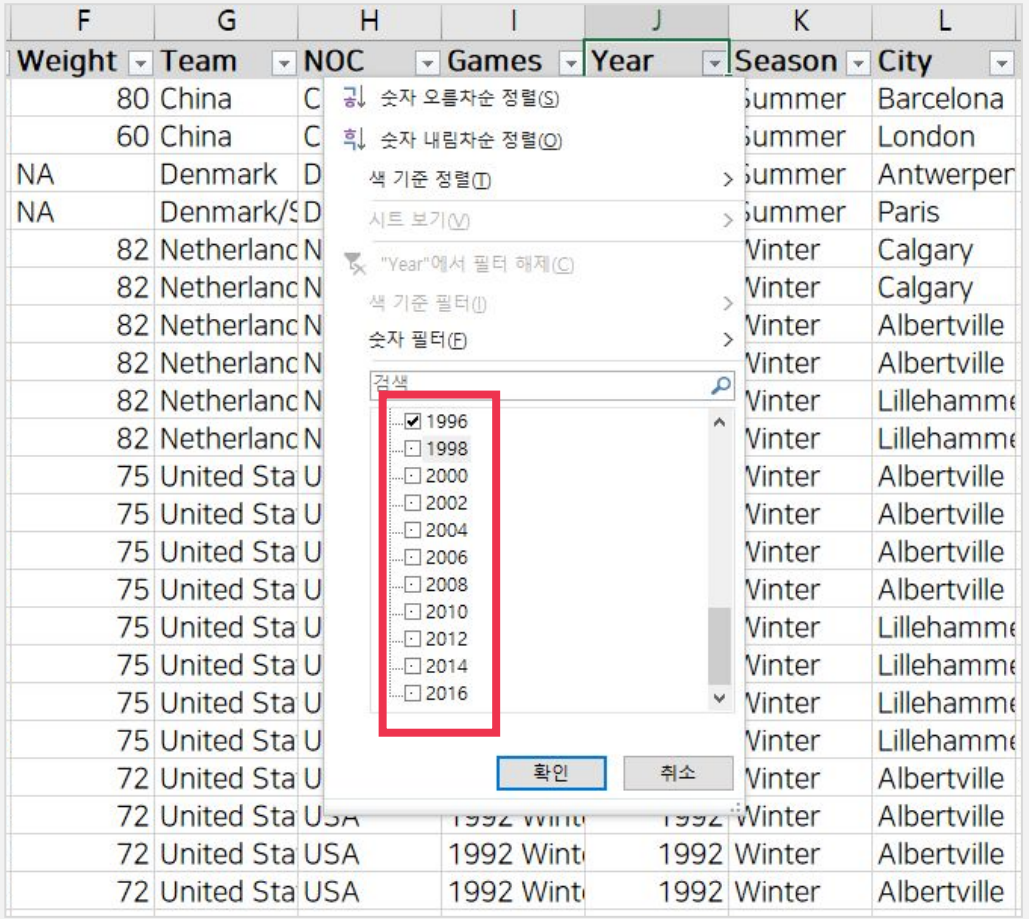
분석 과정 | 데이터 전처리 “개최국”

01 엑셀로 올림픽 선수 데이터 세트의

1998~2016년도 개최년도 이외 데이터 드롭하여

최신 데이터 추출

총 데이터 271,1146개 → 88,863개



분석 과정 | 데이터 전처리 "개최국"

02 위키백과 정보를 바탕으로

1998~2016년도 올림픽 개최국과 NOC의 데이터세트 생성

03 올림픽 선수 데이터 세트의

'Year' **변수의 Value와 일치하는**

새로운 'Host' 변수 생성하고 병합

Summer	개최지	개최국	NOC
2016	리우	브라질	BRA
2012	런던	영국	GBR
2008	베이징	중국	CHN
2004	아테네	그리스	GRE
2000	시드니	오스트레일리아	AUS

Winter	개최지	개최국	NOC
2014	소치	러시아	RUS
2010	밴쿠버	캐나다	CAN
2006	토리노	이탈리아	ITA
2002	솔트레이크시티	미국	USA
1998	나가노	일본	JPN

H	I	J	K	L	
NOC	Host	Games	Year	Season	
CHN	=IF(K2=2016, "BRA",		2012	Summer	
FIN	IF(K2=2014, "RUS",		2002	Winter	
FIN	IF(K2=2012, "GBR",		2000	Summer	
FIN	IF(K2=2010, "CAN",		2000	Summer	Sydney
FIN	IF(K2=2008, "CHN",		2014	Winter	Sochi
FIN	IF(K2=2006, "ITA",		2000	Summer	Sydney
NOR	IF(K2=2004, "GRE",		1998	Winter	Nagano
NOR	IF(K2=2002, "USA",		1998	Winter	Nagano
NOR	IF(K2=2000, "AUS",		1998	Winter	Nagano
NOR	IF(K2=1998, "JPN",		1998	Winter	Nagano
NOR	))))))))))		2002	Winter	Salt Lak
NOR			2002	Winter	Salt Lak
NOR	U IF(logical_test, [value_if_true], [value_if_false])			Winter	Salt Lak
NOR	USA	2002 Winter	2002	Winter	Salt Lak
NOR	USA	2002 Winter	2002	Winter	Salt Lak
NOR	ITA	2006 Winter	2006	Winter	Torino
NOR	ITA	2006 Winter	2006	Winter	Torino
NOR	CHN	2008 Summer	2008	Summer	Beijing
ROU	BRA	2016 Summer	2016	Summer	Rio de J

분석 과정 | 데이터 전처리 “개최국”

04 선수의 메달 성적이 없다면 : 숫자 **0** 변환

선수의 메달 성적이 있다면 : 숫자 **1** 변환

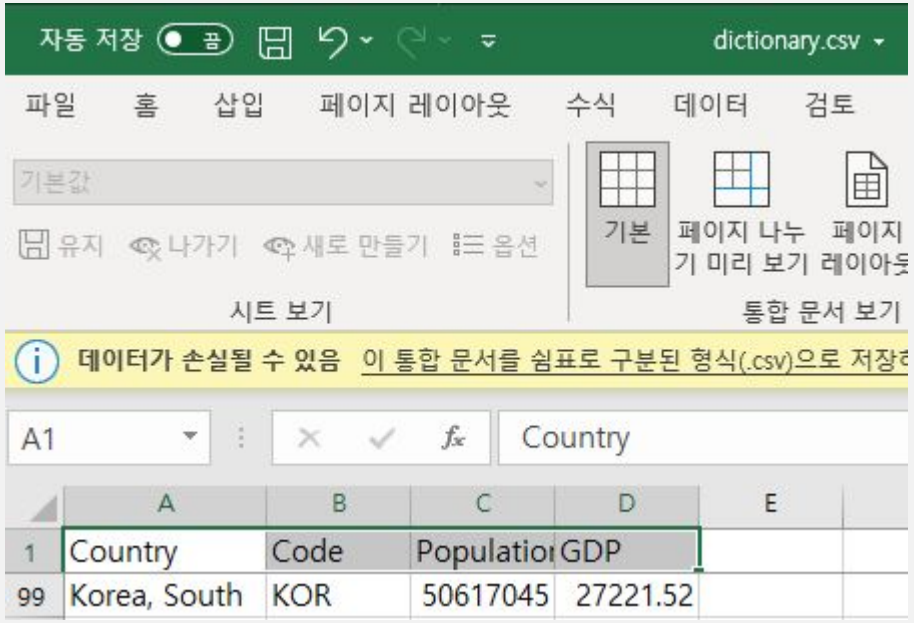
명목 척도화 된

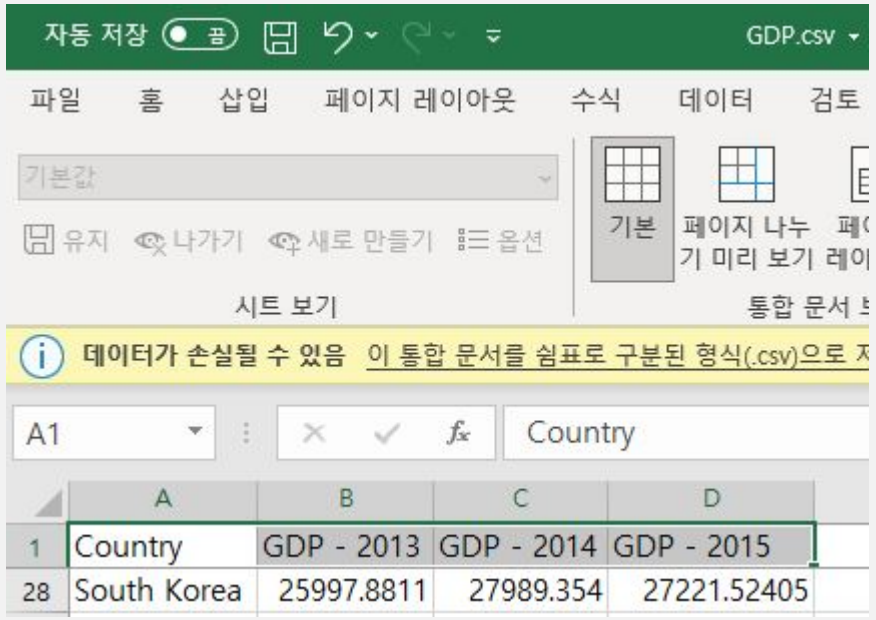
새로운 ‘Medal_bi’ 변수 생성

O	P	Q	R	S	T
Event	Medal	Medal_bi			
Judo Men'	NA	=IF(P2="Gold",1,IF(P2="Silver",1,IF(P2			
Ice Hockey	NA	="Bronze",1,0)))			
Badmintor	NA				
Sailing Wo	NA	IF(logical_test, [value_if_true], [value_if_false])			
Ice Hockey	Bronze	1			
Athletics M	NA	0			
Alpine Skii	NA	0			
Alpine Skii	NA	0			
Alpine Skii	NA	0			
Alpine Skii	NA	0			
Alpine Skii	NA	0			
Alpine Skii	Gold	1			
Alpine Skii	NA	0			
Alpine Skii	NA	0			
Alpine Skii	Gold	1			

분석 과정 | 데이터 전처리 “ 인구수, GDP ”

02 dictionary.csv의 2015년도 국가별 1인당 GDP 평균을 타 자료와 교차 확인하기 위해 GDP.csv파일 확인





분석 과정 | 데이터 전처리 “ 인구수, GDP ”

01 파이썬으로 **1896~2014**년도 올림픽 데이터 중 ‘국가’ , ‘메달수’ 변수만 추출 후 **2015**년도 국가별 **1**인당 **GDP** 평균이 포함된 **dict** 데이터와 병합

```python
import pandas as pd

# 데이터 읽기, 및 데이터 확인
summer_data = pd.read_csv("summer.csv")
winter_data = pd.read_csv("winter.csv")
dic_data = pd.read_csv("dictionary.csv")

# 하계 동계 올림픽 데이터 병합 코드
frame = [summer_data, winter_data]
data = pd.concat(frame)
#print(data.columns)
#print(data)

series = data.groupby(['Country']).Medal.count() # 국가별 메달 수 추출
df = pd.DataFrame({'country':series.index, 'medals':series.values})
#print(df)
#print(type(df))

print()
final_df = pd.merge(df, dic_data, left_on='country', right_on='Code').drop(['Code', 'Country'], axis=1)

# Nan행 제거
final_df = final_df.dropna(how='any')

print(final_df)

# csv 파일로 저장
final_df.to_csv('OlympicDataset.csv', sep=',')
```

분석 과정 | 분석 모델 "개최국"

독립변수	종속변수	분석방법
범주형	범주형	카이제곱 검정
범주형	연속형	T검정
연속형	범주형	로지스틱 회귀분석
연속형	연속형	회귀분석, 구조 방정식

카이 제곱검정 채택

- ✔ 두 개 이상의 변인을 사용하는 이원 카이 제곱 검정 방법을 사용하여 두 변인 사이의 독립성 검정 시행
- ✔ 독립변수 : 개최국 여부
  종속변수 : 선수의 메달 획득 여부
- ✔ 개최국과 메달 획득에는 어떤 관계가 있는지 두 변인의 독립성 판단

분석 과정 | 분석 모델 "개최국" 카이 제곱

카이 제곱 검정 코드

```python
import pandas as pd
import numpy as np
import scipy.stats as stats

# 이원 카이제곱으로 데이터 분석해보기

data = pd.read_csv('team3_recentTenOlympic.csv')
print(data.head(4))

# 일단 필요없는 데이터 지우기 (drop 사용)
df = data.drop(['Sex', 'Name', 'Age', 'Height', 'Weight', 'Sport', 'Event', 'Season', 'Year'], axis=1)
print(df.head())  # type 은 dataframe

print()
print("_____"*10)
print()

# 데이터 정리하기:
df['Country_bi'] = np.where(df["NOC"] == df["Host"], 1, 0)
# 개최국일 때 1, 아닐때 0
```

```python
# 귀무가설 : 개최국인것과 메달 획득은 관계가 없다.
# 대립가설 : 개최국인것과 메달 획득은 관계가 있다.

print()
print("_____"*10)
print()

# 빈도표
ctab = pd.crosstab(index=df['Country_bi'], columns=df['Medal_bi'])
print(ctab)

print()

chi2, p, ddof, expected = stats.chi2_contingency(ctab)
print('chi2:', chi2)  # 55.63413944379663
print('p:', p)  # 8.729439456355027e-14
print('ddof:', ddof)  # 1 : (2-1) * (2-1)
```

카이 제곱 검정 결과

```
Medal_bi         0      1
Country_bi
0            72346  11724
1             3939    854

chi2: 55.63413944379663
p: 8.729439456355027e-14
ddof: 1
```

- ✔ p-value가 8.7294e-14 < 0.05 유의미한 수준
  귀무가설 기각, 대립가설 채택
- ✔ 따라서 개최국인 것과 메달획득은 관계가 있다.

분석 과정 | 분석 모델 “ GDP ”

독립변수	종속변수	분석방법
범주형	범주형	카이제곱 검정
범주형	연속형	T검정
연속형	범주형	로지스틱 회귀분석
연속형	연속형	회귀분석, 구조 방정식

단순 회귀 분석모델 채택

- ✔ 독립변수가 하나일 경우에 종속변수와의 관계를 분석하여 독립변수가 종속변수에 미치는 영향을 분석하는 단순 회귀 분석 시행
- ✔ 독립변수 : **GDP**
  종속변수 : 메달 획득 수
- ✔ **GDP**라는 변수가 메달 획득 수에 어떤 영향을 미치는지 분석

분석 과정 | 분석 모델 "GDP" 단순 회귀 분석

변수 간 상관관계 분석. 상관계수 이용

```
15 df = pd.read_csv('OlympicDataset.csv').drop(['Unnamed: 0'], axis=1)
16 print(df.head(3))
17
18 # ____________________________________________________________
19 prettyLine()
20 # ____________________________________________________________
21
22 print(df.columns)  # ['country', 'medals', 'Population', 'GDP']
23 print(np.corrcoef(df.medals, df.GDP))  # 0.44026533
24 # 상관계수 판단
25 print(df.corr())
26
27 """
28                medals  Population       GDP
29 medals       1.000000    0.205056  0.440265
30 Population   0.205056    1.000000 -0.090703
31 GDP          0.440265   -0.090703  1.000000
32 """
```

- ✔ 변수들 간 상관관계를 분석
- ✔ 메달획득과 GDP간의 상관계수 : 약 0.44
  → 적당한 양의 상관관계

모델에 대한 통계

```
57 # 단순 선형 회귀 분석
58 import statsmodels.formula.api as smf
59 model = smf.ols('medals ~ GDP', data=df)
60 result = model.fit()
61 print(result.summary())
62 # R-squared:                       0.194
63 # Prob (F-statistic):           6.07e-07    --------> 유의한 모델
64 # slope = 0.0135    :     bias(intercept) = 25.6867
65 print('결정계수(설명력): ', result.rsquared)
66 # 결정계수(설명력):  0.19383355740739927 -----> 설명력이 적다.
```

- ✔ p-value : 6.07e-07로 0.05보다 적으므로 유의한 모델
- ✔ 결정계수(설명력) : 약 0.193
- ✔ 모델의 회귀선은 종속변수를 약 19.3%정도 설명한다

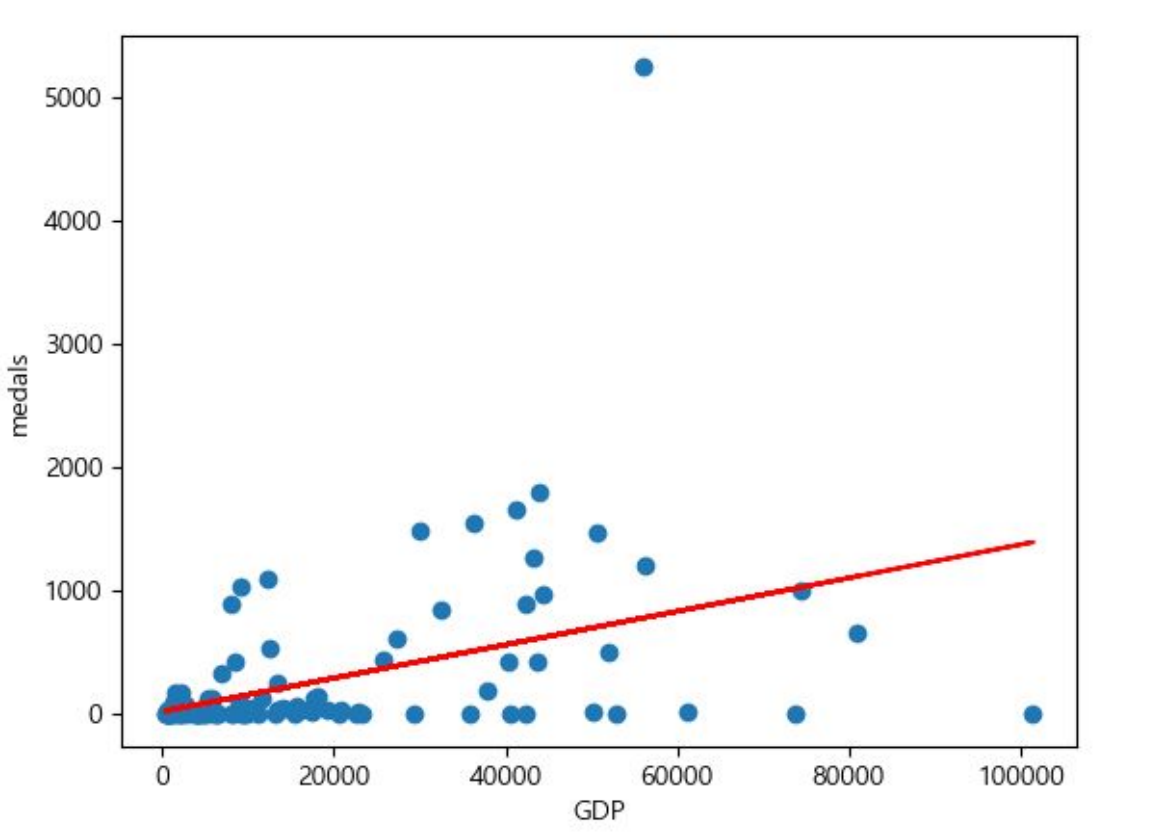
분석 과정 | 분석 모델 “GDP” 단순 회귀 분석

분석 모델의 시각화

```
68 # 시각화
69 # 실제 값으로 산포도 표시
70 plt.scatter(df.GDP, df.medals)
71 # 회귀식을 화면에 표시
72 plt.plot(df.GDP, 0.0135 * df.GDP + 25.6867, 'r')  # Wx + B -> 기울기 * x + intercept / 예측 값으로 산포도 표시
73 plt.xlabel('GDP')
74 plt.ylabel('medals')
75 plt.show()
```

✔ 실제 값의 산포도와 회귀선 표시

시각화 결과 ->



분석 과정 | 분석 모델 “ GDP ” 단순 회귀

모델로 예측 값 얻기

```
81 # 예측 준비 완료:
82 df.GDP = float(input('GDP를 입력하세요: '))
83 pred = result.predict(pd.DataFrame({'GDP':df.GDP}))
84 print('예상 총 메달 개수는:', int(pred[0]), ' 입니다.')
85 print(df.GDP)
```

작성한 모델을 이용해 키보드에서 새로운 값을 받아 예측 값을 얻는다.

입력으로 예측 값 얻기

```
GDP를 입력하세요: 22722
예상 총 메달 개수는: 333  입니다.
```

분석 과정 | 분석 모델 “ GDP ” 적절성 확인

모델의 적절성 확인

- ✔ **5가지 조건 : 정규성, 독립성, 선형성, 등분산성, 다중공선성**
- ✔ **모델의 적절성을 판단하여 각각의 조건을 위배 시에는 변수의 제거나 조정을 신중히 고려해야 한다.**

< 잔차항 >

- ✔ 잔차항 : 실제 값에서 예측 값을 뺀 값
  실제 메달의 개수 - 메달 개수의 예측 값 = 잔차항 계산

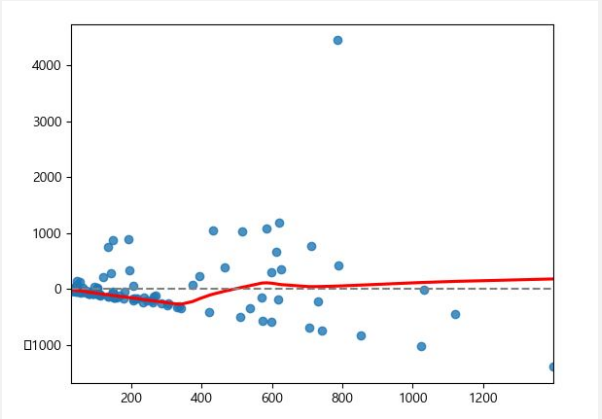
```
95 # 잔차항 (실제값 - 예측값) 구하기 (difference)
96 fitted = result.predict(df)   #  예측값
97 residual = df['medals'] - fitted   #  실제값 - 예측값
```

< 선형성 >

- ✔ 선형성 : 예측 값과 잔차가 비슷한 패턴을 가지는 것
  잔차와 예측값을 시각화 하여 이를 확인

```
 99 print('선형성 : 예측값과 잔차가 비슷한 패턴을 가짐')
100 sns.regplot(fitted, residual, line_kws={'color':'red'}, lowess=True) # regplot(예측값, 잔차값)
101 plt.plot([fitted.min(), fitted.max()], [0, 0], '--', color='grey')
102 plt.show()  # 완벽한 직선이 아니라서... 선형성을 완전하게 만족하지는 못함
```

잔차의 추세선과 예측값이 비슷한 패턴을 가지지 않아
선형성을 완전히 만족시키지는 못한다.



분석 과정 | 분석 모델 “ GDP ” 적절성 확인

< 정규성 >

✔ 정규성 : 잔차가 정규분포를 따라야 함
shapiro test를 사용하여 확인

```
110 print('정규성 : 잔차가 정규분포를 따라야함.')
111 import scipy.stats as stats
112 print('shapiro test: ', stats.shapiro(residual))
113 # pvalue=2.010499007105984e-16 < 0.05 정규성을 만족 못함
```

✔ p-value : 2.0104e-16 < 0.05 이므로
정규성을 만족하지 못함

< 독립성 >

✔ 독립성 : 잔차가 독립적, 자기상관이 없어야 함

```
print('독립성 : 잔차가 독립적, 자기상관(인접 관측치와 오차가 상관되어있음) 이 없어야 함')
print('더빈왓슨 값으로 확인: Durbin-Watson: 0.688')
# 더빈왓슨 값으로 확인: Durbin-Watson:                      0.688
# (0에 가까우면 양의 상관, 4에 가까우면 음의 상관. 2에 가까우면 자기상관이 없다)
# 그러므로 양의 상관관계이지만, 독립성이 부족함
```

```
==============================
Durbin-Watson:                   0.688
```

✔ 모델의 소계를 나타내는 .summary() 에서의
더빈-왓슨 값 : 0.688
이것이 0에 가까워 양의 상관관계가 존재하지만,
2에 가깝진 않으므로 자기상관이 존재할 수 있다.
따라서 해당 모델은 독립성이 부족하다.

◆ 등분산성과 다중공선성은
독립변수가 두개 이상일때 확인하므로 생략한다.

분석 과정 | 분석 모델 “ 인구 수, GDP ”

독립변수	종속변수	분석방법
범주형	범주형	카이제곱 검정
범주형	연속형	T검정
연속형	범주형	로지스틱 회귀분석
연속형	연속형	회귀분석, 구조 방정식

다중 회귀 채택

- ✔ 독립변수와 종속변수는 연속형 변수이나, 독립변수가 둘 이상인 회귀 분석
- ✔ 독립변수 : 인구 수, **GDP**
  종속변수 : 메달 획득 수
- ✔ 인구 수, **GDP**라는 변수가
  메달 획득 수에 어떤 영향을 미치는지 분석

분석 과정 | 분석 모델 "인구 수, GDP" 다중 회귀

다중 회귀 분석 모델 작성

```
53 # 다중 선형 회귀 분석
54 import statsmodels.formula.api as smf
55 model = smf.ols('medals ~ GDP + Population', data=df)
56 result = model.fit()
57 print(result.summary())
58 # Adj. R-squared:                  0.241
59 # Prob (F-statistic):           4.69e-08    ---------> 유의한 모델
60 # slope_GDP = 0.0142 : slope_Population = 8.611e-07 : bias(intercept) = -33.1082
```

- ✔ p-value : 4.69e-08로 0.05보다 적으므로 해당 모델은 유의한 모델
- ✔ 결정계수 : 0.241
  모델의 회귀선이 종속변수를 약 24%정도 설명한다고 할 수 있다.

모델을 이용해 예측 값 얻기

```
69 # 예측 준비 완료:
70 df.GDP = float(input('GDP를 입력하세요: '))
71 df.Population = int(input('Population를 입력하세요: '))
72 pred = result.predict(pd.DataFrame({'GDP':df.GDP,'Population':df.Population}))
73 print('예상 총 메달 개수는:', int(pred[0]), ' 입니다.')
```

작성한 모델을 이용해 키보드에서 새로운 값을 받아 예측 값을 얻는다. 아래는 예측 값을 얻은 실행결과이다.

```
GDP를 입력하세요: 27222
Population를 입력하세요: 100000000
예상 총 메달 개수는: 440  입니다.
```

분석 과정 | 분석 모델 “ 인구 수, GDP ” 적절성 확인

< 잔차항 >

- ✔ 잔차항 : 실제 값에서 예측 값을 뺀 값
- ✔ 실제 메달의 개수 - 메달 개수의 예측 값 = 잔차항을 계산

```
84 # 잔차항 (실제값 - 예측값) 구하기 (difference)
85 fitted = result.predict(df)   #  예측값
86 residual = df['medals'] - fitted   #  실제값 - 예측값
```

< 정규성 > - (불만족)

```
 97 print('정규성 : 잔차가 정규분포를 따라야함. shapiro test 사용')
 98 import scipy.stats as stats
 99 print('shapiro test: ', stats.shapiro(residual))
100 # pvalue=3.4905936175183 79e-19 < 0.05 정규성을 만족 못함
```

< 독립성 > - (불만족)

```
113 print('독립성 : 잔차가 독립적, 자기상관(인접 관측치와 오차가 상관되어있음) 이 없어야 함')
114 print('더빈왓슨 값으로 확인: Durbin-Watson: 0.763')
115 # 더빈왓슨 값으로 확인: Durbin-Watson:                      0.763
116 # (0에 가까우면 양의 상관, 4에 가까우면 음의 상관. 2에 가까우면 자기상관이 없다)
117 # 그러므로 양의 상관관계이지만, 독립성이 부족함
```

```
==========================================
Durbin-Watson:                   0.763
```

- 모델의 선형성, 정규성, 독립성에 대한 설명은 앞의 단순 회귀 분석에 기재했으므로 내용의 중복을 피하기 위해 별도 기재하지 않았다.

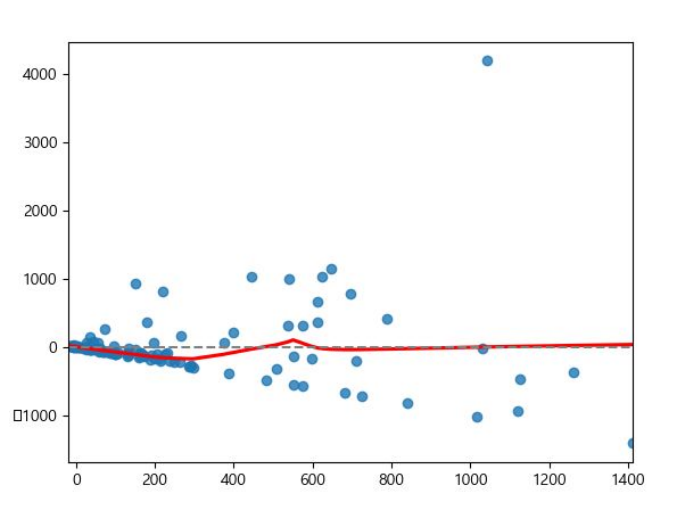
분석 과정 | 분석 모델

“ 인구 수, GDP ” 적절성 확인

< 선형성 > - (불만족)

```
print('선형성 : 예측값과 잔차가 비슷한 패턴을 가짐')
sns.regplot(fitted, residual, line_kws={'color':'red'}, lowess=True)  # regplot(예측값, 잔차값)
plt.plot([fitted.min(), fitted.max()], [0, 0], '--', color='grey')
plt.show()  # 잔차의 추세선과 예측값이 비슷한 패턴을 가지지 않아 선형성을 완전하게 만족하지는 못함
```

- 선형성 확인 결과를 시각화



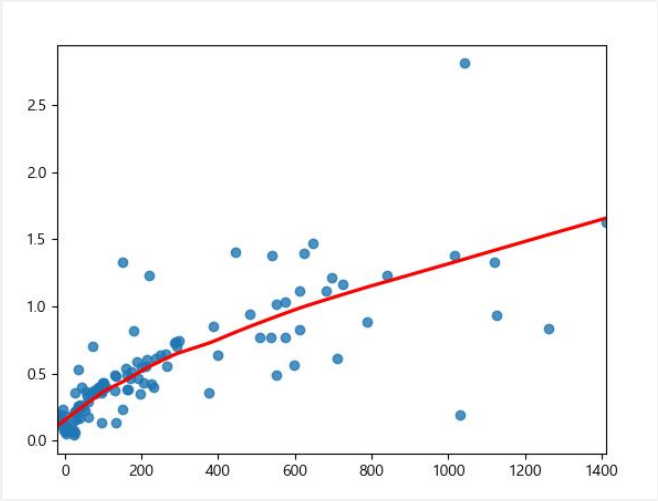
< 등분산성 > - (불만족)

✔ 등분산성 : 잔차의 분산이 일정함

*독립변수가 두 개 이상일 때 확인하는 요소

```
print('등분산성 : 잔차의 분산이 일정')
sns.regplot(fitted, np.sqrt(np.abs(sr)), lowess=True, line_kws={'color':'red'})
plt.show()  # # 잔차의 추세선이 등분산성 만족 못함.
# 또한 정규성과 선형성을 만족하지 못하므로 등분산성에도 문제가 있다.
```

- 등분산성 확인 결과를 시각화



분석 과정 | 분석 모델 “ 인구 수, GDP ” 적절성 확인

< 다중 공선성 > - 만족

다중 공선성 : 독립변수들 간에 강한 상관관계가 있는 경우
*독립변수가 두 개 이상일 때 확인하는 요소

```
print('다중 공선성 : 독립변수들 간에 강한 상관관계가 있는 경우')
# VIF(분산 인플레 요인) 값이 10을 넘으면 다중 공선성 발생
from statsmodels.stats.outliers_influence import variance_inflation_factor
print(model.exog_names) # 모델의 독립변수명들을 출력 ['Intercept', 'GDP', 'Population']
print(variance_inflation_factor(model.exog, 1))
# 모델의 독립변수 중 GDP의 다중 공선성 확인. 1.008295250855819 < 10
print(variance_inflation_factor(model.exog, 2))
# 모델의 독립변수 중 Population의 1.008295250855819 < 10
#출력되는 수치가 VIF이고, 해당 값이 10 이상일 경우 다중공선성이 발생한다.
```

결론

01 개최국과 올림픽 성적은 양의 상관 관계가 있다

02 GDP와 올림픽 성적은 양의 상관 관계가 있다

03 인구수, GDP와 올림픽 성적은 양의 상관 관계가 있다

보완점

개최국

- 카이 제곱 검정으로 인하여 시각화 구현 불가

- 분석결과로 독립변수와 종속변수 사이의 연관성이 있다는 결론을 내렸지만, 분석방법이 '카이 제곱'이었기 때문에 '인과성'을 나타낼 수는 없다. (독립변수 (개최국여부)) 자체로서 종속변수에 영향을 미쳤다고 할 수는 없다.

GDP

- 한정된 기간(2015년)의 GDP 데이터이므로 모델 검정에 있어서 만족하지 못함.

보완점

인구수, GDP

- 해당 모델에서 이상치가 2개가 발견되었으나
  이를 제거했을 때 예측 값을 찾는데
  오류가 발생하여 제거하지 못함

- 다중 선형회귀분석모델 시각화를 위해
  여러가지 방법을 찾아서 노력해보았으나 구현 불가

참조

- R-squared(결정계수) 참고 :
GDP, GDP+인구수 모델의 결정계수가 각각 0.194, 0.241로 약 19%, 24%정도로 비교적 높지 않아 보이지 않는다. 하지만 통계학자 Cohen, J에 의하면, 사회과학 연구에서 결정계수가 13%이상만 되면 어느정도 효과가 있다고 할 수 있다.

사용 도구

Google

eclipse

Python

Excel

pandas

NumPy

Matplotlib

scipy

stats

seaborn

참고 사이트

- ✔ kaggle 120년의 올림픽 선수 데이터 세트
  https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results
- ✔ 위키백과 올림픽 개최국, 해당 개최국의 IOC 국가 코드 정보 수집
  https://ko.wikipedia.org/wiki/%EB%8F%99%EA%B3%84_%EC%98%AC%EB%A6%BC%ED%94%BD
  https://ko.wikipedia.org/wiki/IOC_%EA%B5%AD%EA%B0%80_%EC%BD%94%EB%93%9C_%EB%AA%A9%EB%A1%9D

- ✔ kaggle, github 국가별 코드, 인구 수, GDP, 메달 수
  https://www.kaggle.com/the-guardian/olympic-games
  https://github.com/sonia3187/Light-Pollution-Project/blob/master/GDP.csv

- ✔ 결정계수 참고
  https://m.cafe.daum.net/ILoveSPSS/EOon/28?q=D_ohkQUOSAyUE0&

감사합니다