016. 간단한 회귀 문제 - 삶의 만족도

DataScience

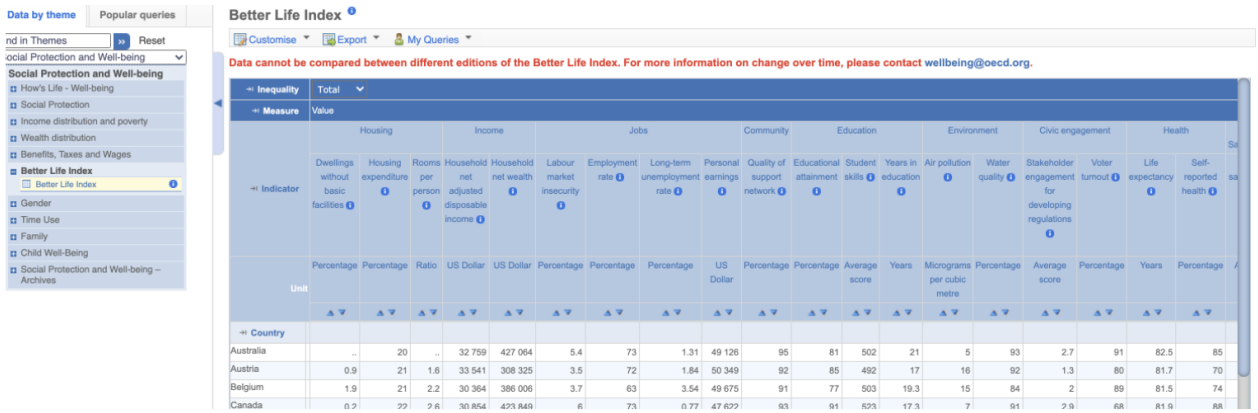
Exported on 11/16/2020

Table of Contents

1 삶의 만족도 예측

1.1 더 나은 삶의 지표 데이터

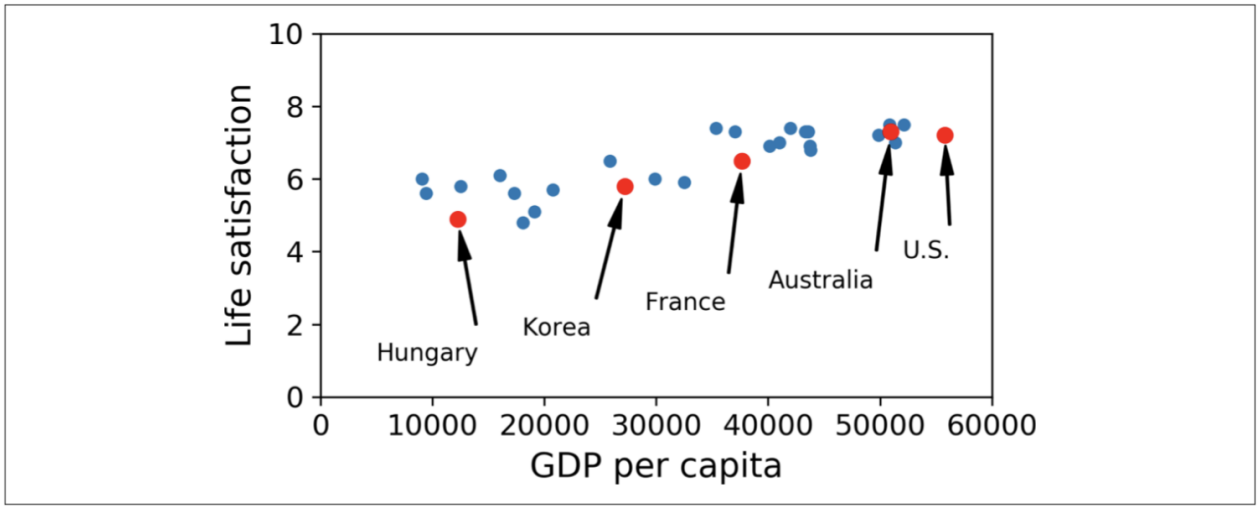
- https://stats.oecd.org/index.aspx?DataSetCode=BLI



1.2 그 중 몇몇 나라의 GDP와 삶의 만족도

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

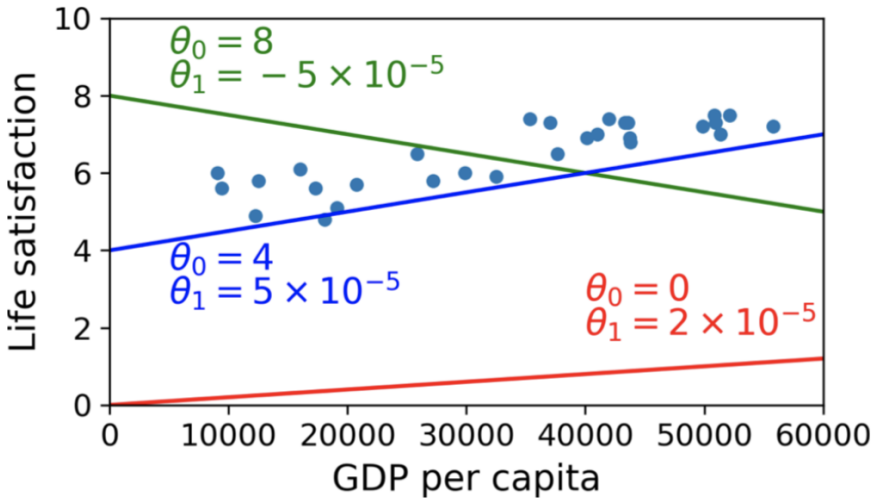
1.3 전체 데이터 시각화를 대충 손으로~



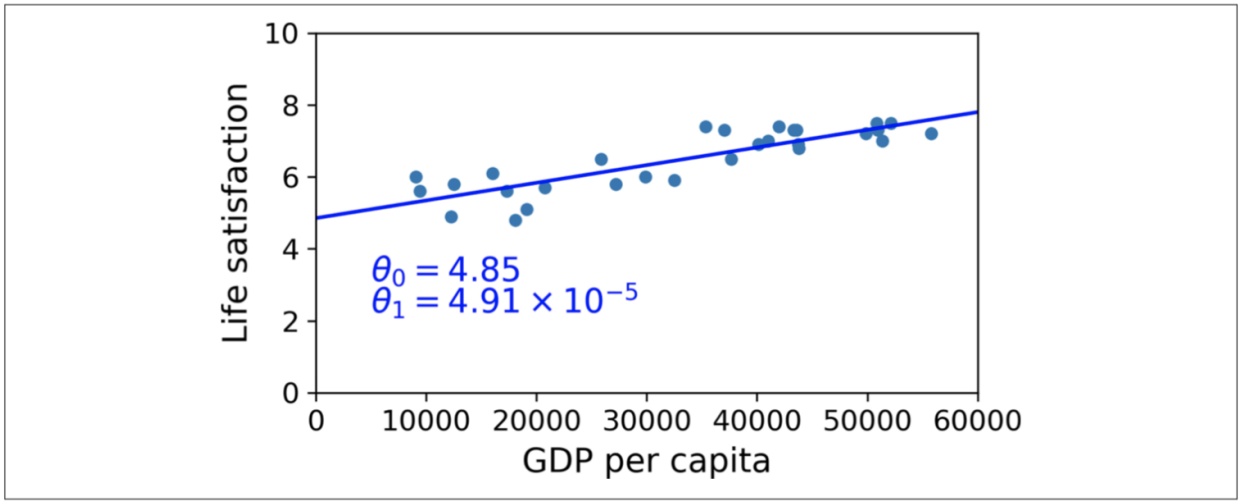
- trend가 보이나요?

1.4 삶의 만족도는 1인당 GPD로 회귀 모델을 구성할 수 있을까

$$\text{life\_satisfaction} = \theta_0 + \theta_1 \times \text{GDP\_per\_capita}$$



1.5 어떻게 찾았다고 치자~

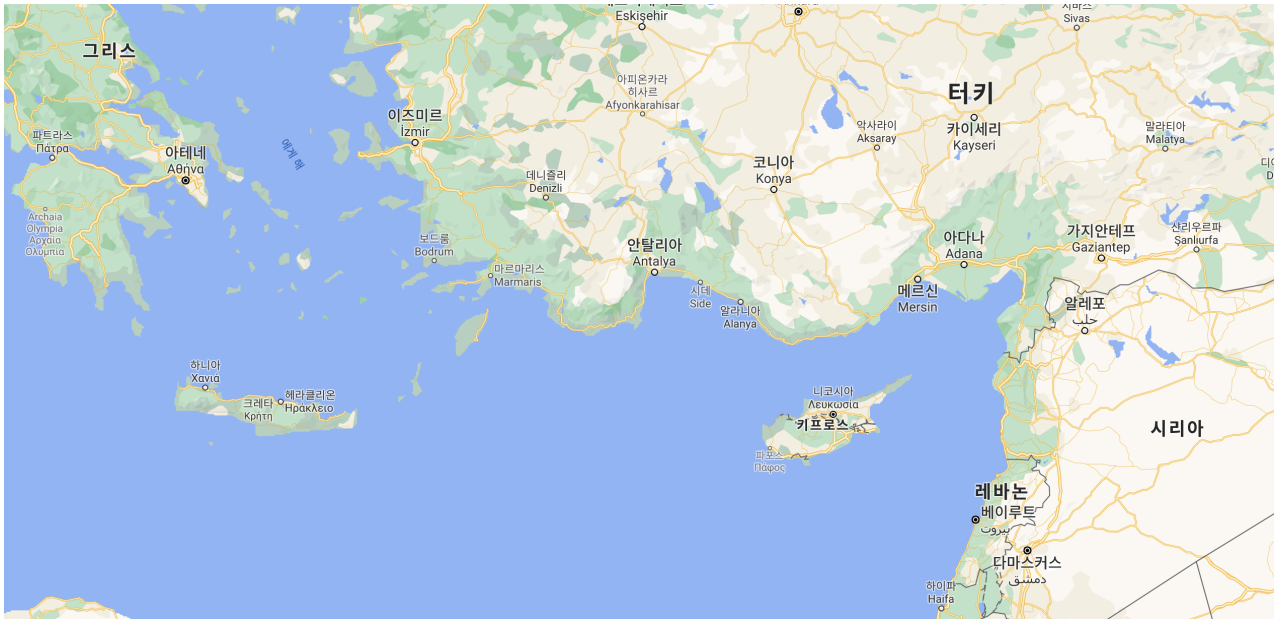


1.6 이제 모델을 사용해보자~

- 키프로스 : 이 나라는 위 사이트에 없다.
- 키프로스의 GDP는 22,587달러이고

$$4.85 + 22{,}587 \times 4.91 \times 10^{-5} = 5.96.$$

1.7 파란 만장한 역사의 키프로스~



1.8 dive to code

```
import sklearn
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sklearn.linear_model
```

1.9 이젠 이런 스타일도 익숙해지자

```
%matplotlib inline
import matplotlib as mpl
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)
```

1.10 데이터 읽고

```
oecd_bli = pd.read_csv("./data/oecd_bli_2015.csv", thousands=',')
gdp_per_capita = pd.read_csv("./data/gdp_per_capita.csv",
                             thousands=',',
                             delimiter='\t',
                             encoding='latin1', na_values="n/a")
```

1.11 복잡하지만

```
oecd_bli.head()
```

	LOCATION	Country	INDICATOR	Indicator	MEASURE	Measure	INEQUALITY	Inequality	Unit Code	Unit	P
0	AUS	Australia	HO_BASE	Dwellings without basic facilities	L	Value	TOT	Total	PC	Percentage	0
1	AUT	Austria	HO_BASE	Dwellings without basic facilities	L	Value	TOT	Total	PC	Percentage	0
2	BEL	Belgium	HO_BASE	Dwellings without basic facilities	L	Value	TOT	Total	PC	Percentage	0

1.12 데이터 정리

```
oecd_bli = oecd_bli[oecd_bli["INEQUALITY"]=="TOT"]
oecd_bli.head()
```

	LOCATION	Country	INDICATOR	Indicator	MEASURE	Measure	INEQUALITY	Inequality	Unit Code	Unit	P
0	AUS	Australia	HO_BASE	Dwellings without basic facilities	L	Value	TOT	Total	PC	Percentage	0
1	AUT	Austria	HO_BASE	Dwellings without basic facilities	L	Value	TOT	Total	PC	Percentage	0
2	BEL	Belgium	HO_BASE	Dwellings without basic facilities	L	Value	TOT	Total	PC	Percentage	0
3	CAN	Canada	HO_BASE	Dwellings without basic facilities	L	Value	TOT	Total	PC	Percentage	0

1.13 국가별로 정리

```
oecd_bli = oecd_bli.pivot(index="Country", columns="Indicator", values="Value")
oecd_bli.head()
```

Indicator	Air pollution	Assault rate	Consultation on rule-making	Dwellings without basic facilities	Educational attainment	Employees working very long hours	Employment rate	Homicide rate	Household net adjusted disposable income
Country									
Australia	13.0	2.1	10.5	1.1	76.0	14.02	72.0	0.8	31588.0
Austria	27.0	3.4	7.1	1.0	83.0	7.61	72.0	0.4	31173.0
Belgium	21.0	6.6	4.5	2.0	72.0	4.57	62.0	1.1	28307.0
Brazil	18.0	7.9	4.0	6.7	45.0	10.41	67.0	25.5	11664.0
Canada	15.0	1.3	10.5	0.2	89.0	3.94	72.0	1.5	29365.0

5 rows × 24 columns

1.14 데이터 결합

```
gdp_per_capita.rename(columns={"2015": "GDP per capita"}, inplace=True)
gdp_per_capita.set_index("Country", inplace=True)
full_country_stats = pd.merge(left=oecd_bli, right=gdp_per_capita,
                              left_index=True, right_index=True)
full_country_stats.head()
```

Country	Air pollution	Assault rate	Consultation on rule-making	Dwellings without basic facilities	Educational attainment	Employees working very long hours	Employment rate	Homicide rate	Household net adjusted disposable income
Australia	13.0	2.1	10.5	1.1	76.0	14.02	72.0	0.8	31588.0
Austria	27.0	3.4	7.1	1.0	83.0	7.61	72.0	0.4	31173.0

1.15 살짝 정렬

```
full_country_stats.sort_values(by="GDP per capita", inplace=True)
full_country_stats.head()
```

Country	Air pollution	Assault rate	Consultation on rule-making	Dwellings without basic facilities	Educational attainment	Employees working very long hours	Employment rate	Homicide rate	Household net adjusted disposable income
Brazil	18.0	7.9	4.0	6.7	45.0	10.41	67.0	25.5	11664.0
Mexico	30.0	12.8	9.0	4.2	37.0	28.83	61.0	23.4	13085.0

1.16 이상치 데이터 정리

```
remove_indices = [0, 1, 6, 8, 33, 34, 35]
keep_indices = list(set(range(36)) - set(remove_indices))
full_country_stats[["GDP per capita",
                    'Life satisfaction']].iloc[keep_indices]
full_country_stats.head()
```

Country	Air pollution	Assault rate	Consultation on rule-making	Dwellings without basic facilities	Educational attainment	Employees working very long hours	Employment rate	Homicide rate	Household net adjusted disposable income
Brazil	18.0	7.9	4.0	6.7	45.0	10.41	67.0	25.5	11664.0

1.17 아무튼 위 과정을 모아서

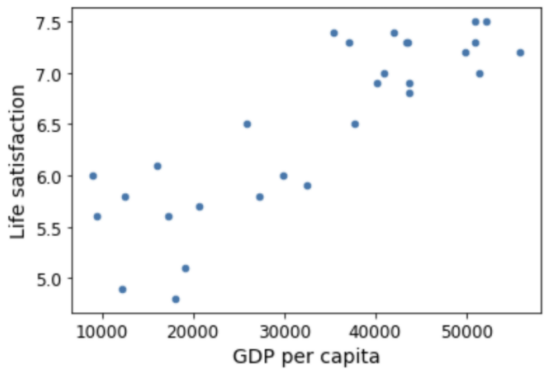
```
def prepare_country_stats(oecd_bli, gdp_per_capita):
    oecd_bli = oecd_bli[oecd_bli["INEQUALITY"]=="TOT"]
    oecd_bli = oecd_bli.pivot(index="Country", columns="Indicator",
                              values="Value")
    gdp_per_capita.rename(columns={"2015": "GDP per capita"}, inplace=True)
    gdp_per_capita.set_index("Country", inplace=True)
    full_country_stats = pd.merge(left=oecd_bli, right=gdp_per_capita,
                                  left_index=True, right_index=True)
    full_country_stats.sort_values(by="GDP per capita", inplace=True)
    remove_indices = [0, 1, 6, 8, 33, 34, 35]
    keep_indices = list(set(range(36)) - set(remove_indices))
    return full_country_stats[["GDP per capita",
                               'Life satisfaction']].iloc[keep_indices]
```

1.18 그림으로~

```
country_stats = prepare_country_stats(oecd_bli, gdp_per_capita)

X = np.c_[country_stats["GDP per capita"]]
y = np.c_[country_stats["Life satisfaction"]]

country_stats.plot(kind='scatter', x="GDP per capita", y='Life satisfaction')
plt.show()
```



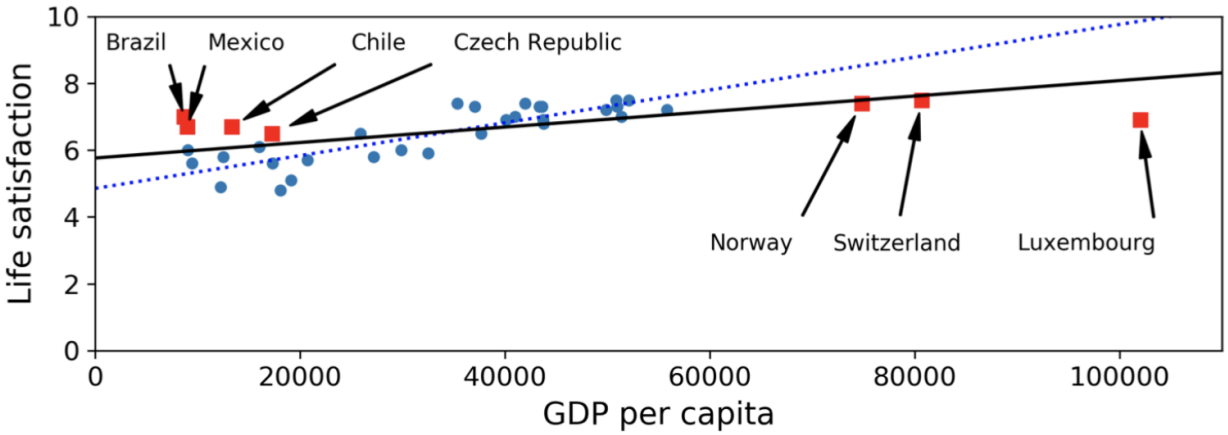
1.19 그래서 키프로스는?

```
model = sklearn.linear_model.LinearRegression()
model.fit(X, y)

X_new = [[22587]]  # 키프로스 1인당 GDP
print(model.predict(X_new))
```

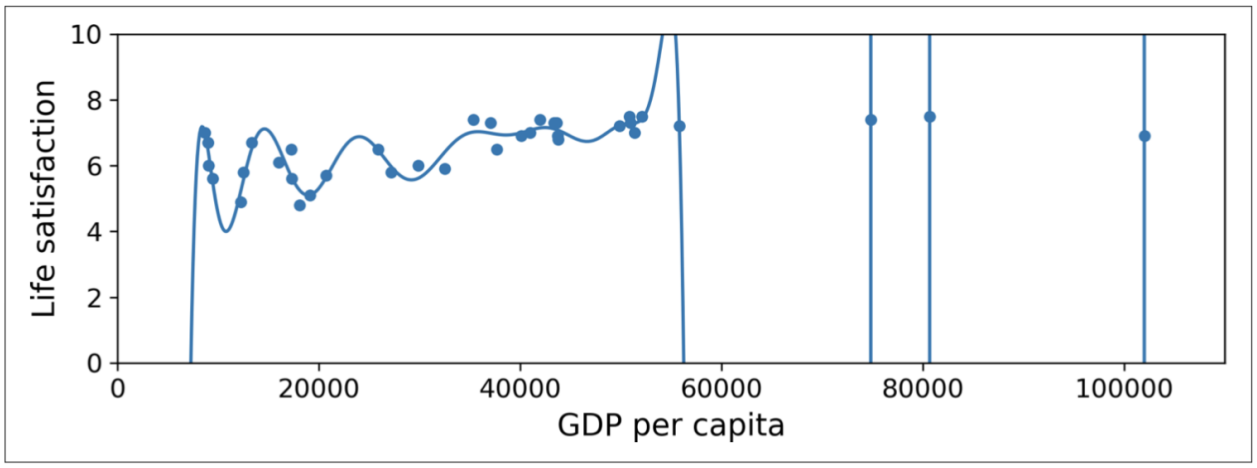
```
[[5.96242338]]
```

1.20 이상치를 제거하지 않았을때



- 일반화가 잘 되려면 일반화하기 원하는 새루운 사례를 훈련 데이터가 잘 대표해야~
- 샘플이 작으면 샘플링 잡음이 생김
- 샘플이 많아도 추출 방법이 잘못되면 샘플링 편향이 생길 수도 있음

1.21 과적합



- 주어진 데이터에 너무 잘 맞는 현상
- 훈련 데이터의 잡음의 양에 비해 모델이 너무 복잡할때 발생

1.22 공짜 점심 없음 이론

- 1996년 데이비드 월퍼트
  - 데이터에 관해 완벽하게 어떤 가정도 하지 않으면 한 모델을 다른 모델보다 선호할 근거가 없음
  - 경험하기 전에는 더 잘 맞을 것이라고 보장할 수 있는 모델은 없다