Ⅵ. 부록

[부록 목차]

[부록1] 데이터 출처와 변수 설명

데이터 출처: Kaggle Dataset – WHO Data & UN Data

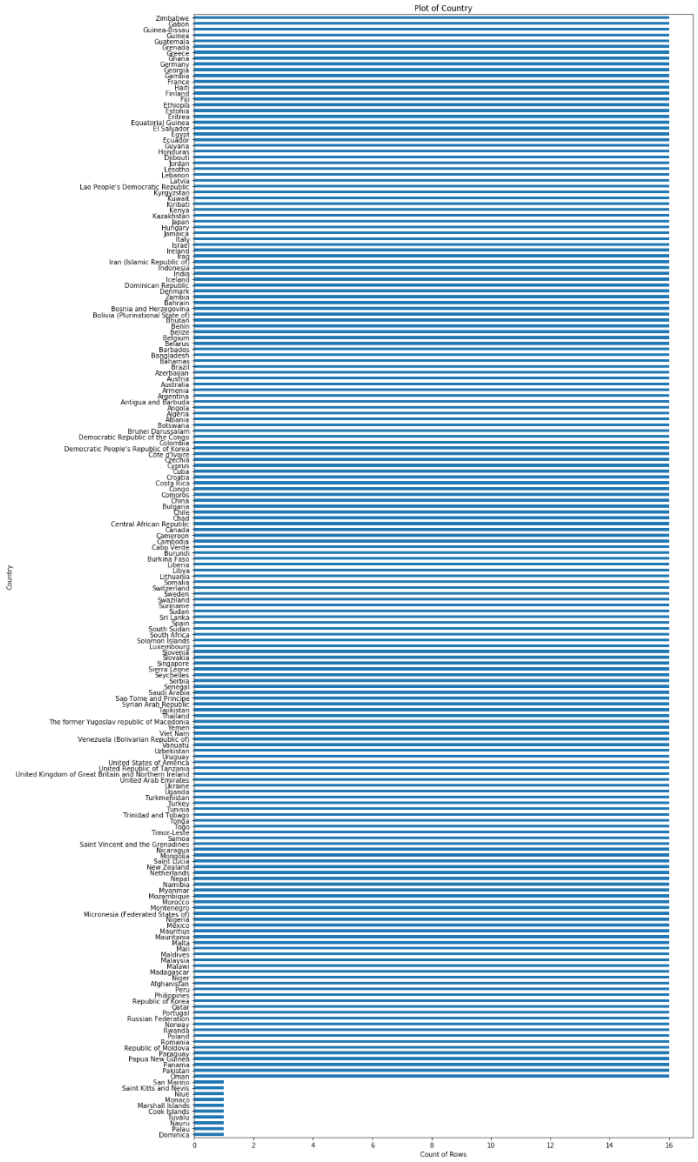
데이터요약과 변수요약은 다음과 같다.

	Country	Year	Status	Life_Expectancy	Adult_Mortality	Infant_Deaths	…	Schooling
1	Afghanistan	2015	Developing	65.0	263.0	62	…	10.1
2	Afghanistan	2014	Developing	59.9	271.0	64	…	10.0
3	Afghanistan	2013	Developing	59.9	268.0	66	…	9.9
4	Afghanistan	2012	Developing	59.5	272.0	69	…	9.8
5	Afghanistan	2011	Developing	59.2	275.0	71	…	9.5
…	…	…	…	…	…	…	…	…
2934	Zimbabwe	2004	Developing	44.3	723.0	27	…	9.2
2935	Zimbabwe	2003	Developing	44.5	715.0	26	…	9.5
2936	Zimbabwe	2002	Developing	44.8	73.0	25	…	10.0
2937	Zimbabwe	2001	Developing	45.3	686.0	25	…	9.8
2938	Zimbabwe	2000	Developing	46.0	665.0	24	…	9.8

변수명	변수 설명
Country	193개국
Year	정보가 수집된 2000 – 2015년까지의 년도
Status	1=개발도상국, 0=선진국
Life_Expectancy	기대 수명
Adult_Mortality	성인 사망률 (천분율 ‰)
Infant_Deaths	유아 사망률 (천분율 ‰)
Alcohol	1인당 알코올 소비 비율
Percentage_Expenditure	1인당 GDP대비 건강에 대한 지출 비율
Hepatitis_B	1세 아동의 B형간염 예방 접종 비율
Measles	인구 1000명당 홍역 발병 횟수 (천분율 ‰)
BMI	전체 인구의 평균 체질량지수
Under5_Deaths	인구 1000명당 5세이하 사망자수 (천분율 ‰)
Polio	1세 아동의 소아마비 예방 접종 비율
Total_Expenditure	정부 지출 대비 건강에 대한 지출 비율
Diphtheria	1세 영아들의 DTP3 예방접종 비율
HIV_AIDS	HIV_AIDS로 인한 사망률 (천분율 ‰)
GDP	1인당 국내총생산
Population	인구 수
Thinnes_10_19_years	저체중 10-19세 청소년의 비율
Thinness_5_9_years	저체중 5-9세 어린이의 비율
Income_Composition_Of_Resources	HRDI
Schooling	평균 교육 기간

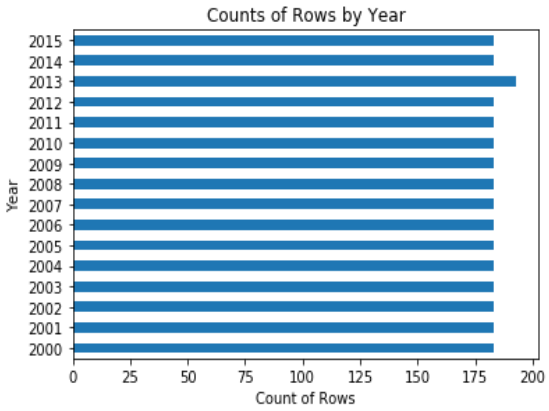
① Country

Country는 건강과 수명에 관한 정보가 수집된 193개국에 대한 범주형 변수이다.



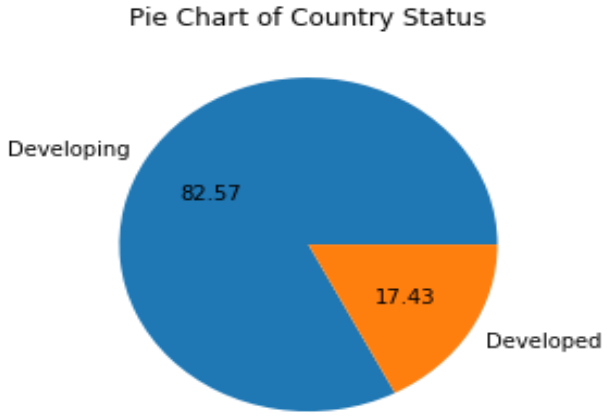
② Year

Year는 정보가 수집된 2000 – 2015년까지의 년도에 대한 정수형 변수이다. 대부분의 연도가 차지하는 행수는 비슷하지만 2013년의 경우 차지하는 행수가 더 많은 것을 확인할 수 있다. 이는 다른 해에 비해 2013년에 더 많은 정보가 수집되었다고 판단할 수 있다.



③ Status

Status는 각 국가가 개발도상국인지 선진국인지를 나타내는 범주형 변수로, 개발도상국은 developing, 선진국은 developed의 범주로 분류된다. 파이차트로 보아, 데이터 내에 선진국보다 개발도상국인 국가가 더 많음을 확인할 수 있다.

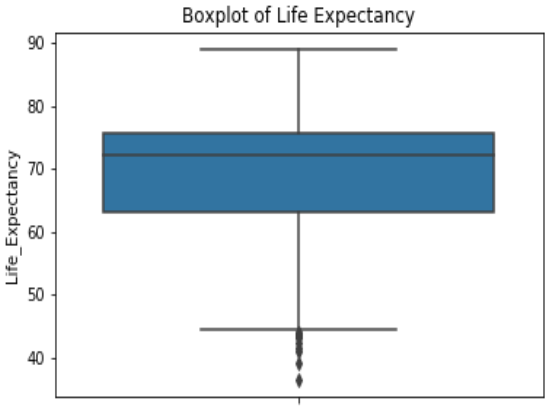


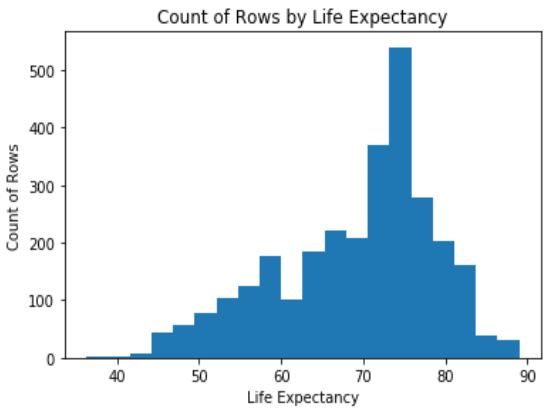
④ Life_Expectancy

Life_Expectancy는 각 국가의 기대수명에 대한 변수로, 본 조의 모델에서 반응변수에 해당하는 변수이다.

해당변수의 통계량은 아래와 같다. 70대 중후반의 기대수명을 가지는 국가가 많음을 알 수 있다.

Life_Expectancy			
Mean	69.2249	Median	72.1000
Min	36.3000	Max	89.0000
25% Percentile	63.1000	75% Percentile	75.7000
Std	9.5239		

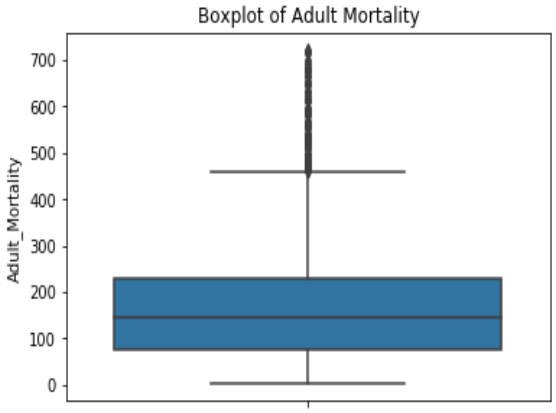


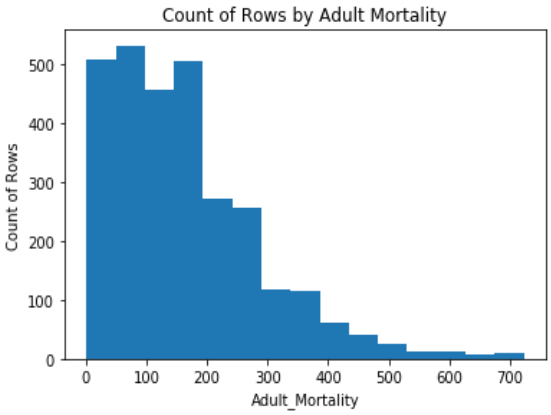


⑤ Adult_Mortality

Adult_Mortality는 성인 1000명당 사망한 15-60세 인구수에 대한 변수로, 통계량은 다음과 같다. 인구 1000명당 사망하는 15세-60세 인구는 주로 150명 이하임을 알 수 있다. 또한, 성인의 평균 사망률이 16.5%이다.

Adult_Mortality			
Mean	164.7964	Median	14.0000
Min	1.0000	Max	723.0000
25% Percentile	74.0000	75% Percentile	228.0000
Std	124.2921		

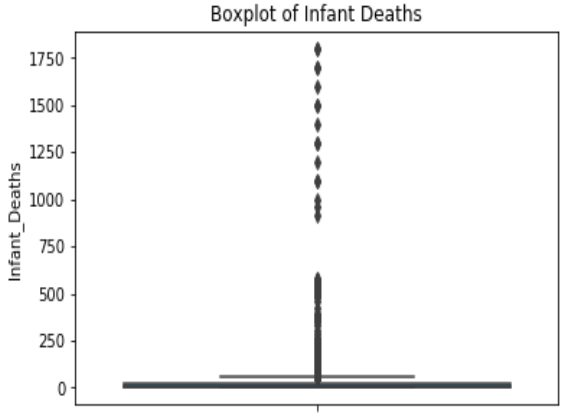


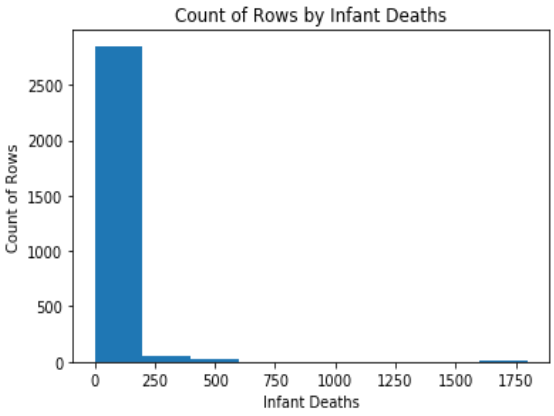


⑥ Infant_Deaths

Infant_Deaths는 유아 1000명당 사망한 신생아 수에 대한 변수로, 통계량은 다음과 같다. 대부분의 국가에서 인구 1000명당 사망하는 신생아수가 0명에 가까운것을 알 수 있다. 또한, 신생아의 평균 사망률이 3%이다.

Infant_Deaths			
Mean	30.3039	Median	3.0000
Min	0.0000	Max	1800.0000
25% Percentile	0.0000	75% Percentile	22.0000
Std	117.9265		

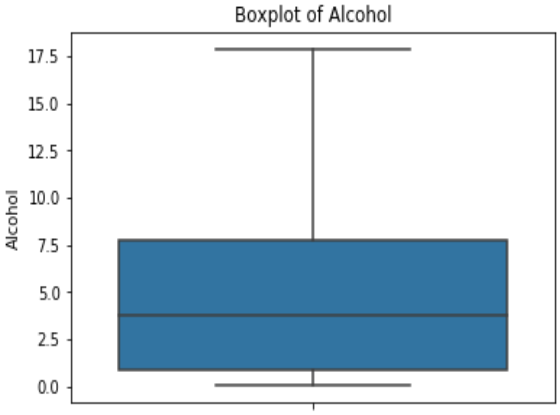


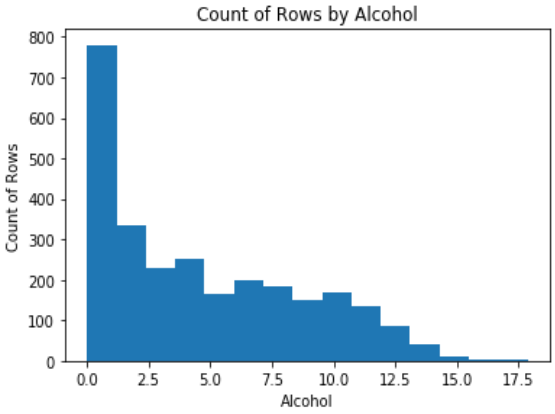


⑦ Alcohol

Alcohol은 인구 1명당 술 소비량의 비율에 대한 변수로, 통계량은 다음과 같다.

Infant_Deaths			
Mean	4.6029	Median	3.7550
Min	0.0100	Max	17.8700
25% Percentile	0.8775	75% Percentile	7.7025
Std	4.0524		

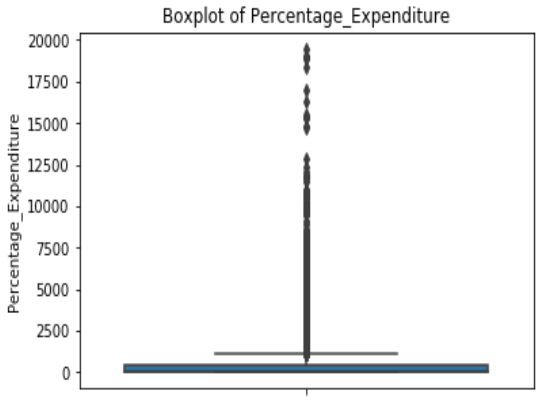


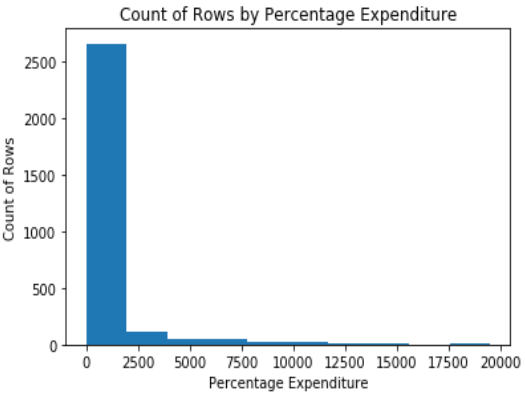


⑧ Percentage_Expenditure

Percentage_Expenditure는 인구 1명당 국가의 GDP에 비하여 건강에 지출하는 비용을 비율로 나타낸 변수로, 통계량은 다음과 같다.

Infant_Deaths			
Mean	738.2513	Median	64.9129
Min	0.0000	Max	19479.911610
25% Percentile	4.6853	75% Percentile	441.5341
Std	1987.914858		

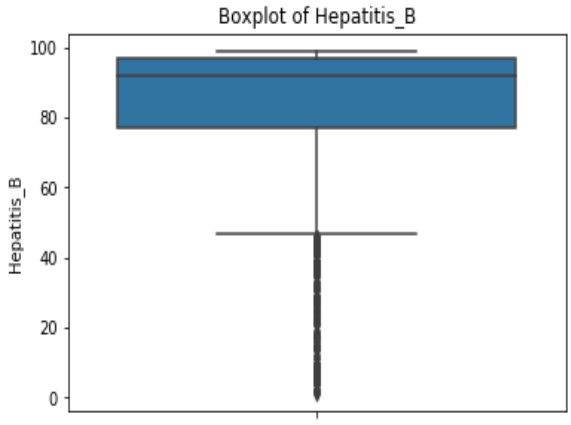


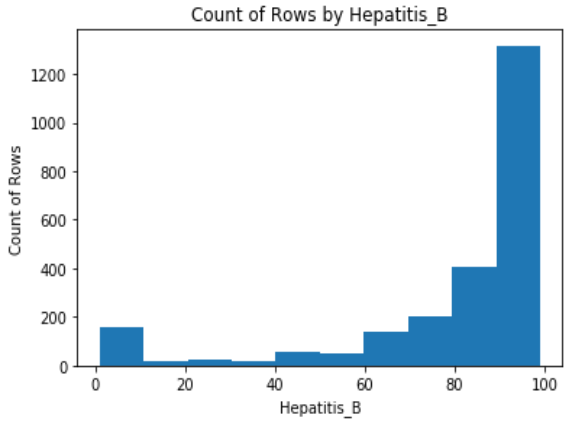


⑨ Hepatitis_B

Hepatitis_B는 1세 영아들의 B형간염 예방접종 비율을 의미하는 변수로, 통계량은 다음과 같다. Hepatitis_B의 비율이 80%이상인 국가가 많음을 알 수 있다.

Hepatitis_B			
Mean	80.9405	Median	92.0000
Min	1.0000	Max	99.0000
25% Percentile	77.0000	75% Percentile	97.0000
Std	25.0700		

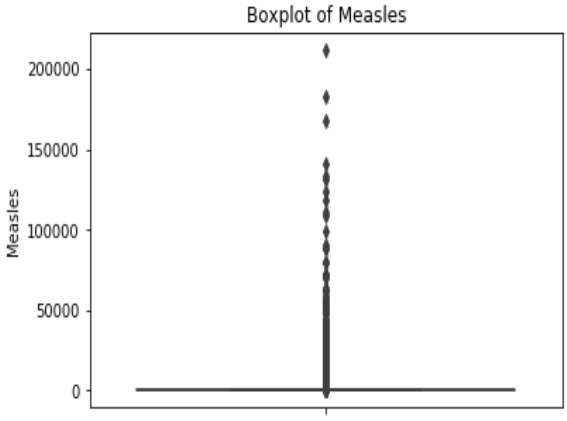


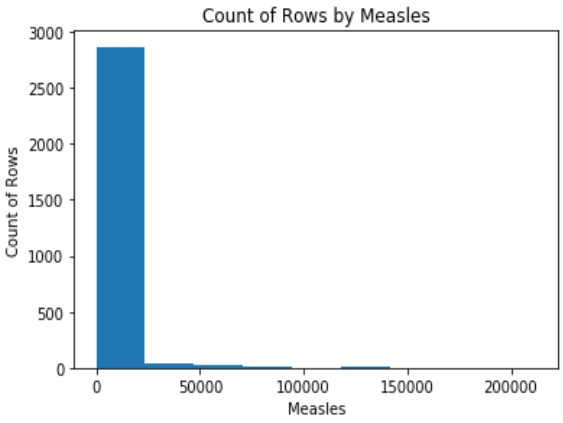


⑩ Measles

Measles는 인구 1000명당 홍역 발병 수에 관한 변수로, 통계량은 다음과 같다. 대부분의 국가에서 Measles가 400 이하라는 것을 알 수 있다. 또한, 홍역 발생률의 중위값은 1.7%이다.

Measles			
Mean	2419.5922	Median	17.0000
Min	0.0000	Max	212183
25% Percentile	0.0000	75% Percentile	360.2500
Std	111467.2725		

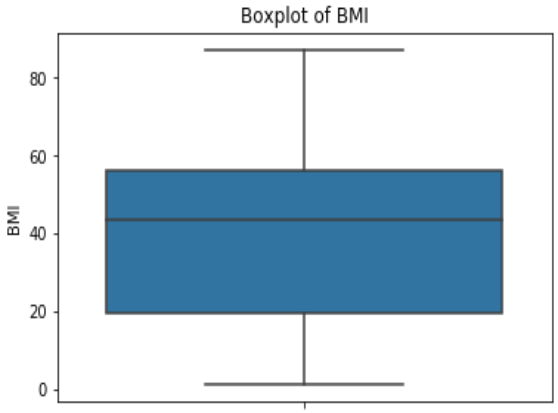


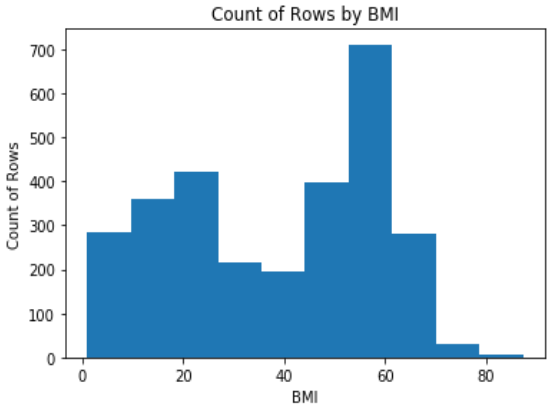


⑪ BMI

BMI는 각 국가마다 전체 인구의 평균 체질량지수를 의미하는 변수로, 통계량은 다음과 같다. BMI가 50 – 60% 사이의 빈도가 가장 높다.

BMI			
Mean	38.3212	Median	43.5000
Min	1.0000	Max	87.3000
25% Percentile	19.3000	75% Percentile	56.2000
Std	20.0440		

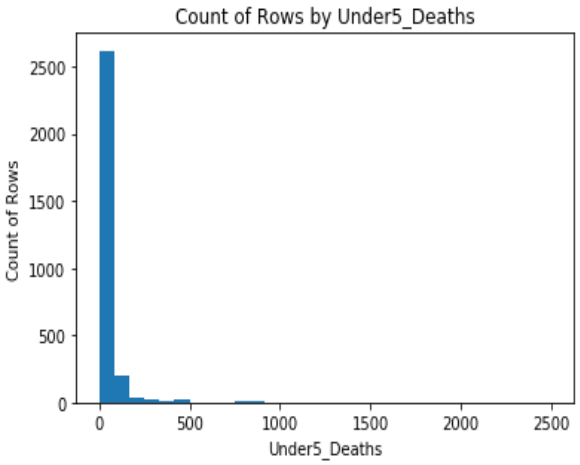




⑫ Under5_Deaths

Under5_Deaths는 인구 1000명당 사망한 5세 이하 유아수에 대한 변수로, 통계량은 다음과 같다. Under5_Deaths는 0 – 100명의 범위내에 편향되어 있는 것을 볼 수 있다. 유아의 평균 사망률은 4.2%이다.

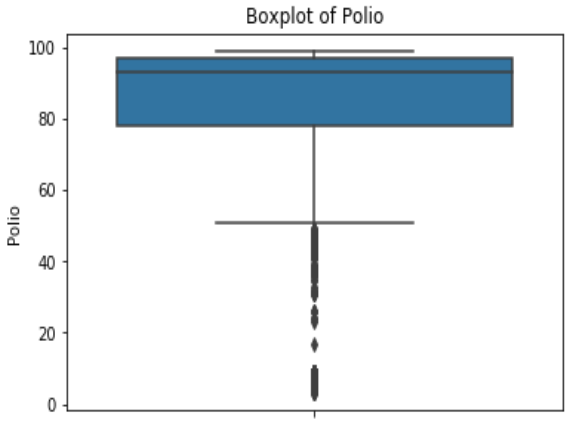
Under5_Deaths			
Mean	42.0357	Median	4.0000
Min	0.0000	Max	2500
25% Percentile	0.0000	75% Percentile	28.0000
Std	160.4455		

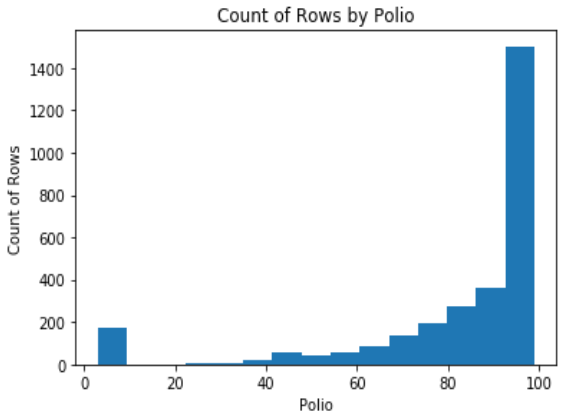


⑬ Polio

Polio는 1세 영아들의 소아마비 예방접종 비율에 관한 변수로, 통계량은 다음과 같다. Polio는 80 – 100% 사이에 편향되어 있음을 알 수 있다.

Polio			
Mean	82.5502	Median	93.0000
Min	3.0000	Max	99.0000
25% Percentile	78.0000	75% Percentile	97.0000
Std	23.428046		

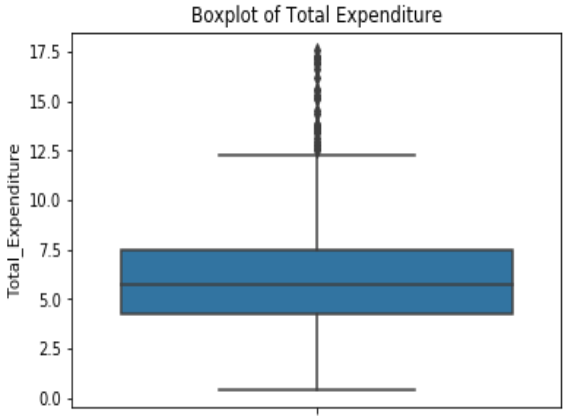


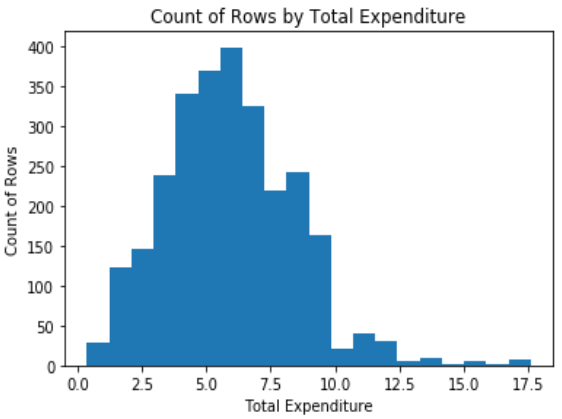


⑭ Total_Expenditure

Total_Expenditure는 정부의 전체지출에 대한 건강부문에 지출하는 비용에 대한 비율로, 통계량은 다음과 같다. Total Expenditure가 4.0 – 10.0%인 국가가 많음을 알 수 있다.

Total_Expenditure			
Mean	5.9382	Median	5.7550
Min	0.3700	Max	12.6000
25% Percentile	4.2600	75% Percentile	7.4925
Std	2.4983		

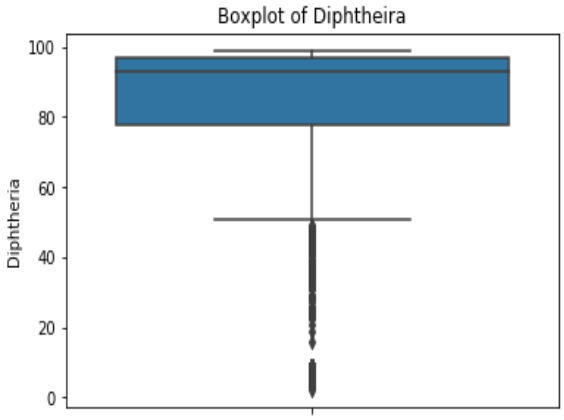


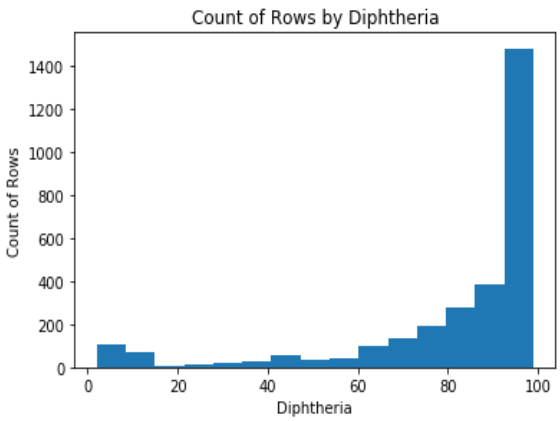


⑮ Diphtheria

Diphtheria는 각 국가별 1세 영아들의 DTP3 예방접종 비율에 관한 변수로, 통계량은 다음과 같다. Diphtheria는 90 – 100%에 편향되어 있는 것을 볼 수 있다.

Diphtheria			
Mean	82.3241	Median	93.0000
Min	2.0000	Max	99.0000
25% Percentile	78.0000	75% Percentile	97.0000
Std	23.7169		

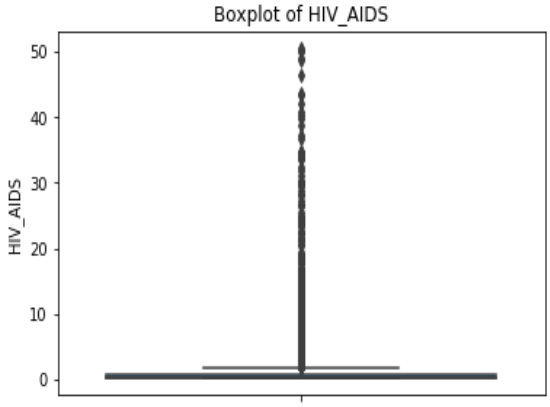


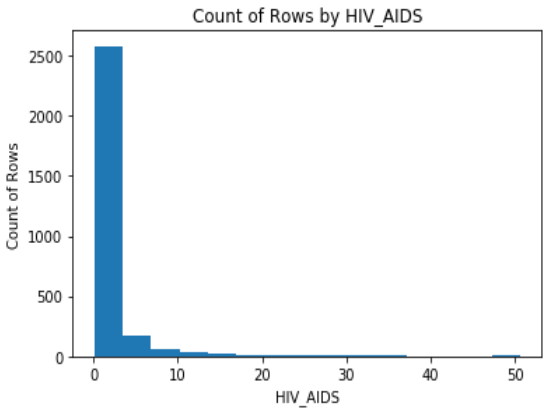


⑯ HIV_AIDS

HIV_AIDS는 유아 1000명당 AIDS로 사망한 유아의 비율을 나타내는 변수로, 통계량은 아래와 같다. HIV_AIDS는 0 – 5의 범위내에 편향되어 있음을 알 수 있다. 또한, 에이즈로 인한 유아 평균 사망률은 0.17%이다.

HIV_AIDS			
Mean	1.7421	Median	0.1000
Min	0.0000	Max	50.6000
25% Percentile	0.1000	75% Percentile	0.8000
Std	5.0778		

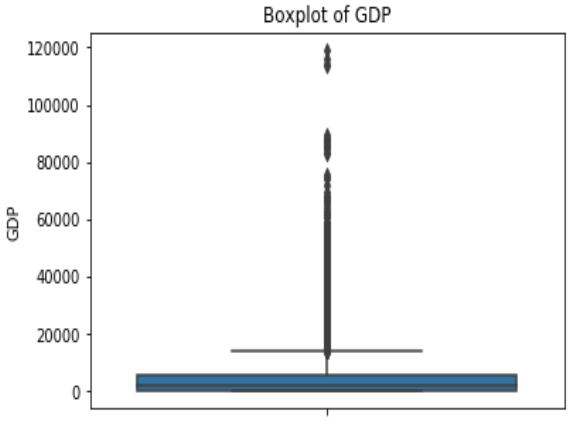


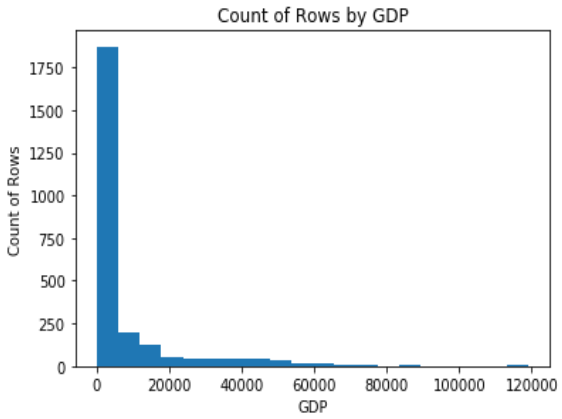


⑰ GDP

GDP는 각 국가별 1인당 국내총생산으로 단위는 미국 달러(USD)이며 통계량은 다음과 같다.

GDP			
Mean	1.6814	Median	1766.9476
Min	1.6814	Max	119172.7418
25% Percentile	463.9536	75% Percentile	5910.8063
Std	14270.1693		

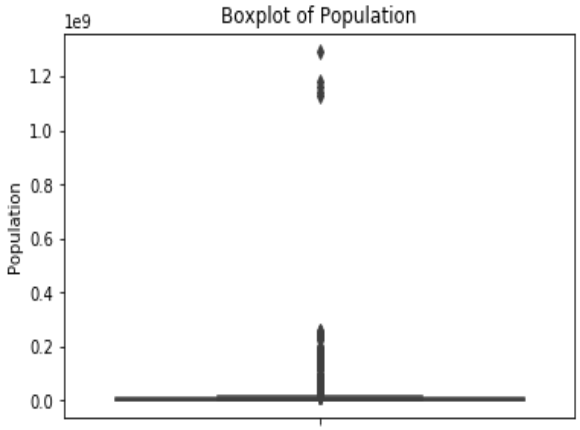


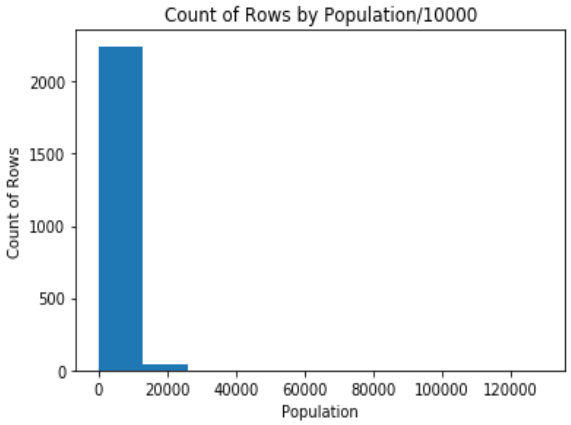


⑱ Population

Population은 각 국가의 인구수로, 통계량은 다음과 같다.

Population			
Mean	1.275338e+07	Median	1.386542e+06
Min	3.4000e+01	Max	1.293859e+09
25% Percentile	1.957932e+05	75% Percentile	7.420359e+06
Std	6.101210e+07		

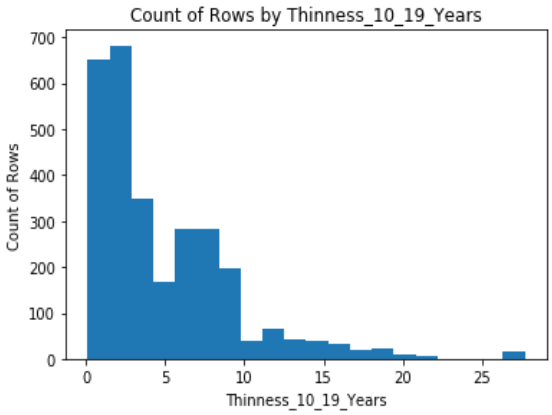




⑲ Thinness_10_19_Years

Thinness_10_19_years는 야윈 10-19세 청소년의 비율에 대한 변수이므로 저체중인 10-19세 청소년의 비율로 생각할 수 있다. 통계량은 다음과 같고 Thinness_10_19_Years는 오른쪽으로 꼬리가 긴 형태의 분포를 따른다고 볼 수 있다.

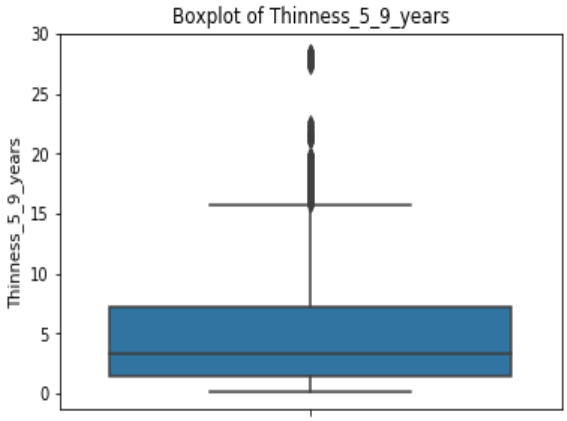
Thinness_10_19_Years			
Mean	4.8397	Median	3.3000
Min	0.1000	Max	27.7000
25% Percentile	1.6000	75% Percentile	7.2000
Std	4.4202		

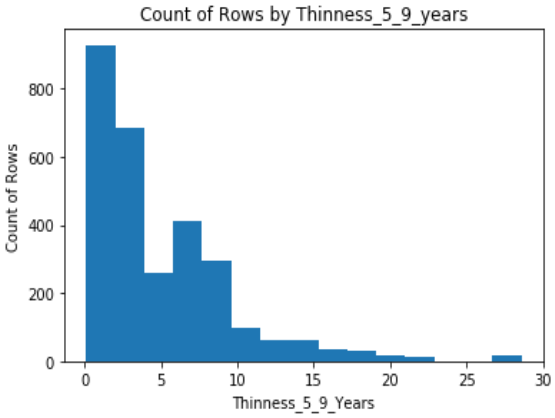


⑳ Thinness_5_9_years

Thinness_5_9_years는 각 국가별 야윈 5-9세 어린이의 비율이므로 저체중인 5-9세 어린이의 비율로 생각할수 있다. 통계량은 다음과 같고 Thinness_5_9_years는 오른쪽으로 꼬리가 긴 형태의 분포를 따름을 알 수 있다.

Thinness_5_9_years			
Mean	4.8703	Median	3.3000
Min	0.1000	Max	28.6000
25% Percentile	1.5000	75% Percentile	7.2000
Std	4.5089		

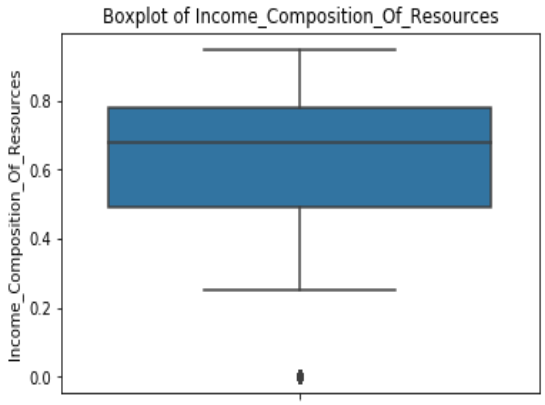


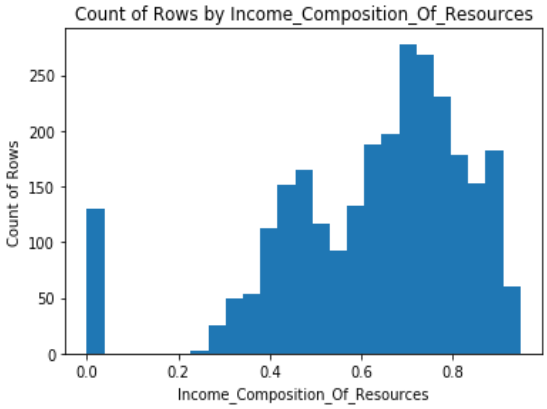


㉑ Income_Composition_Of_Resources

Income_Composition_Of_Resources는 인적자원개발지수를 나타내는 변수로, HRDI라 표기한다. 통계량은 다음과 같다.

Income_Composition_Of_Resources			
Mean	0.6276	Median	0.6770
Min	0.0000	Max	0.9480
25% Percentile	0.4930	75% Percentile	0.7790
Std	0.2109		

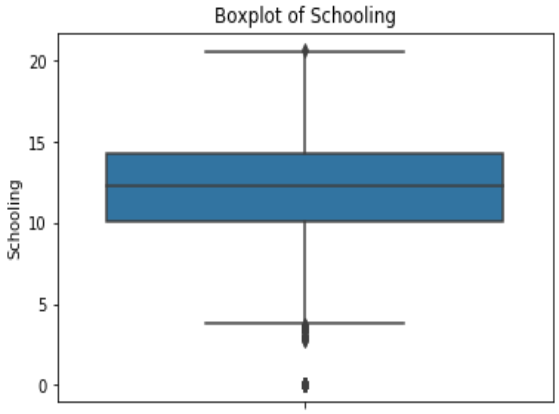


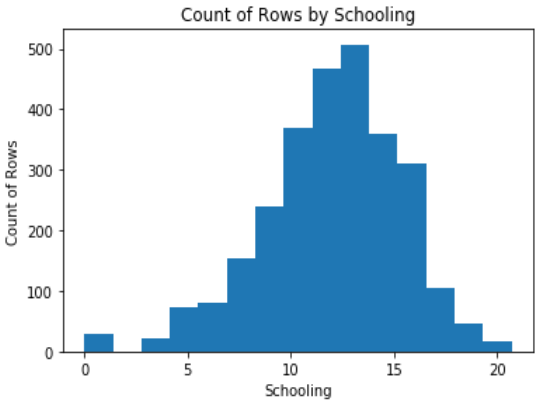


㉒ Schooling

Schooling은 각 국가별 교육 년수에 관한 변수이며 통계량은 다음과 같다. Schooling은 10-15년의 범위내에 가장 도수가 크다는 것을 알 수 있다.

Schooling			
Mean	11.9928	Median	12.3000
Min	0.0000	Max	20.7000
25% Percentile	10.1000	75% Percentile	14.3000
Std	3.3589		

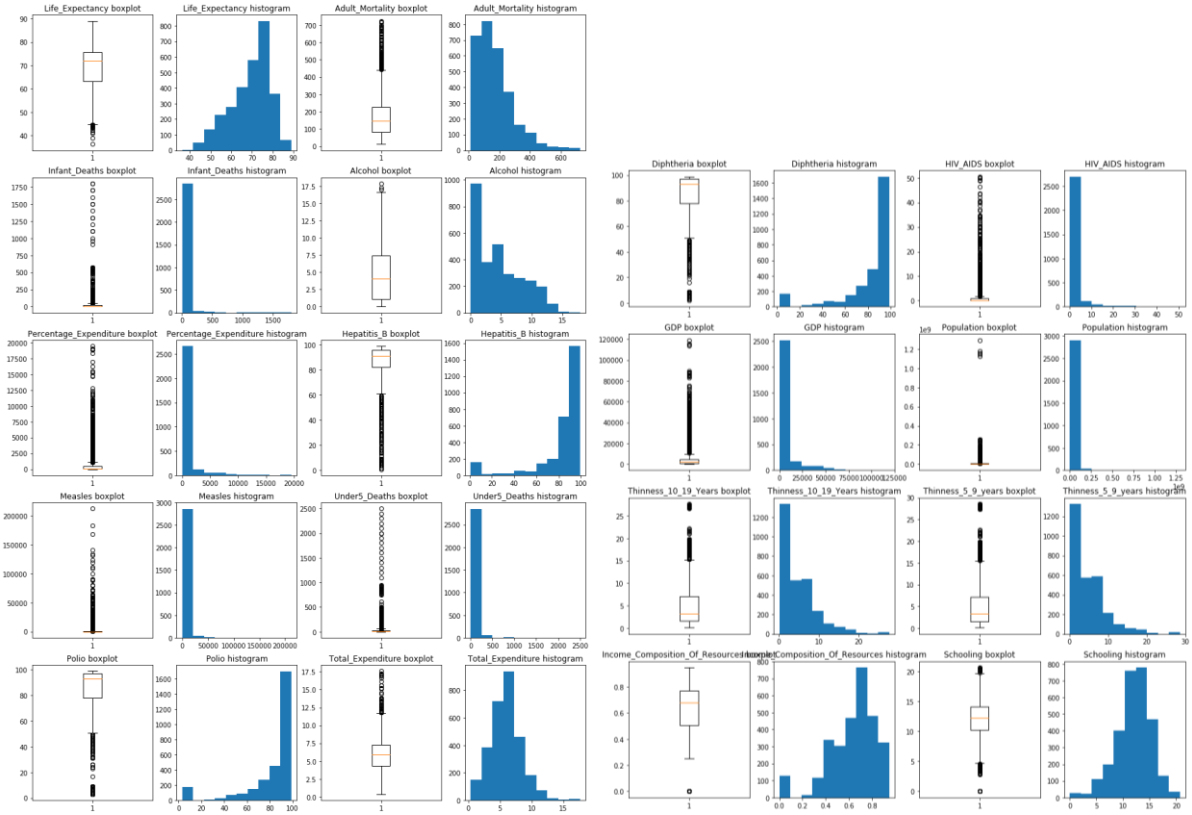




[부록2] 19개 변수에 존재하는 결측치 개수와 비율

	결측치 개수	결측치 비율
Life_Expectancy	10	0.34%
Adult_Mortality	155	5.28%
Infant_Deaths	848	28.86%
Alcohol	194	6.6%
Percentage_Expenditure	0	0%
Hepatitis_B	553	18.82%
Measles	0	0%
BMI	1456	49.56%
Under5_Deaths	785	26.72%
Polio	19	0.65%
Total_Expenditure	226	7.69%
Diphtheria	19	0.65%
HIV_AIDS	0	0%
GDP	448	15.25%
Population	652	22.19%
Thinnes_10_19_years	34	1.16%
Thinness_5_9_years	34	1.16%
Income_Composition_Of_Resources	167	5.68%
Schooling	163	5.55%

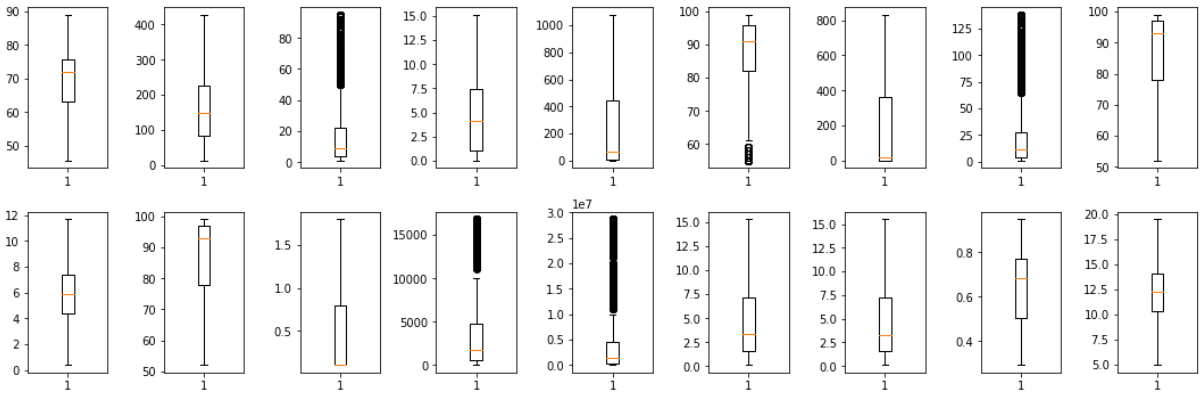
[부록3] 18개 각 변수의 이상치 존재를 파악하기 위한 boxplot과 histogram



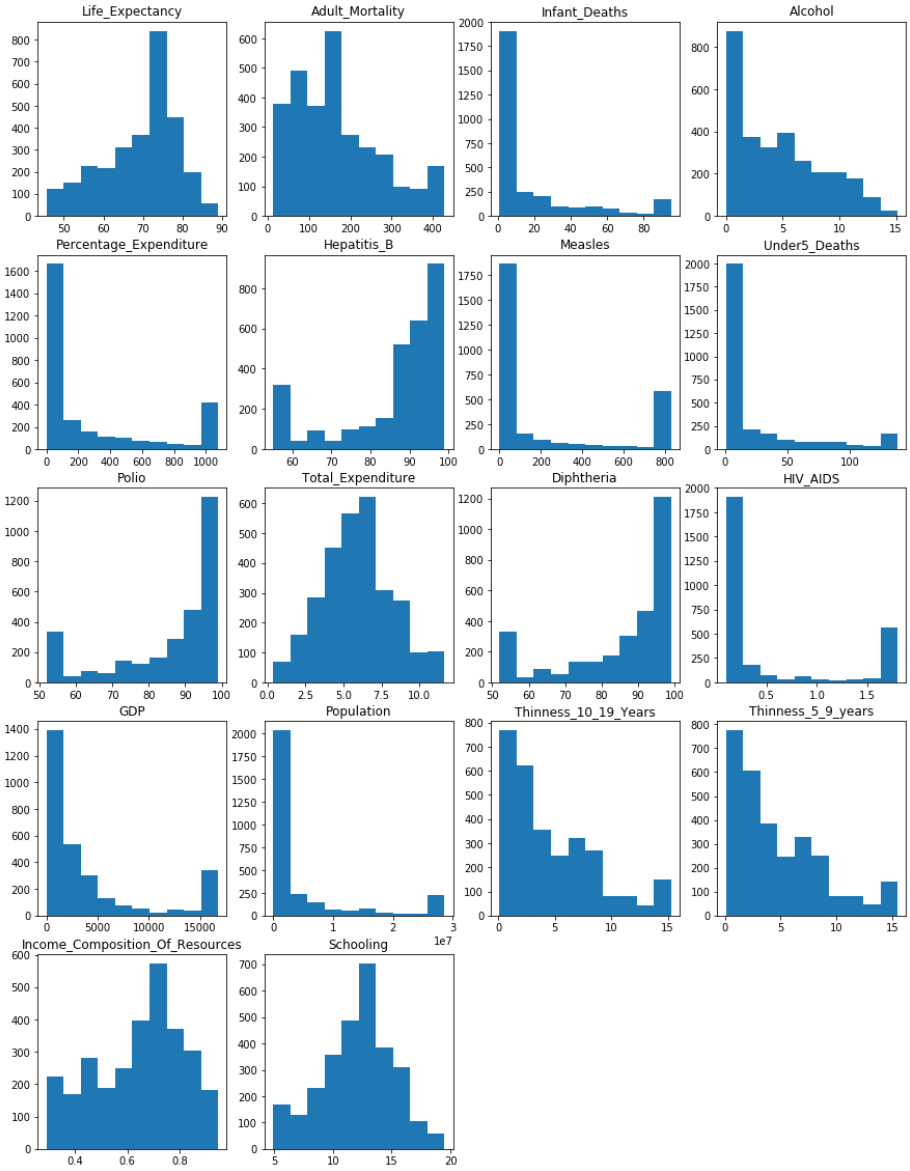
[부록4] 18개 각 변수의 이상치 개수와 비율

	이상치 개수	이상치 비율
Life_Expectancy	1	0.03%
Adult_Mortality	50	1.7%
Infant_Deaths	290	9.87%
Alcohol	0	0%
Percentage_Expenditure	349	11.88%
Hepatitis_B	289	9.84%
Measles	513	17.46%
Under5_Deaths	371	12.63%
Polio	202	6.88%
Total_Expenditure	26	0.88%
Diphtheria	230	7.83%
HIV_AIDS	474	16.13%%
GDP	383	13.04%
Population	404	13.75%
Thinnes_10_19_years	53	1.8%
Thinness_5_9_years	55	1.87%
Income_Composition_Of_Resources	0	0%
Schooling	28	0.95%

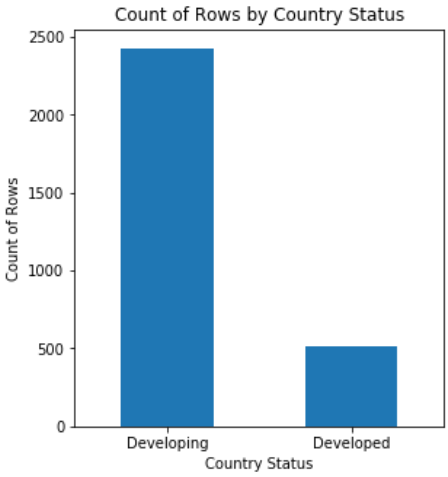
[부록5] 이상치가 처리된 변수들의 boxplot

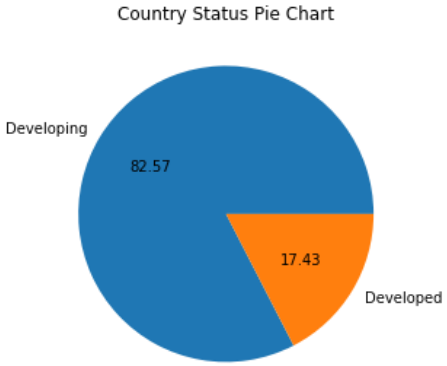


[부록6] 18개 각 변수의 히스토그램

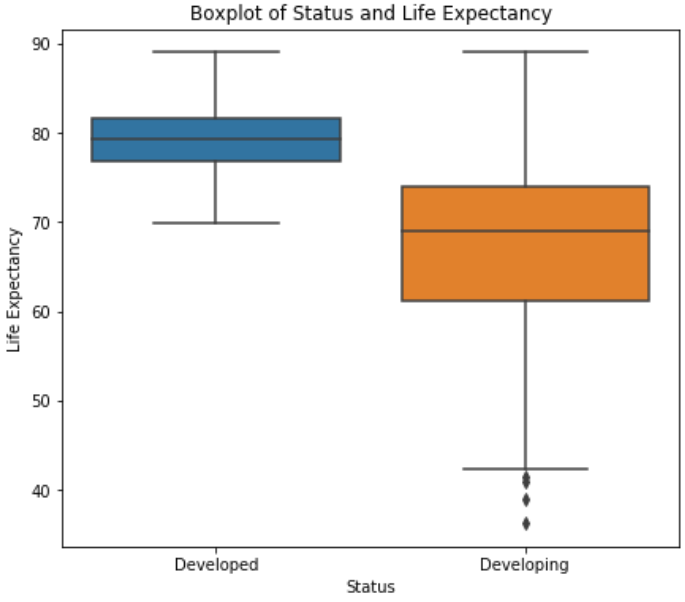


[부록7] Status 변수의 분포 비율 그래프

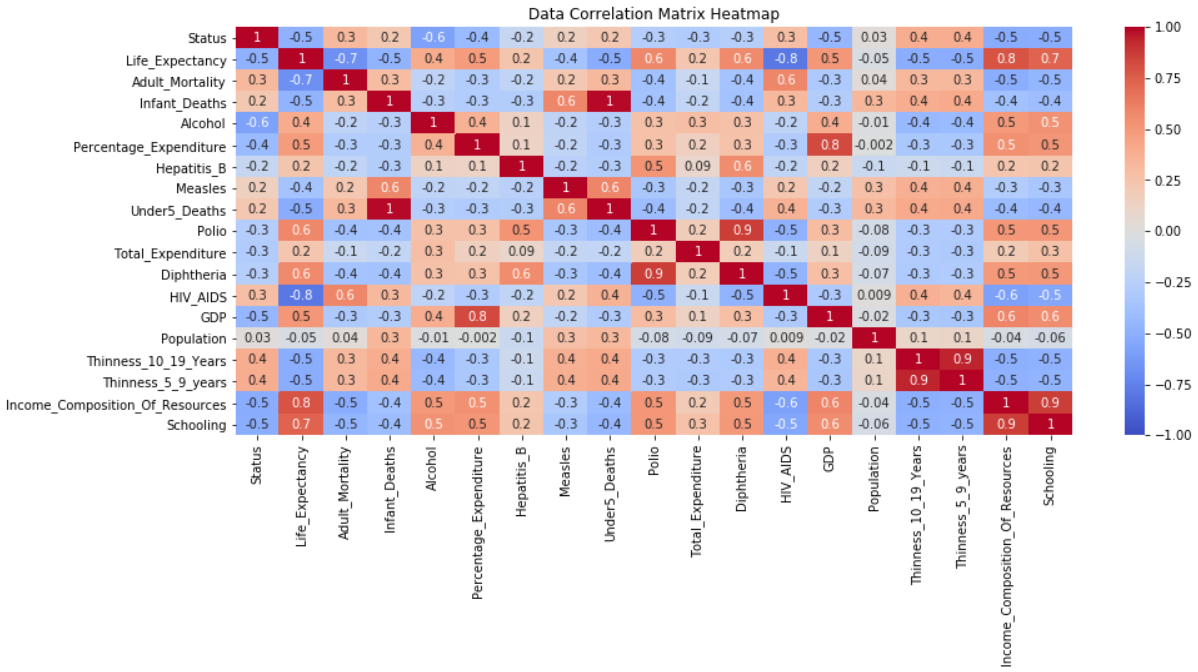




[부록8] 선진국과 개발도상국의 기대수명 차이



[부록9] 상관관계



[부록10] 전체데이터: 단순선형회귀 구축 결과 summary

OLS Regression Results

Dep. Variable:	Life_Expectancy	**R-squared:**	0.628
Model:	OLS	**Adj. R-squared:**	0.628
Method:	Least Squares	**F-statistic:**	3470.
Date:	Thu, 18 Jun 2020	**Prob (F-statistic):**	0.00
Time:	16:18:28	**Log-Likelihood:**	-6542.2
No. Observations:	2056	**AIC:**	1.309e+04
Df Residuals:	2054	**BIC:**	1.310e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	75.1235	0.164	459.036	0.000	74.803	75.444
HIV_AIDS	-11.2000	0.190	-58.905	0.000	-11.573	-10.827

Omnibus:	5.627	**Durbin-Watson:**	1.967
Prob(Omnibus):	0.060	**Jarque-Bera (JB):**	6.696
Skew:	-0.007	**Prob(JB):**	0.0352
Kurtosis:	3.279	**Cond. No.**	2.10

[부록11] 전체데이터: 변수 추가 및 제거 과정

단계	변수	Adj. R-squared	AIC	BIC
1단계	Status	0.701	1.264e+04	1.266e+04
2단계	Income_Composition_Of_Resources	0.799	1.182 e+04	1.185 e+04
3단계	Adult_Mortality	0.824	1.155 e+04	1.158 e+04
4단계	Status:Adult_Mortality	0.824	1.155 e+04	1.158 e+04
5단계	Diphtheria	0.835	1.143 e+04	1.146 e+04
6단계	Infant_Deaths	0.844	1.130 e+04	1.134 e+04
7단계	GDP	0.846	1.128 e+04	1.133 e+04
8단계	Thinness_5_9_years	0.849	1.124 e+04	1.130 e+04
9단계	Status:Thinness_5_9_years	0.852	1.120 e+04	1.126 e+04
10단계	Status:Adult_Mortality (제거)	0.852	1.120 e+04	1.126 e+04
11단계	Measles	0.853	1.119 e+04	1.125 e+04
12단계	Status:Measles	0.854	1.118 e+04	1.125 e+04

[부록12] 전체데이터: 다중선형회귀 구축 결과 summary

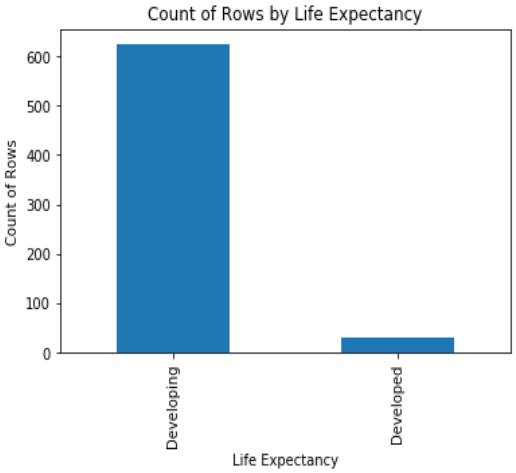
OLS Regression Results

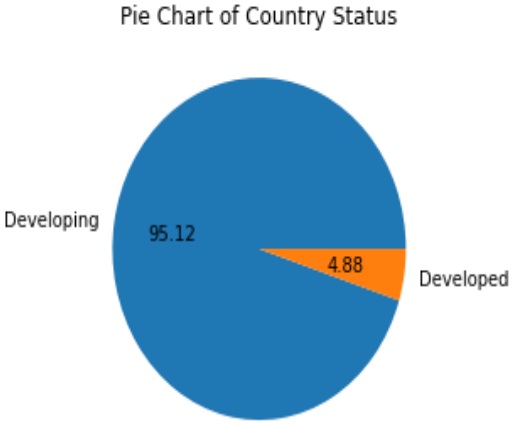
Dep. Variable:	Life_Expectancy	**R-squared:**	0.854
Model:	OLS	**Adj. R-squared:**	0.854
Method:	Least Squares	**F-statistic:**	1091.
Date:	Fri, 12 Jun 2020	**Prob (F-statistic):**	0.00
Time:	16:38:43	**Log-Likelihood:**	-5578.0
No. Observations:	2056	**AIC:**	1.118e+04
Df Residuals:	2044	**BIC:**	1.125e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	63.6379	0.879	72.389	0.000	61.914	65.362
HIV_AIDS	-5.4020	0.167	-32.255	0.000	-5.730	-5.074
Status	-3.9478	0.455	-8.679	0.000	-4.840	-3.056
Income_Composition_Of_Resources	16.3900	0.774	21.173	0.000	14.872	17.908
Adult_Mortality	-0.0154	0.001	-15.943	0.000	-0.017	-0.013
Diphtheria	0.0602	0.007	9.246	0.000	0.047	0.073
Infant_Deaths	-0.0275	0.004	-6.506	0.000	-0.036	-0.019
GDP	7.073e-05	1.97e-05	3.593	0.000	3.21e-05	0.000
Thinness_5_9_years	-1.8869	0.254	-7.433	0.000	-2.385	-1.389
Measles	0.0021	0.001	2.454	0.014	0.000	0.004
Status:Thinness_5_9_years	1.7738	0.255	6.965	0.000	1.274	2.273
Status:Measles	-0.0035	0.001	-3.779	0.000	-0.005	-0.002

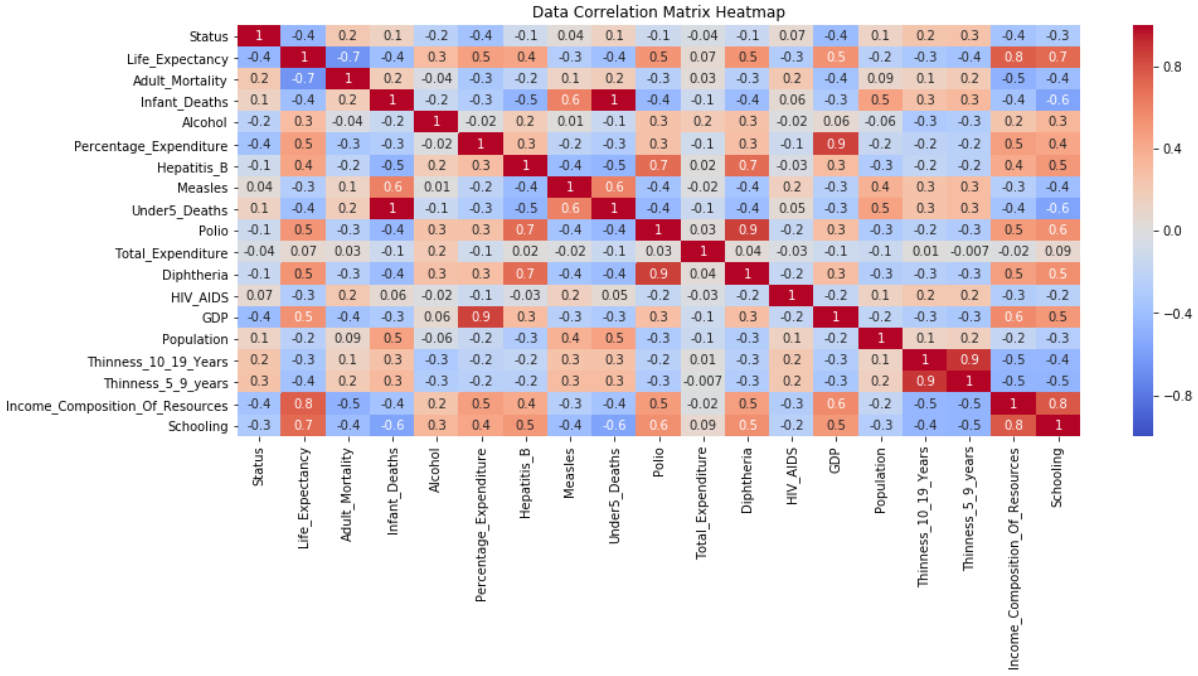
Omnibus:	59.214	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	145.193
Skew:	0.038	**Prob(JB):**	2.96e-32
Kurtosis:	4.300	**Cond. No.**	8.82e+04

[부록13] 아시아 국가의 Status 변수 분포 비율과 기대수명 차이





[부록14] 상관관계



[부록15] 아시아데이터: 단순선형회귀 구축 결과 summary

OLS Regression Results

Dep. Variable:	Life_Expectancy	**R-squared:**	0.509
Model:	OLS	**Adj. R-squared:**	0.508
Method:	Least Squares	**F-statistic:**	678.6
Date:	Fri, 19 Jun 2020	**Prob (F-statistic):**	3.58e-103
Time:	19:31:29	**Log-Likelihood:**	-1836.5
No. Observations:	656	**AIC:**	3677.
Df Residuals:	654	**BIC:**	3686.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	48.2450	0.877	54.995	0.000	46.522	49.968
Schooling	1.9274	0.074	26.050	0.000	1.782	2.073

Omnibus:	1.749	**Durbin-Watson:**	2.072
Prob(Omnibus):	0.417	**Jarque-Bera (JB):**	1.568
Skew:	0.096	**Prob(JB):**	0.457
Kurtosis:	3.143	**Cond. No.**	67.3

[부록16] 아시아데이터: 변수 추가 및 제거 과정

단계	변수	Adj.R-squared	AIC	BIC
1단계	Adult_Mortality	0.665	3426	3440
2단계	GDP	0.679	3398	3416
3단계	Polio	0.699	3359	3382
4단계	Income_Composition_Of_Resources	0.726	3299	3326
5단계	Diphtheria	0.730	3290	3321
6단계	Polio (제거)	0.730	3289	3316
7단계	Percentage_Expenditure (제거)	0.729	3290	3322
8단계	Status	0.756	3224	3255
9단계	GDP (제거)	0.756	3223	3249
10단계	Thinness_5_9_years (제거)	0.756	3223	3254
11단계	Thinness_10_19_Years (제거)	0.756	3221	3253
12단계	Infant_Deaths (제거)	0.756	3224	3255
13단계	Alcohol	0.762	3207	3238
14단계	HIV_AIDS	0.764	3203	3239
15단계	Measles (제거)	0.763	3204	3245
16단계	Status:Schooling (제거)	0.763	3205	3245
17단계	Status:Adult_Mortality (제거)	0.763	3204	3245
18단계	Status:Income (제거)	0.763	3205	3245
19단계	Status:Diphtheria (제거)	0.763	3205	3245
20단계	Status:Alcohol (제거)	0.763	3204	3245
21단계	Status:HIV_AIDS (제거)	0.764	3203	3239

20단계에서 추가된 교호작용항 Status:HIV_AIDS은 기준통계량을 개선시켰으나, F-test 결과에서 교호작용항이 추가되지 않은 모델이 더욱 효과적이라는 결론이 나왔기 때문에 추가하지 않는다.

[부록17] 아시아 데이터 다중선형회귀모델 구축 결과 summary

OLS Regression Results

Dep. Variable:	Life_Expectancy	**R-squared:**	0.766
Model:	OLS	**Adj. R-squared:**	0.764
Method:	Least Squares	**F-statistic:**	303.2
Date:	Thu, 18 Jun 2020	**Prob (F-statistic):**	1.14e-199
Time:	15:27:37	**Log-Likelihood:**	-1593.4
No. Observations:	656	**AIC:**	3203.
Df Residuals:	648	**BIC:**	3239.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	60.5934	1.175	51.587	0.000	58.287	62.900
Schooling	0.4796	0.088	5.438	0.000	0.306	0.653
Adult_Mortality	-0.0281	0.002	-16.256	0.000	-0.031	-0.025
Income_Composition_Of_Resources	10.6569	1.310	8.136	0.000	8.085	13.229
Diphtheria	0.0667	0.010	6.946	0.000	0.048	0.086
Status	-4.5740	0.547	-8.355	0.000	-5.649	-3.499
Alcohol	0.2193	0.051	4.339	0.000	0.120	0.319
HIV_AIDS	-1.4520	0.593	-2.447	0.015	-2.617	-0.287

Omnibus:	29.230	**Durbin-Watson:**	2.009
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	84.289
Skew:	0.019	**Prob(JB):**	4.98e-19
Kurtosis:	4.756	**Cond. No.**	2.40e+03

[부록18] 분석 코드

```
# 데이터 불러오기
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.stats.mstats import winsorize
import statsmodels.formula.api as smf
import os

life_data = pd.read_csv('Life Expectancy Data.csv')
life_data.info()

life_data.columns = ['Country', 'Year', 'Status', 'Life_Expectancy', 'Adult_Mortality',
                     'Infant_Deaths', 'Alcohol', 'Percentage_Expenditure', 'Hepatitis_B',
                     'Measles', 'BMI', 'Under5_Deaths', 'Polio', 'Total_Expenditure',
                     'Diphtheria', 'HIV_AIDS', 'GDP', 'Population', 'Thinness_10_19_Years',
                      'Thinness_5_9_years', 'Income_Composition_Of_Resources', 'Schooling']

life_data['Country']=life_data['Country'].astype('category')
life_data['Status']=life_data['Status'].astype('category')
life_data.head()

#EDA
life_data.head()
life_data.tail()

plt.figure(figsize=(15, 25))
life_data.Country.value_counts(ascending=True).plot(kind='barh')
```

```
plt.title('Count of Rows by Country')
plt.xlabel('Count of Rows')
plt.ylabel('Country')
plt.title("Plot of Country")
plt.tight_layout()
plt.show()

life_data.Year.value_counts().sort_index().plot(kind='barh')
plt.title('Count of Rows by Year')
plt.xlabel('Count of Rows')
plt.ylabel('Year')
plt.title("Counts of Rows by Year")
plt.show()

life_data.Status.value_counts().plot(kind='pie', autopct='%.2f')
plt.ylabel('')
plt.title('Pie Chart of Country Status')

plot = sns.boxplot(x='Status',y='Life_Expectancy',data=life_data)
plot.set_xlabel("Status")
plot.set_ylabel("Life Expectancy")
plot.set_title("Boxplot of Status and Life Expectancy")

life_data["Life_Expectancy"].describe()

sns.boxplot(y="Life_Expectancy",data=life_data)
plt.title("Boxplot of Life Expectancy")

plt.hist(life_data["Life_Expectancy"],bins=20)
plt.title('Count of Rows by Life Expectancy')
```

```
plt.xlabel('Life Expectancy')
plt.ylabel('Count of Rows')

life_data["Adult_Mortality"].describe()

sns.boxplot(y="Adult_Mortality",data=life_data)
plt.title("Boxplot of Adult Mortality")

plt.hist(life_data["Adult_Mortality"],bins=15)
plt.title('Count of Rows by Adult Mortality')
plt.xlabel('Adult_Mortality')
plt.ylabel('Count of Rows')

life_data["Infant_Deaths"].describe()

sns.boxplot(y="Infant_Deaths",data=life_data)
plt.title("Boxplot of Infant Deaths")

plt.hist(life_data["Infant_Deaths"],bins=9)
plt.title('Count of Rows by Infant Deaths')
plt.xlabel('Infant Deaths')
plt.ylabel('Count of Rows')

life_data["Alcohol"].describe()

sns.boxplot(y="Alcohol",data=life_data)
plt.title("Boxplot of Alcohol")

plt.hist(life_data["Alcohol"],bins=15)
plt.title('Count of Rows by Alcohol')
```

```
plt.xlabel('Alcohol')
plt.ylabel('Count of Rows')

life_data["Percentage_Expenditure"].describe()

sns.boxplot(y="Percentage_Expenditure",data=life_data)
plt.title("Boxplot of Percentage_Expenditure")

plt.hist(life_data["Percentage_Expenditure"])
plt.title('Count of Rows by Percentage Expenditure')
plt.xlabel('Percentage Expenditure')
plt.ylabel('Count of Rows')

sns.boxplot(x="Status",y="Percentage_Expenditure",data=life_data)
plt.title("Boxplot of Percentage_Expenditure and Status")

life_data["Hepatitis_B"].describe()

sns.boxplot(y="Hepatitis_B",data=life_data)
plt.title("Boxplot of Hepatitis_B")

plt.hist(life_data["Hepatitis_B"],bins=10)
plt.title('Count of Rows by Hepatitis_B')
plt.xlabel('Hepatitis_B')
plt.ylabel('Count of Rows')

life_data["Measles"].describe()

sns.boxplot(y="Measles",data=life_data)
plt.title("Boxplot of Measles")
```

```
plt.hist(life_data["Measles"],bins=15)
plt.title('Count of Rows by Measles')
plt.xlabel('Measles')
plt.ylabel('Count of Rows')

sns.boxplot(x="Status",y="Measles",data=life_data)
plt.title("Boxplot of Status and Measles")

life_data["BMI"].describe()

sns.boxplot(y="BMI",data=life_data)
plt.title("Boxplot of BMI")

plt.hist(life_data["BMI"],bins=10)
plt.title('Count of Rows by BMI')
plt.xlabel('BMI')
plt.ylabel('Count of Rows')

life_data["Under5_Deaths"].describe()

sns.boxplot(y="Under5_Deaths",data=life_data)
plt.title("Boxplot of Under5_Deaths")

plt.hist(life_data["Under5_Deaths"],bins=50)
plt.title('Count of Rows by Under5_Deaths')
plt.xlabel('Under5_Deaths')
plt.ylabel('Count of Rows')

life_data["Polio"].describe()
```

```
sns.boxplot(y="Polio",data=life_data)
plt.title("Boxplot of Polio")

plt.hist(life_data["Polio"],bins=15)
plt.title('Count of Rows by Polio')
plt.xlabel('Polio')
plt.ylabel('Count of Rows')

sns.boxplot(x="Status",y="Polio",data=life_data)
plt.title("Boxplot of Status and Polio")

life_data["Total_Expenditure"].describe()

plt.hist(life_data["Total_Expenditure"],bins=20)
plt.title('Count of Rows by Total Expenditure')
plt.xlabel('Total Expenditure')
plt.ylabel('Count of Rows')

sns.boxplot(y="Total_Expenditure",data=life_data)
plt.title("Boxplot of Total Expenditure")

life_data["Diphtheria"].describe()

sns.boxplot(y="Diphtheria",data=life_data)
plt.title("Boxplot of Diphtheira")

plt.hist(life_data["Diphtheria"],bins=15)
plt.title('Count of Rows by Diphtheria')
plt.xlabel('Diphtheria')
plt.ylabel('Count of Rows')
```

```
life_data["HIV_AIDS"].describe()

sns.boxplot(y="HIV_AIDS",data=life_data)
plt.title("Boxplot of HIV_AIDS")

plt.hist(life_data["HIV_AIDS"],bins=15)
plt.title('Count of Rows by HIV_AIDS')
plt.xlabel('HIV_AIDS')
plt.ylabel('Count of Rows')

sns.boxplot(x="Status",y="HIV_AIDS",data=life_data)
plt.title("Boxplot of Status and HIV_AIDS")

life_data["GDP"].describe()

sns.boxplot(y="GDP",data=life_data)
plt.title("Boxplot of GDP")

plt.hist(life_data["GDP"],bins=20)
plt.title('Count of Rows by GDP')
plt.xlabel('GDP')
plt.ylabel('Count of Rows')

sns.boxplot(x='Status',y="GDP",data=life_data)
plt.title("Boxplot of Status and GDP")

life_data["Population"].describe()

sns.boxplot(y="Population",data=life_data)
plt.title("Boxplot of Population")
```

```
plt.hist(life_data["Population"]/1000000000,bins=10)
plt.title('Count of Rows by Population/1000000000')
plt.xlabel('Population')
plt.ylabel('Count of Rows')

life_data["Thinness_10_19_Years"].describe()

sns.boxplot(y="Thinness_10_19_Years",data=life_data)
plt.title("Boxplot of Thinness_10_19_Years")

plt.hist(life_data["Thinness_10_19_Years"],bins=20)
plt.title('Count of Rows by Thinness_10_19_Years')
plt.xlabel('Thinness_10_19_Years')
plt.ylabel('Count of Rows')

life_data['Thinness_5_9_years'].describe()

sns.boxplot(y="Thinness_5_9_years",data=life_data)
plt.title("Boxplot of Thinness_5_9_years")

plt.hist(life_data["Thinness_5_9_years"],bins=15)
plt.title('Count of Rows by Thinness_5_9_years')
plt.xlabel('Thinness_5_9_Years')
plt.ylabel('Count of Rows')

life_data["Income_Composition_Of_Resources"].describe()

sns.boxplot(y="Income_Composition_Of_Resources",data=life_data)
plt.title("Boxplot of Income_Composition_Of_Resources")
```

```
plt.hist(life_data["Income_Composition_Of_Resources"],bins=25)
plt.title('Count of Rows by Income_Composition_Of_Resources')
plt.xlabel('Income_Composition_Of_Resources')
plt.ylabel('Count of Rows')

life_data['Schooling'].describe()

sns.boxplot(y="Schooling",data=life_data)
plt.title("Boxplot of Schooling")

plt.hist(life_data["Schooling"],bins=15)
plt.title('Count of Rows by Schooling')
plt.xlabel('Schooling')
plt.ylabel('Count of Rows')

# 전처리
life_data.describe().iloc[:, 1:]

# 이상치 확인후, 이상치를 결측으로 처리한 뒤 결측값대체
plt.figure(figsize=(15,10))
for i, col in enumerate(['Adult_Mortality', 'Infant_Deaths', 'BMI', 'Under5_Deaths', 'GDP', 'Population'], start=1):
    plt.subplot(2, 3, i)
    life_data.boxplot(col)

mort_5_percentile = np.percentile(life_data.Adult_Mortality.dropna(), 5)
life_data.Adult_Mortality = life_data.apply(lambda x: np.nan if x.Adult_Mortality < mort_5_percentile else x.Adult_Mortality, axis=1)
life_data.Infant_Deaths = life_data.Infant_Deaths.replace(0, np.nan)
life_data.BMI = life_data.apply(lambda x: np.nan if (x.BMI < 10 or x.BMI > 50) else x.BMI, axis=1)
life_data['Under5_Deaths'] = life_data['Under5_Deaths'].replace(0, np.nan)
```

```
# 결측 분포 파악
plt.figure(figsize=(12,12))
sns.heatmap(life_data.isnull(),cbar=False)

# 각 열의 결측 개수와 비율 출력 함수
def null_column_percentage(df):
    df_cols = list(df.columns)
    cols_total_count = len(list(df.columns))
    cols_count = 0
    for loc, col in enumerate(df_cols):
        null_count = df[col].isnull().sum()
        total_count = df[col].isnull().count()
        percent_null = round(null_count/total_count*100, 2)
        if null_count > 0:
            cols_count += 1
            print('[iloc = {}] {} has {} null values: {}% null'.format(loc, col, null_count, percent_null))
    cols_percent_null = round(cols_count/cols_total_count*100, 2)
    print('Out of {} total columns, {} contain null values; {}% columns contain null
values.'.format(cols_total_count, cols_count, cols_percent_null))

null_column_percentage(life_data)
life_data.drop(columns='BMI', inplace=True)

# 연도별로 각 열의 중앙값으로 결측치 대체 함수(시계열 데이터 특성)
life_impute = []     # 결측치가 대체된 데이터를 저장하는 리스트

for Year in list(life_data.Year.unique()):
    year_data = life_data[life_data.Year == Year].copy()
    for col in list(year_data.columns)[3:]:
        year_data[col] = year_data[col].fillna(year_data[col].dropna().median()).copy()
    life_impute.append(year_data)
```

```
life_data = pd.concat(life_impute).copy()
null_column_percentage(life_data)

plt.figure(figsize=(12,12))
sns.heatmap(life_data.isnull(),cbar=False)

# 나라명, 연도, 국가 상태 변수는 제외 => 설명변수, 반응변수 역할을 할 수 있는 변수만 남김
var = list(life_data.columns)[3:]

# 각 변수의 boxplot과 histogram 그래프로 이상치 존재 파악 함수
def outliers_graph(df):
    plt.figure(figsize=(15, 40))
    i = 0
    for col in var:
        i += 1
        plt.subplot(9, 4, i)
        plt.boxplot(df[col])
        plt.title('{} boxplot'.format(col))
        i += 1
        plt.subplot(9, 4, i)
        plt.hist(df[col])
        plt.title('{} histogram'.format(col))
    plt.show()
outliers_graph(life_data)

# 2*iqr을 기준으로 각 열의 이상치 개수와 비율 출력 함수
def outlier_number_percent(col, df):
    print(12*'-' + col + 12*'-')
    pct75, pct25 = np.percentile(df[col], [75, 25])
    iqr = pct75 - pct25
```

```
    min_value = pct25 - (iqr*2)
    max_value = pct75 + (iqr*2)
    count = len(np.where((df[col] > max_value) | (df[col] < min_value))[0])
    percent = round(count/len(df[col])*100, 2)
    print('Number of outliers: {}'.format(count))
    print('Percent outlier: {}%'.format(percent))

for col in var:
    outlier_number_percent(col,life_data)

# winsorize 방법으로 이상치 처리
life_wins_dict = {}       # 이상치가 처리된 데이터를 저장하는 딕셔너리

def winsorize_graph(df, col, lower_limit=0, upper_limit=0, show_plot=True):
    life_wins = winsorize(df[col], limits=(lower_limit, upper_limit))
    life_wins_dict[col] = life_wins
    if show_plot == True:
        plt.figure(figsize=(15,5))
        plt.subplot(121)
        plt.boxplot(df[col])
        plt.title('original {}'.format(col))
        plt.subplot(122)
        plt.boxplot(life_wins)
        plt.title('wins=({},{}) {}'.format(lower_limit, upper_limit, col))
        plt.show()

# 이상치 처리된 변수들 boxplot 그래프
plt.figure(figsize=(15,5))
for i, col in enumerate(var, 1):
    plt.subplot(2, 9, i)
```

```
    plt.boxplot(life_wins_dict[col])
plt.tight_layout()
plt.show()

life_wins_data = life_data.iloc[:,0:3]
for col in var:
    life_wins_data[col] = life_wins_dict[col]

# 변수들 히스토그램
plt.figure(figsize=(15, 20))
for i, col in enumerate(var, 1):
    plt.subplot(5, 4, i)
    plt.hist(life_wins_data[col])
    plt.title(col)

plt.figure(figsize=(15, 25))
life_wins_data.Country.value_counts(ascending=True).plot(kind='barh')
plt.title('Count of Rows by Country')
plt.xlabel('Count of Rows')
plt.ylabel('Country')
plt.tight_layout()
plt.show()

life_wins_data.Year.value_counts().sort_index().plot(kind='barh')
plt.title('Count of Rows by Year')
plt.xlabel('Count of Rows')
plt.ylabel('Year')
plt.show()

plt.figure(figsize=(10, 5))
```

```
plt.subplot(121)
life_wins_data.Status.value_counts().plot(kind='bar')
plt.title('Count of Rows by Country Status')
plt.xlabel('Country Status')
plt.ylabel('Count of Rows')
plt.xticks(rotation=0)

plt.subplot(122)
life_wins_data.Status.value_counts().plot(kind='pie', autopct='%.2f')
plt.ylabel('')
plt.title('Country Status Pie Chart')

plt.show()

# Status에 따른 T-test
life_wins_data.groupby('Status').Life_Expectancy.agg(['mean'])

plt.figure(figsize=(7,6))
plot = sns.boxplot(x='Status',y='Life_Expectancy',data=life_data)
plot.set_xlabel("Status")
plot.set_ylabel("Life Expectancy")
plot.set_title("Boxplot of Status and Life Expectancy")

developed_le = life_wins_data[life_wins_data.Status == 'Developed'].Life_Expectancy
developing_le = life_wins_data[life_wins_data.Status == 'Developing'].Life_Expectancy
stats.ttest_ind(developed_le, developing_le, equal_var=False)

# developed와 developing으로 모델 분류
developed_data = life_wins_data[life_wins_data.Status == 'Developed']
developing_data = life_wins_data[life_wins_data.Status == 'Developing']
```

```
mask = np.triu(life_wins_data[var].corr())

plt.figure(figsize=(15,6))

sns.heatmap(life_wins_data[var].corr(), annot=True, fmt='.1g', vmin=-1, vmax=1, center=0,
cmap='coolwarm', mask=mask)

plt.ylim(18, 0)

plt.title('Data Correlation Matrix Heatmap')

plt.show()


mask = np.triu(developing_data[var].corr())

plt.figure(figsize=(15,6))

sns.heatmap(developing_data[var].corr(), annot=True, fmt='.1g', vmin=-1, vmax=1, center=0,
cmap='coolwarm', mask=mask)

plt.ylim(18, 0)

plt.title('Developing Data Correlation Matrix Heatmap')

plt.show()


mask = np.triu(developed_data[var].corr())

plt.figure(figsize=(15,6))

sns.heatmap(developed_data[var].corr(), annot=True, fmt='.1g', vmin=-1, vmax=1, center=0,
cmap='coolwarm', mask=mask)

plt.ylim(18, 0)

plt.title('Developed Data Correlation Matrix Heatmap')

plt.show()


################# 회귀모델링

from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

life_wins_data['Status'] = encoder.fit_transform(life_wins_data['Status'])


var2 = list(life_data.columns)[2:]
```

```
# train/test set 분리 : 7대3

from sklearn.model_selection import train_test_split

life_train , life_test = train_test_split(life_wins_data, test_size = 0.3, train_size=0.7,random_state=1234,
shuffle = True, stratify = life_wins_data.Status)

life_train.info()


import csv

write =   open(life_train.csv','w',encoding='utf-8',newline='')

wr = csv.writer(write)

wr.writerow(life_train)

for i in range(1,2056):

      wr.writerow(life_train.iloc[i,:])

write.close()


plt.figure(figsize=(15,6))

sns.heatmap(life_train[var2].corr(), annot=True, fmt='.1g', vmin=-1, vmax=1, center=0,
cmap='coolwarm')

plt.title('Data Correlation Matrix Heatmap')

plt.show()


##### 단순선형회귀

# 상수항만 존재하는 모델

result_intercept = smf.ols('Life_Expectancy~1', data=life_train).fit()

result_intercept.summary()


# 단순선형회귀 적합

result_slr = smf.ols('Life_Expectancy ~ HIV_AIDS', data=life_train).fit()

result_slr.summary()


# 단순선형회귀 시각화

plt.figure(figsize=(13,7))
```

```
plt.scatter(life_train['HIV_AIDS'], life_train['Life_Expectancy'], marker='o', label='realtrain')

plt.scatter(life_train['HIV_AIDS'], result_slr.fittedvalues, marker='.', label='fitted')

plt.plot(life_train['HIV_AIDS'], result_slr.predict(life_train['HIV_AIDS']), color='blue', linestyle='dashed',
label='regression', markersize=0)


pred_slr = result_slr.get_prediction().summary_frame()


plt.plot(life_train['HIV_AIDS'], pred_slr['mean_ci_lower'], 'r-.', label='means 95% CI', linewidth=1)

plt.plot(life_train['HIV_AIDS'], pred_slr['mean_ci_upper'], 'r-.', linewidth=1)

plt.plot(life_train['HIV_AIDS'], pred_slr['obs_ci_lower'], 'g-.', label='obs 95% CI', linewidth=1)

plt.plot(life_train['HIV_AIDS'], pred_slr['obs_ci_upper'], 'g-.', linewidth=1)

plt.xlabel('HIV_AIDS')

plt.legend()


# 선형성 (상관계수 = -0.8)

plt.scatter('HIV_AIDS','Life_Expectancy',data=life_train)


# 정규성

resid_slr = result_slr.resid

stats.probplot(resid_slr,plot=plt)

plt.show()


# 등분산성

fitted_slr = result_slr.predict(life_train)

sns.regplot(fitted_slr, stats.zscore(resid_slr),lowess = True, line_kws={'color' : 'red'})

plt.xlim(55.0, 72)

plt.show()


## mlr1 : Status 추가

result_mlr1 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status', data=life_train).fit()
```

```
result_mlr1.summary()

# mlr1_Scatter plot

y = life_test['Life_Expectancy']

y_hat1 = result_mlr1.predict(life_test[['HIV_AIDS' , 'Status']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat1, marker='o')

x = np.arange(50, 90)

plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat1')

plt.title('Scatter Plot of mlr1')


# mlr2 : Income 변수 추가

result_mlr2 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources',
data=life_train).fit()

result_mlr2.summary()


# mlr2_Scatter Plot

y = life_test['Life_Expectancy']

y_hat2 = result_mlr2.predict(life_test[['HIV_AIDS' , 'Status','Income_Composition_Of_Resources']])


# 시각화

plt.figure(figsize=(13,7))


plt.scatter(y, y_hat2, marker='o')

x = np.arange(50, 90)

plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat2')
```

```
plt.title('Scatter Plot of mlr2')

# mlr3 : Adult_Mortality 변수 추가

result_mlr3 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources + Adult_Mortality', data=life_train).fit()

result_mlr3.summary()


# mlr3_Scatter Plot

y = life_test['Life_Expectancy']

y_hat3 = result_mlr3.predict(life_test[['HIV_AIDS' , 'Status','Income_Composition_Of_Resources','Adult_Mortality']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat3, marker='o')

x = np.arange(50, 90)

plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat3')

plt.title('Scatter Plot of mlr3')


# Status:Adult_Mortality 추가

result_mlr31 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources + Adult_Mortality + Status:Adult_Mortality', data=life_train).fit()

result_mlr31.summary()


# F test

anova_lm(result_mlr3, result_mlr31, typ=1)


# VIF확인

y, X = dmatrices('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources + Adult_Mortality', life_train, return_type = 'dataframe')
```

```
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif["features"] = X.columns
vif


# mlr31_Scatter Plot
y = life_test['Life_Expectancy']
y_hat31 = result_mlr31.predict(life_test[['HIV_AIDS' ,
'Status','Income_Composition_Of_Resources','Adult_Mortality']])


# 시각화
plt.figure(figsize=(13,7))
plt.scatter(y, y_hat31, marker='o')
x = np.arange(50, 90)
plt.plot(x,x, 'red')
plt.xlabel('Life Expectancy')
plt.ylabel('Y_hat31')
plt.title('Scatter Plot of mlr31')


# mlr4 : Diphtheria 변수 추가
result_mlr4 = smf.ols(
    'Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources + Adult_Mortality +
Diphtheria + Status:Adult_Mortality',
    data=life_train).fit()
result_mlr4.summary()


# VIF지수 확인
y, X = dmatrices('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria', life_train, return_type = 'dataframe')
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
```

```
vif["features"] = X.columns

# mlr4_Scatter Plot

y = life_test['Life_Expectancy']

y_hat4 = result_mlr4.predict(life_test[['HIV_AIDS' ,
'Status','Income_Composition_Of_Resources','Adult_Mortality','Diphtheria']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat4, marker='o')

x = np.arange(50, 90)

plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat4')

plt.title('Scatter Plot of mlr4')


# mlr5 : Infant_Deaths 변수 추가

result_mlr5 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths', data=life_train).fit()

result_mlr5.summary()


# VIF확인

y, X = dmatrices('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths', life_train, return_type = 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr5_Scatter Plot

y = life_test['Life_Expectancy']

y_hat5 = result_mlr5.predict(life_test[['HIV_AIDS' ,
'Status','Income_Composition_Of_Resources','Adult_Mortality', 'Diphtheria','Infant_Deaths']])
```

```
# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat5, marker='o')

x = np.arange(50, 90)

plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat5')

plt.title('Scatter Plot of mlr5')


# mlr6 : GDP 변수 추가

result_mlr6 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Status:Adult_Mortality', data=life_train).fit()

result_mlr6.summary()


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP', life_train, return_type = 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr6_Scatter Plot

y = life_test['Life_Expectancy']

y_hat6 = result_mlr6.predict(life_test[['HIV_AIDS' ,
'Status','Income_Composition_Of_Resources','Adult_Mortality', 'Diphtheria','Infant_Deaths','GDP']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat6, marker='o')

x = np.arange(50, 90)
```

```
plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat6')

plt.title('Scatter Plot of mlr6')


# mlr7 : Thinness_5_9_years 변수 추가

result_mlr7 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Status:Adult_Mortality',
data=life_train).fit()

result_mlr7.summary()


# Status : Thinness_5_9_years 추가

result_mlr71 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Status:Adult_Mortality +
Status:Thinness_5_9_years', data=life_train).fit()

result_mlr71.summary()


# 기존에 존재하던 Status:Adult_Mortality 제거

result_mlr72 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Status:Thinness_5_9_years',
data=life_train).fit()

result_mlr72.summary()


# F test

anova_lm(result_mlr72, result_mlr71, typ=1)


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years', life_train, return_type =
'dataframe')


vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
```

```
vif["features"] = X.columns


# mlr72_Scatter Plot

y = life_test['Life_Expectancy']

y_hat72 = result_mlr72.predict(life_test[['HIV_AIDS' ,
'Status','Income_Composition_Of_Resources','Adult_Mortality',
'Diphtheria','Infant_Deaths','GDP','Thinness_5_9_years']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat72, marker='o')

x = np.arange(50, 90)

plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat72')

plt.title('Scatter Plot of mlr72')


# mlr8 : Measles 변수 추가

result_mlr8 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Measles +
Status:Thinness_5_9_years', data=life_train).fit()

result_mlr8.summary()


# Status:Measles 추가

result_mlr81 = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Measles +
Status:Thinness_5_9_years + Status:Measles', data=life_train).fit()

result_mlr81.summary()


# F test

anova_lm(result_mlr8, result_mlr81, typ=1)
```

```
# VIF 확인

y, X = dmatrices('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Measles', life_train, return_type
= 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr81_Scatter Plot

y = life_test['Life_Expectancy']

y_hat81 = result_mlr81.predict(life_test[['HIV_AIDS' ,
'Status','Income_Composition_Of_Resources','Adult_Mortality',
'Diphtheria','Infant_Deaths','GDP','Thinness_5_9_years','Measles']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat81, marker='o')

x = np.arange(50, 90)

plt.plot(x,x, 'red')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat81')

plt.title('Scatter Plot of mlr81')


# 최종 회귀 모형

result = smf.ols('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Measles +
Status:Thinness_5_9_years + Status:Measles', data=life_train).fit()

result.summary()


# 가정확인

final_var = ['Life_Expectancy', 'HIV_AIDS', 'Status', 'Income_Composition_Of_Resources',
```

```
                'Adult_Mortality', 'Diphtheria', 'Infant_Deaths', 'GDP',
               'Thinness_5_9_years', 'Measles']

# 선형성
plt.figure(figsize=(15,6))
sns.heatmap(life_train[final_var].corr(), annot=True, fmt='.1g', vmin=-1, vmax=1, center=0,
cmap='coolwarm')
plt.title('Data Correlation Matrix Heatmap')
plt.show()

# 다중공선성
y, X = dmatrices('Life_Expectancy ~ HIV_AIDS + Status + Income_Composition_Of_Resources +
Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years + Measles', life_train, return_type
= 'dataframe')

vif = pd.DataFrame()
vif["features"] = X.columns
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif

# 정규성
resid = result.resid
stats.probplot(resid,plot=plt)
plt.show()

# 등분산성
fitted = result.predict(life_train)
sns.regplot(fitted, np.sqrt(np.abs(stats.zscore(resid))),lowess = True, line_kws={'color' : 'red'})
plt.xlim(55.0, 71.50)
plt.show()
```

```
# 회귀 모형 예측력
y = life_test['Life_Expectancy']
y_hat = result.predict(life_test[['HIV_AIDS' , 'Status','Income_Composition_Of_Resources','Adult_Mortality',

'Diphtheria','Infant_Deaths','GDP','Thinness_5_9_years','Measles']])

# 시각화
plt.figure(figsize=(13,7))
plt.scatter(y, y_hat, marker='o')
x = np.arange(50, 90)
plt.plot(x,x, 'red')
plt.xlabel('test y')
plt.ylabel('train Y_hat')
plt.title('Scatter Plot of Life_expectancy')

X_test = life_test.copy()
del X_test['Life_Expectancy']
mse = mean_squared_error(y_true = life_test['Life_Expectancy'], y_pred = result.predict(X_test))
print('mse =', mse)

rmse= np.sqrt(mse)
print("rmse =",rmse)

y_true = life_test["Life_Expectancy"]
y_pred = result.predict(X_test)

def MAE(y_true, y_pred):
    print ("mae =", np.mean(np.abs((y_true - y_pred))))

MAE(y_true, y_pred)
```

```
########################## 아시아 데이터


# 아시아 데이터 생성
Asia = ["Oman", "Nepal","Jordan","Republic of Korea", "Uzbekistan","Timor-Leste", "Iraq","Laos",
        "Iran (Islamic Republic of)","Lebanon","Malaysia","India",
"Maldives","Indonesia","Mongolia","Japan","Myanmar",
        "Democratic People's Republic of Korea", "Bahrain", "China", "Bangladesh", "Kazakhstan",
        "Viet Nam", "Qatar", "Bhutan", "Cambodia", "Brunei Darussalam", "Kuwait", "Saudi Arabia" ,
"Kyrgyzstan",
        "Sri Lanka", "Thailand", 'Syrian Arab Republic', "Taiwan", "Singapore", "Tajikistan", "Arab
Emirates", "Turkey",
        "Turkmenistan", "Azerbaijan", "Pakistan", "Afghanistan", "Philippines", "Yemen"]
asia_data = life_wins_data[life_wins_data['Country'].isin(Asia)]


import csv
write =   open('asia.csv','w',encoding='utf-8',newline='')
wr = csv.writer(write)
wr.writerow(asia_data)
for i in range(1,656):
    wr.writerow(asia_data.iloc[i,:])
write.close()


# Status에 따른 차이
asia_data.Status.value_counts().plot(kind='pie', autopct='%.2f')
plt.ylabel('')
plt.title('Pie Chart of Country Status')


asia_data.Status.value_counts().plot(kind='bar')
plt.title('Count of Rows by Life Expectancy')
plt.xlabel('Life Expectancy')
plt.ylabel('Count of Rows')
```

```
asia_data.groupby('Status').Life_Expectancy.agg(['mean'])

developed_le = asia_data[asia_data.Status == 'Developed'].Life_Expectancy
developing_le = asia_data[asia_data.Status == 'Developing'].Life_Expectancy
stats.ttest_ind(developed_le, developing_le, equal_var=False)

###################### 회귀 모델링
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()
asia_data['Status'] = encoder.fit_transform(asia_data['Status'])

var2 = list(asia_data.columns)[2:]

asia_data.info()

plt.figure(figsize=(15,6))
sns.heatmap(asia_data[var2].corr(), annot=True, fmt='.1g', vmin=-1, vmax=1, center=0,
cmap='coolwarm')
#plt.ylim(18, 0)
plt.title('Data Correlation Matrix Heatmap')
plt.show()

# 상수항만 있는 모형
result_intercept = smf.ols('Life_Expectancy ~ 1', data=asia_data).fit()
result_intercept.summary()

# 단순선형회귀 적합
result_slr = smf.ols('Life_Expectancy ~ Schooling', data=asia_data).fit()
result_slr.summary()
```

```
# 단순선형회귀 시각화
plt.figure(figsize=(13,7))
plt.scatter(asia_data['Schooling'], asia_data['Life_Expectancy'], marker='o', label='realtrain')
plt.scatter(asia_data['Schooling'], result_slr.fittedvalues, marker='.', label='fitted')
plt.plot(asia_data['Schooling'], result_slr.predict(asia_data['Schooling']), color='blue', linestyle='dashed',
label='regression', markersize=0)

pred_slr = result_slr.get_prediction().summary_frame()
plt.plot(asia_data['Schooling'], pred_slr['mean_ci_lower'], 'r-.', label='means 95% CI', linewidth=1)
plt.plot(asia_data['Schooling'], pred_slr['mean_ci_upper'], 'r-.', linewidth=1)
plt.plot(asia_data['Schooling'], pred_slr['obs_ci_lower'], 'g-.', label='obs 95% CI', linewidth=1)
plt.plot(asia_data['Schooling'], pred_slr['obs_ci_upper'], 'g-.', linewidth=1)
plt.xlabel('Schooling')
plt.legend()

# 선형성 (상관계수 = 0.7)
plt.scatter('Schooling','Life_Expectancy',data=asia_data)

# 정규성
resid_slr = result_slr.resid
stats.probplot(resid_slr,plot=plt)
plt.show()

# 등분산성
fitted_slr = result_slr.predict(asia_data)
sns.regplot(fitted_slr, stats.zscore(resid_slr),lowess = True, line_kws={'color' : 'red'})
#plt.xlim(55.0, 71.50)
plt.show()

# mlr1 : Adult_Mortality 변수 추가
result_mlr1 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality', data=asia_data).fit()
```

```
result_mlr1.summary()


# VIF 확인
y, X = dmatrices('Life_Expectancy ~ + Schooling + Adult_Mortality', asia_data, return_type = 'dataframe')
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif["features"] = X.columns
vif


# mlr1_Scatter Plot
y = asia_data['Life_Expectancy']
y_hat1 = result_mlr1.predict(asia_data[['Schooling','Adult_Mortality']])


# 시각화
plt.figure(figsize=(13,7))
plt.scatter(y, y_hat1, marker='o', label='Scatter plot of mlr1')
plt.xlabel('Life Expectancy')
plt.ylabel('Y_hat1')


# mlr2 : GDP 변수 추가
result_mlr2 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + GDP ', data=asia_data).fit()
result_mlr2.summary()


# VIF 확인
y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + GDP', asia_data, return_type =
'dataframe')
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif["features"] = X.columns
vif
```

```
# mlr2_Scatter Plot
y = asia_data['Life_Expectancy']
y_hat2 = result_mlr2.predict(asia_data[['Schooling','Adult_Mortality','GDP']])

# 시각화
plt.figure(figsize=(13,7))
plt.scatter(y, y_hat2, marker='o')
plt.xlabel('Life Expectancy')
plt.ylabel('Y_hat2')
plt.title('Scatter Plot of mlr2')

# mlr3 : Polio 변수 추가
result_mlr3 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + GDP + Polio', data=asia_data).fit()
result_mlr3.summary()

# VIF 확인
y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + GDP + Polio', asia_data, return_type =
'dataframe')
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif["features"] = X.columns
vif

# mlr3_Scatter Plot
y = asia_data['Life_Expectancy']
y_hat3 = result_mlr3.predict(asia_data[['Schooling','Adult_Mortality','GDP','Polio']])

# 시각화
plt.figure(figsize=(13,7))
plt.scatter(y, y_hat3, marker='o')
plt.xlabel('Life Expectancy')
```

```
plt.ylabel('Y_hat3')

plt.title('Scatter Plot of mlr3')


# mlr4 : Income 변수 추가

result_mlr4 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + GDP + Polio +
Income_Composition_Of_Resources', data=asia_data).fit()

result_mlr4.summary()


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + GDP + Polio +
Income_Composition_Of_Resources', asia_data, return_type = 'dataframe')


vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr4_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat4 =
result_mlr4.predict(asia_data[['Schooling','Adult_Mortality','GDP','Polio','Income_Composition_Of_Resourc
es']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat4, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y')

plt.title('Scatter Plot of mlr4')


# mlr5 : Diphtheria 변수 추가

result_mlr5 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + GDP + Polio +
```

```
Income_Composition_Of_Resources + Diphtheria', data=asia_data).fit()

result_mlr5.summary()


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + GDP + Polio +
Income_Composition_Of_Resources + Diphtheria', asia_data , return_type = 'dataframe')


vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr5_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat5 =
result_mlr5.predict(asia_data[['Schooling','Adult_Mortality','GDP','Polio','Income_Composition_Of_Resourc
es','Diphtheria']])


# 시각화

plt.figure(figsize=(13,7))


plt.scatter(y, y_hat5, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat5')

plt.title('Scatter Plot of mlr5')


# mlr6:Polio변수제거

result_mlr6 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + GDP +
Income_Composition_Of_Resources + Diphtheria', data=asia_data).fit()

result_mlr6.summary()


# VIF 확인
```

```
y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + GDP +
Income_Composition_Of_Resources + Diphtheria', asia_data, return_type = 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr6_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat6 =
result_mlr6.predict(asia_data[['Schooling','Adult_Mortality','GDP','Income_Composition_Of_Resources','Di
phtheria']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat6, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat6')

plt.title('Scatter Plot of mlr6')


# mlr7 : Percentage_Expenditure 변수 추가

result_mlr7 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + GDP +
Income_Composition_Of_Resources + Diphtheria + Percentage_Expenditure', data=asia_data).fit()

result_mlr7.summary()


# mlr7_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat7 =
result_mlr7.predict(asia_data[['Schooling','Adult_Mortality','GDP','Income_Composition_Of_Resources','Di
phtheria','Percentage_Expenditure']])


# 시각화
```

```
plt.figure(figsize=(13,7))

plt.scatter(y, y_hat7, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat7')

plt.title('Scatter Plot of mlr7')


# mlr8 : Status 변수 추가

result_mlr8 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + GDP +
Income_Composition_Of_Resources + Diphtheria + Status', data=asia_data).fit()

result_mlr8.summary()


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + GDP +
Income_Composition_Of_Resources + Diphtheria + Status', asia_data, return_type = 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr8 _ Scatter Plot

y = asia_data['Life_Expectancy']

y_hat8 =
result_mlr8.predict(asia_data[['Schooling','Adult_Mortality','GDP','Income_Composition_Of_Resources',
'Diphtheria','Status']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat8, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat8')

plt.title('Scatter Plot of mlr8')
```

```
# mlr9 : GDP변수 제거

result_mlr9 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources + Diphtheria + Status', data=asia_data).fit()

result_mlr9.summary()


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources + Diphtheria + Status', asia_data, return_type = 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr9_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat9 = result_mlr9.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphtheria','Status']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat9, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat9')

plt.title('Scatter Plot of mlr9')


# mlr10 : Thinness_5_9_years 변수 추가

result_mlr10 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources + Diphtheria + Status + Thinness_5_9_years ', data=asia_data).fit()

result_mlr10.summary()


# mlr10_Scatter Plot
```

```
y = asia_data['Life_Expectancy']

y_hat10 =
result_mlr10.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphtheria','Status', 'Thinness_5_9_years']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat10, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat10')

plt.title('Scatter Plot of mlr10')


# mlr11 : Thinness_10_19_Years 변수 추가

result_mlr11 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Thinness_10_19_Years ', data=asia_data).fit()

result_mlr11.summary()


# mlr11_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat11 =
result_mlr11.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphtheria','Status', 'Thinness_10_19_Years']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat11, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat11')

plt.title('Scatter Plot of mlr11')


# mlr12 : Infant_Deaths 변수 추가

result_mlr12 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
```

```
Income_Composition_Of_Resources + Diphtheria + Status + Infant_Deaths', data=asia_data).fit()

result_mlr12.summary()


# mlr12_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat12 =
result_mlr12.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphther
ia','Status', 'Infant_Deaths']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat12, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat12')

plt.title('Scatter Plot of mlr12')


# mlr13 : Alcohol 변수 추가

result_mlr13 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol', data=asia_data).fit()

result_mlr13.summary()


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources +
Diphtheria + Status + Alcohol', asia_data, return_type = 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr13_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat13 =
```

```
result_mlr13.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphtheria','Status', 'Alcohol']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat13, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat13')

plt.title('Scatter Plot of mlr13')


# mlr14 : HIV_AIDS 변수 추가

result_mlr14 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS', data=asia_data).fit()

result_mlr14.summary()


# VIF 확인

y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS', asia_data, return_type = 'dataframe')

vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

vif


# mlr14_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat14 = result_mlr14.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphtheria','Status', 'Alcohol','HIV_AIDS']])


# 시각화

plt.figure(figsize=(13,7))
```

```
plt.scatter(y, y_hat14, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat14')

plt.title('Scatter Plot of mlr14')


# mlr15 : Measles 변수 추가

result_mlr15 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS + Measles',
data=asia_data).fit()

result_mlr15.summary()


# mlr15_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat15 =
result_mlr15.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphther
ia','Status', 'Alcohol','HIV_AIDS','Measles']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat15, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat15')

plt.title('Scatter Plot of mlr15')


# mlr16 : Status:Schooling 교호작용 확인

result_mlr16 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS + Status:Schooling',
data=asia_data).fit()

result_mlr16.summary()


# mlr16_Scatter Plot

y = asia_data['Life_Expectancy']
```

```
y_hat16 =
result_mlr16.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphtheria','Status', 'Alcohol','HIV_AIDS']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat16, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat16')

plt.title('Scatter Plot of mlr15')


# mlr17 : Status:Adult_Mortality 교호작용 확인

result_mlr17 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS + Status:Adult_Mortality',
data=asia_data).fit()

result_mlr17.summary()


# mlr17_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat17 =
result_mlr17.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphtheria','Status', 'Alcohol','HIV_AIDS']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat17, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat17')

plt.title('Scatter Plot of mlr17')


# mlr18 : Status:Income 교호작용확인

result_mlr18 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS +
```

```
Status:Income_Composition_Of_Resources', data=asia_data).fit()

result_mlr18.summary()


# mlr18 _Scatter Plot

y = asia_data['Life_Expectancy']

y_hat18 =
result_mlr18.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphther
ia','Status', 'Alcohol','HIV_AIDS']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat18, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat18')

plt.title('Scatter Plot of mlr18')


# mlr19 : Status : Diphtheria 교호작용 확인

result_mlr19 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS + Status:Diphtheria',
data=asia_data).fit()

result_mlr19.summary()


# mlr19_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat19 =
result_mlr19.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphther
ia','Status', 'Alcohol','HIV_AIDS']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat19, marker='o')

plt.xlabel('Life Expectancy')
```

```
plt.ylabel('Y_hat19')

plt.title('Scatter Plot of mlr19')


# mlr20 : Status:Alcohol 교호작용 확인

result_mlr20 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS + Status:Alcohol',
data=asia_data).fit()

result_mlr20.summary()


# mlr20 _ Scatter Plot

y = asia_data['Life_Expectancy']

y_hat20 =
result_mlr20.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphther
ia','Status', 'Alcohol','HIV_AIDS']])


# 시각화

plt.figure(figsize=(13,7))

plt.scatter(y, y_hat20, marker='o')

plt.xlabel('Life Expectancy')

plt.ylabel('Y_hat20')

plt.title('Scatter Plot of mlr20')


# mlr21 : Status:HIV_AIDS 교호작용 확인

result_mlr21 = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality +
Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS + Status:HIV_AIDS',
data=asia_data).fit()

result_mlr21.summary()


# mlr21_Scatter Plot

y = asia_data['Life_Expectancy']

y_hat21 =
result_mlr21.predict(asia_data[['Schooling','Adult_Mortality','Income_Composition_Of_Resources','Diphther
ia','Status', 'Alcohol','HIV_AIDS']])
```

```
# 시각화
plt.figure(figsize=(13,7))
plt.scatter(y, y_hat21, marker='o')
plt.plot(y,y,'r')
plt.xlabel('Life Expectancy')
plt.ylabel('Y_hat21')
plt.title('Scatter Plot of mlr21')

# F test
anova_lm(result_mlr14, result_mlr21, typ=1)

# 최종모형 선택
result = smf.ols('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS ', data=asia_data).fit()
result.summary()

# 가정확인
final_var = ['Life_Expectancy', 'Schooling', 'Adult_Mortality',
             'Income_Composition_Of_Resources', 'Diphtheria', 'Status',
             'Alcohol', 'HIV_AIDS']

# 선형성
plt.figure(figsize=(15,6))
sns.heatmap(asia_data[final_var].corr(), annot=True, fmt='.1g', vmin=-1, vmax=1, center=0, cmap='coolwarm')
plt.title('Data Correlation Matrix Heatmap')
plt.show()

# 다중공선성
y, X = dmatrices('Life_Expectancy ~ Schooling + Adult_Mortality + Income_Composition_Of_Resources + Diphtheria + Status + Alcohol + HIV_AIDS', asia_data, return_type = 'dataframe')
```

```
vif = pd.DataFrame()
vif["features"] = X.columns
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif

# 정규성
resid = result.resid

stats.probplot(resid,plot=plt)
plt.show()

# 등분산성
fitted = result.predict(asia_data)
sns.regplot(fitted,stats.zscore(resid),lowess = True, line_kws={'color' : 'red'})
plt.show()

############# R을 통한 표준화계수 확인
library(QuantPsyc)
asia <- read.csv('asia.csv')
result1 <- lm(Life_Expectancy ~ Schooling + Adult_Mortality+ Income_Composition_Of_Resources+
                    Diphtheria + Status + Alcohol + HIV_AIDS , data=asia)
lm.beta(result1)

life_train <- read.csv('life_train.csv')
result2 <- lm(Life_Expectancy ~ HIV_AIDS +Status + Income_Composition_Of_Resources+
                                Adult_Mortality + Diphtheria + Infant_Deaths + GDP + Thinness_5_9_years+
                                Measles+Status*Thinness_5_9_years,data=life_train)
lm.beta(result2)
```

```
############# R을 통한 잔차의 독립성 확인
result1_2 <- lm(Life_Expectancy ~ HIV_AIDS,data=life_train)
result2_2 <- lm(Life_Expectancy~Schooling, data=asia)

dwtest(result1)
dwtest(result1_2)
dwtest(result2)
dwtest(result2_2)

############# R을 통한 아시아국가 랜덤포레스트
library("randomForest")
set.seed(1234)
asia_rf = randomForest(Life_Expectancy ~ Schooling + Adult_Mortality +
                                        Income_Composition_Of_Resources + Diphtheria+
                                        Status + Alcohol + HIV_AIDS, data = asia, importance = T)
importance(asia_rf)
varImpPlot(asia_rf, main="varImpPlot of asia")
```