

Education Statistics Dataset

by Pittawat Taveekitworachai 61130500220

Data Source: [World Bank](#)

Setup

Import required dependencies

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.dates import DateFormatter
import matplotlib.dates as mdates
```

General config

```
In [ ]: sns.set_style("darkgrid")
```

```
In [ ]: from google.colab import drive
drive.mount('/gdrive')
```

Mounted at /gdrive

```
In [ ]: prefix_path = '/gdrive/MyDrive/Colab Notebooks/CSC536/'
```

Import dataset and Data Examination

```
In [ ]: country = pd.read_csv(f'{prefix_path}country.csv')
education = pd.read_csv(f'{prefix_path}education.csv')
```

```
In [ ]: country.head()
```

Out[]:

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	In (
0	ABW	Aruba	Aruba	Aruba	AW	Aruban florin	SNA data for 2000-2011 are updated from offici...	Latin America & Caribbean	in non
1	AFG	Afghanistan	Afghanistan	Islamic State of Afghanistan	AF	Afghan afghani	Fiscal year end: March 20; reporting period fo...	South Asia	ir
2	AGO	Angola	Angola	People's Republic of Angola	AO	Angolan kwanza	April 2013 database update: Based on IMF data,...	Sub-Saharan Africa	r ir
3	ALB	Albania	Albania	Republic of Albania	AL	Albanian lek	NaN	Europe & Central Asia	r ir
4	AND	Andorra	Andorra	Principality of Andorra	AD	Euro	NaN	Europe & Central Asia	in non

In []:

```
education.head()
```

Out[]:	Country Name	Country Code	Indicator Name	Indicator Code	1970	1971	1972	197
0	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	Na
1	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	Na
2	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN	NaN	Na
3	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	Na
4	Arab World	ARB	Adjusted net enrolment rate, primary, both sex...	SE.PRM.TENR	54.822121	54.894138	56.209438	57.26710

```

In [ ]: country.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241 entries, 0 to 240
Data columns (total 32 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Country Code                          241 non-null    obj
1   Short Name                            241 non-null    obj
2   Table Name                            241 non-null    obj
3   Long Name                             241 non-null    obj
4   2-alpha code                          238 non-null    obj
5   Currency Unit                          215 non-null    obj
6   Special Notes                          145 non-null    obj
7   Region                                214 non-null    obj

```

```

8   Income Group                214 non-null    obj
ect
9   WB-2 code                   240 non-null    obj
ect
10  National accounts base year  205 non-null    obj
ect
11  National accounts reference year  32 non-null     flo
at64
12  SNA price valuation         197 non-null    obj
ect
13  Lending category           144 non-null    obj
ect
14  Other groups                58 non-null     obj
ect
15  System of National Accounts  215 non-null    obj
ect
16  Alternative conversion factor  47 non-null     obj
ect
17  PPP survey year            145 non-null    obj
ect
18  Balance of Payments Manual in use  181 non-null    obj
ect
19  External debt Reporting status  124 non-null    obj
ect
20  System of trade            200 non-null    obj
ect
21  Government Accounting concept  161 non-null    obj
ect
22  IMF data dissemination standard  181 non-null    obj
ect
23  Latest population census      213 non-null    obj
ect
24  Latest household survey      141 non-null    obj
ect
25  Source of most recent Income and expenditure data  160 non-null    obj
ect
26  Vital registration complete    111 non-null    obj
ect
27  Latest agricultural census    142 non-null    obj
ect
28  Latest industrial data        107 non-null    flo
at64
29  Latest trade data            185 non-null    flo
at64
30  Latest water withdrawal data   179 non-null    obj
ect
31  Unnamed: 31                  0 non-null      flo
at64
dtypes: float64(4), object(28)
memory usage: 60.4+ KB

```

In []:

```
education.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 886930 entries, 0 to 886929
Data columns (total 70 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Country Name          886930 non-null object
1   Country Code          886930 non-null object
2   Indicator Name        886930 non-null object
3   Indicator Code        886930 non-null object
4   1970                  72288 non-null float64
5   1971                  35537 non-null float64

```

6	1972	35619 non-null	float64
7	1973	35545 non-null	float64
8	1974	35730 non-null	float64
9	1975	87306 non-null	float64
10	1976	37483 non-null	float64
11	1977	37574 non-null	float64
12	1978	37576 non-null	float64
13	1979	36809 non-null	float64
14	1980	89122 non-null	float64
15	1981	38777 non-null	float64
16	1982	37511 non-null	float64
17	1983	38460 non-null	float64
18	1984	38606 non-null	float64
19	1985	90296 non-null	float64
20	1986	39372 non-null	float64
21	1987	38641 non-null	float64
22	1988	38552 non-null	float64
23	1989	37540 non-null	float64
24	1990	124405 non-null	float64
25	1991	74437 non-null	float64
26	1992	75543 non-null	float64
27	1993	75793 non-null	float64
28	1994	77462 non-null	float64
29	1995	131361 non-null	float64
30	1996	76807 non-null	float64
31	1997	73453 non-null	float64
32	1998	84914 non-null	float64
33	1999	118839 non-null	float64
34	2000	176676 non-null	float64
35	2001	123509 non-null	float64
36	2002	124205 non-null	float64
37	2003	130363 non-null	float64
38	2004	128814 non-null	float64
39	2005	184108 non-null	float64
40	2006	140312 non-null	float64
41	2007	137272 non-null	float64
42	2008	134387 non-null	float64
43	2009	142108 non-null	float64
44	2010	242442 non-null	float64
45	2011	146012 non-null	float64
46	2012	147264 non-null	float64
47	2013	137509 non-null	float64
48	2014	113789 non-null	float64
49	2015	131058 non-null	float64
50	2016	16460 non-null	float64
51	2017	143 non-null	float64
52	2020	51436 non-null	float64
53	2025	51436 non-null	float64
54	2030	51436 non-null	float64
55	2035	51436 non-null	float64
56	2040	51436 non-null	float64
57	2045	51436 non-null	float64
58	2050	51436 non-null	float64
59	2055	51436 non-null	float64
60	2060	51436 non-null	float64
61	2065	51436 non-null	float64
62	2070	51436 non-null	float64
63	2075	51436 non-null	float64
64	2080	51436 non-null	float64
65	2085	51436 non-null	float64
66	2090	51436 non-null	float64
67	2095	51436 non-null	float64
68	2100	51436 non-null	float64
69	Unnamed: 69	0 non-null	float64

dtypes: float64(66), object(4)

memory usage: 473.7+ MB

In []:

```
country.describe()
```

Out[]:

	National accounts reference year	Latest industrial data	Latest trade data	Unnamed: 31
count	32.00000	107.000000	185.000000	0.0
mean	2001.53125	2008.102804	2010.994595	NaN
std	5.24856	2.616834	2.569675	NaN
min	1987.00000	2000.000000	1995.000000	NaN
25%	1996.75000	2007.500000	2011.000000	NaN
50%	2002.00000	2009.000000	2012.000000	NaN
75%	2005.00000	2010.000000	2012.000000	NaN
max	2012.00000	2010.000000	2012.000000	NaN

In []:

```
education.describe()
```

Out[]:

	1970	1971	1972	1973	1974	
count	7.228800e+04	3.553700e+04	3.561900e+04	3.554500e+04	3.573000e+04	8.73
mean	1.974772e+09	4.253638e+09	4.592365e+09	5.105006e+09	5.401493e+09	2.31
std	1.211687e+11	1.804814e+11	1.914083e+11	2.059170e+11	2.112150e+11	1.37
min	-1.435564e+00	-1.594625e+00	-3.056522e+00	-4.032582e+00	-4.213563e+00	-3.65
25%	8.900000e-01	8.853210e+00	9.240920e+00	9.595200e+00	9.861595e+00	1.40
50%	6.317724e+00	6.316240e+01	6.655139e+01	6.969595e+01	7.087760e+01	9.67
75%	6.251250e+01	5.655200e+04	5.863650e+04	6.202900e+04	6.383675e+04	7.85
max	1.903929e+13	1.986457e+13	2.100916e+13	2.238367e+13	2.282991e+13	2.30

In []:

```
country.isnull().sum()
```

```
Out[ ]: Country Code      0
        Short Name       0
        Table Name       0
        Long Name        0
        2-alpha code     3
        Currency Unit    26
        Special Notes    96
        Region          27
        Income Group     27
        WB-2 code        1
        National accounts base year 36
        National accounts reference year 209
        SNA price valuation 44
        Lending category 97
        Other groups     183
        System of National Accounts 26
        Alternative conversion factor 194
        PPP survey year  96
        Balance of Payments Manual in use 60
        External debt Reporting status 117
        System of trade  41
        Government Accounting concept 80
        IMF data dissemination standard 60
        Latest population census 28
        Latest household survey 100
        Source of most recent Income and expenditure data 81
        Vital registration complete 130
        Latest agricultural census 99
        Latest industrial data 134
        Latest trade data 56
        Latest water withdrawal data 62
        Unnamed: 31      241
        dtype: int64
```

```
In [ ]: education.isnull().sum()
```

```
Out[ ]: Country Name      0
        Country Code     0
        Indicator Name    0
        Indicator Code    0
        1970              814642
        ...
        2085              835494
        2090              835494
        2095              835494
        2100              835494
        Unnamed: 69      886930
        Length: 70, dtype: int64
```

Data Preparation

```
In [ ]: country['Unnamed: 31'].unique()
```

```
Out[ ]: array([nan])
```

```
In [ ]: education['Unnamed: 69'].unique()
```

```
Out[ ]: array([nan])
```

```
In [ ]: country.drop('Unnamed: 31', axis='columns', inplace=True)
education.drop('Unnamed: 69', axis='columns', inplace=True)
```

```
In [ ]: education = education.melt(id_vars=['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code'],
    var_name='Year',
    value_name='Value')
```

```
In [ ]: education.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57650450 entries, 0 to 57650449
Data columns (total 6 columns):
#   Column          Dtype
---  -
0   Country Name    object
1   Country Code    object
2   Indicator Name  object
3   Indicator Code  object
4   Year            object
5   Value           float64
dtypes: float64(1), object(5)
memory usage: 2.6+ GB
```

```
In [ ]: education.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57650450 entries, 0 to 57650449
Data columns (total 6 columns):
#   Column          Dtype
---  -
0   Country Name    object
1   Country Code    object
2   Indicator Name  object
3   Indicator Code  object
4   Year            object
5   Value           float64
dtypes: float64(1), object(5)
memory usage: 2.6+ GB
```

```
In [ ]: education['Year'] = pd.to_datetime(education['Year'], format='%Y')
```

```
In [ ]: education.info()
```



```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57650450 entries, 0 to 57650449
Data columns (total 6 columns):
#   Column                Dtype
---  -
0   Country Name          object
1   Country Code          object
2   Indicator Name        object
3   Indicator Code        object
4   Year                  datetime64[ns]
5   Value                 float64
dtypes: datetime64[ns](1), float64(1), object(4)
memory usage: 2.6+ GB

```

```
In [ ]: education.head()
```

```
Out[ ]:
```

	Country Name	Country Code	Indicator Name	Indicator Code	Year	Value
0	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	1970-01-01	NaN
1	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	1970-01-01	NaN
2	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	1970-01-01	NaN
3	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	1970-01-01	NaN
4	Arab World	ARB	Adjusted net enrolment rate, primary, both sex...	SE.PRM.TENR	1970-01-01	54.822121

```
In [ ]: education = education.dropna()
```

```
In [ ]: education['Indicator Name'].unique()
```

```
Out[ ]: array(['Adjusted net enrolment rate, primary, both sexes (%)',
               'Adjusted net enrolment rate, primary, female (%)',
               'Adjusted net enrolment rate, primary, gender parity index (GPI)',
               ...,
               'SABER: (Tertiary Education) Policy Goal 6 Lever 2: Fostering RDI and Innovation',
               'SABER: (Tertiary Education) Policy Goal 6 Lever 3: Fostering Social and Cultural Development and Environmental Protection and Sustainability',
               'SABER: (Tertiary Education) Policy Goal 6: The Relevance of Tertiary Education for Economic and Social Needs'],
              dtype=object)
```

Data Visualization

GDP per capita, PPP (current international \$) Thailand vs World

```
In [ ]: gdp = education[(education['Indicator Name'] == 'GDP per capita, PPP (current international $)')]
```

```
In [ ]: gdp_th_world = gdp[(gdp['Country Name'] == 'Thailand') | (gdp['Country Name'] == 'World')]
gdp_th_world.head()
```

```
Out[ ]:
```

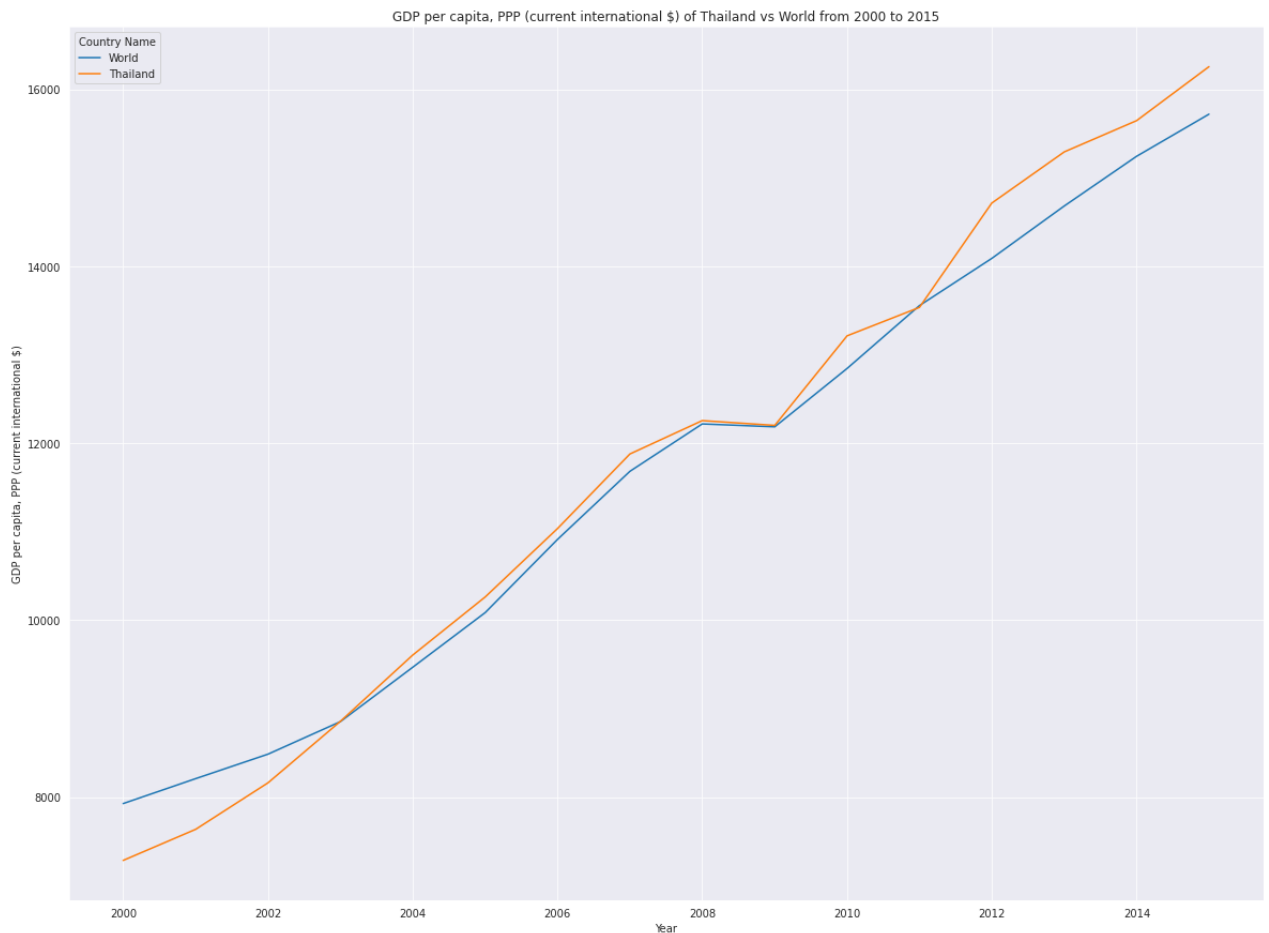
	Country Name	Country Code	Indicator Name	Indicator Code	Year	Value
26697107	World	WLD	GDP per capita, PPP (current international \$)	NY.GDP.PCAP.PP.CD	2000-01-01	7926.047904
27404452	Thailand	THA	GDP per capita, PPP (current international \$)	NY.GDP.PCAP.PP.CD	2000-01-01	7283.511290
27584037	World	WLD	GDP per capita, PPP (current international \$)	NY.GDP.PCAP.PP.CD	2001-01-01	8208.189373
28291382	Thailand	THA	GDP per capita, PPP (current international \$)	NY.GDP.PCAP.PP.CD	2001-01-01	7635.093003
28470967	World	WLD	GDP per capita, PPP (current international \$)	NY.GDP.PCAP.PP.CD	2002-01-01	8484.916473

```
In [ ]: plt.figure(figsize=(20,15))

ax = sns.lineplot(data=gdp_th_world, x="Year", y="Value", hue='Country Name')

ax.set_title("GDP per capita, PPP (current international $) of Thailand vs World")
ax.set_xlabel('Year')
ax.set_ylabel('GDP per capita, PPP (current international $)')
```

```
Out[ ]: Text(0, 0.5, 'GDP per capita, PPP (current international $)')
```



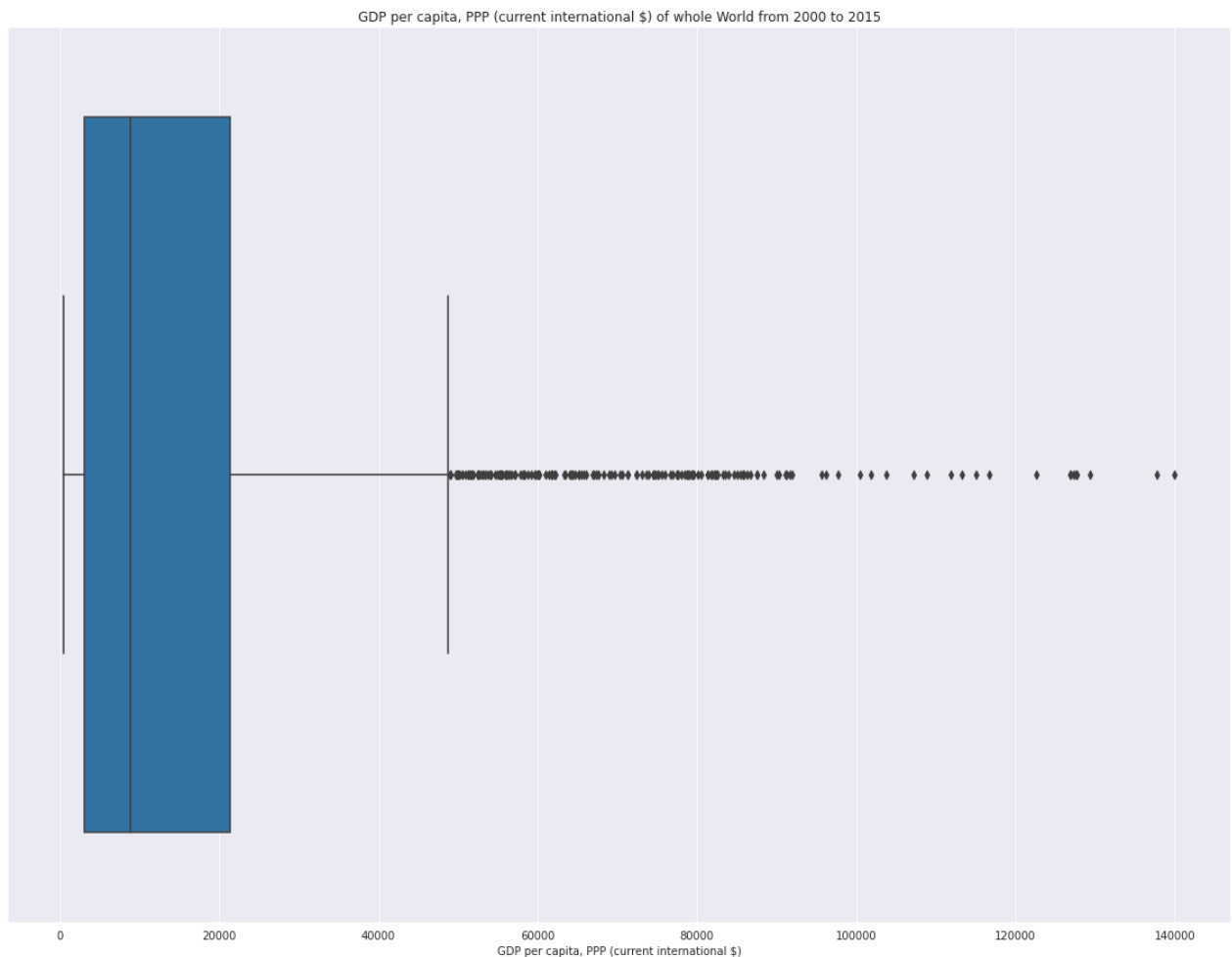
In this time-series plot, it shows that gross domestic product scaled with purchasing power parity and current international dollars. It means the GDP is scaled to the current value of dollars unit and apply the scaling on the price of common goods to have the same scale for comparison. It can be seen that comparing Thailand to the average of the world, Thailand is doing quite good. Because the GDP (PPP) of Thailand over time is around the average from 2000 to 2015. GDP is one of a very good indicator on how each country doing in term of economic. As generally know that economic status can affect other aspects of the country, for example, education, politics, health services.

```
In [ ]: plt.figure(figsize=(20,15))

ax = sns.boxplot(x="Value", data=gdp)

ax.set_title("GDP per capita, PPP (current international $) of whole World")
ax.set_xlabel('GDP per capita, PPP (current international $)')
```

```
Out[ ]: Text(0.5, 0, 'GDP per capita, PPP (current international $)')
```



Observing the boxplot and it can be seen that most country fall below the average, so Thailand doing around average is quite good to be said. But the maximum and minimum also very wide. So, some countries has very high and extremely low GDP.

PISA Thailand vs World

```
In [ ]: pisa_math = education[education['Indicator Name'] == 'PISA: Mean performance on the math test']
pisa_read = education[education['Indicator Name'] == 'PISA: Mean performance on the reading test']
pisa_sci = education[education['Indicator Name'] == 'PISA: Mean performance on the science test']
```

```
In [ ]: pisa_math_th = pisa_math[pisa_math['Country Name'] == 'Thailand']
pisa_read_th = pisa_read[pisa_read['Country Name'] == 'Thailand']
pisa_sci_th = pisa_sci[pisa_sci['Country Name'] == 'Thailand']
```

```
In [ ]: pisa_math_mean_world = pisa_math.groupby('Year', as_index=False).mean()
pisa_math_mean_world['Country Name'] = 'World'
pisa_math_mean_world['Country Code'] = 'WRD'
pisa_math_mean_world['Indicator Name'] = 'PISA: Mean performance on the math test'
pisa_math_mean_world['Indicator Code'] = 'LO.PISA.MAT'
pisa_math_mean_world.head()
```

Out[]:

	Year	Value	Country Name	Country Code	Indicator Name	Indicator Code
0	2000-01-01	470.226190	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT
1	2003-01-01	485.760587	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT
2	2006-01-01	467.643993	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT
3	2009-01-01	461.926995	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT
4	2012-01-01	473.109462	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT

In []:

```
pisa_read_mean_world = pisa_read.groupby('Year', as_index=False).mean()
pisa_read_mean_world['Country Name'] = 'World'
pisa_read_mean_world['Country Code'] = 'WRD'
pisa_read_mean_world['Indicator Name'] = 'PISA: Mean performance on the reading scale'
pisa_read_mean_world['Indicator Code'] = 'LO.PISA.REA'
pisa_read_mean_world.head()
```

Out[]:

	Year	Value	Country Name	Country Code	Indicator Name	Indicator Code
0	2000-01-01	471.349832	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA
1	2003-01-01	480.876088	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA
2	2006-01-01	458.893651	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA
3	2009-01-01	458.963963	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA
4	2012-01-01	473.204464	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA

In []:

```
pisa_sci_mean_world = pisa_sci.groupby('Year', as_index=False).mean()
pisa_sci_mean_world['Country Name'] = 'World'
pisa_sci_mean_world['Country Code'] = 'WRD'
pisa_sci_mean_world['Indicator Name'] = 'PISA: Mean performance on the science scale'
pisa_sci_mean_world['Indicator Code'] = 'LO.PISA.SCI'
pisa_sci_mean_world.head()
```

Out[]:

	Year	Value	Country Name	Country Code	Indicator Name	Indicator Code
0	2000-01-01	473.170070	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI
1	2003-01-01	488.519952	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI
2	2006-01-01	472.159265	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI
3	2009-01-01	466.125810	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI
4	2012-01-01	477.916935	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI

In []:

```
pisa_th_world = pd.concat([pisa_math_th, pisa_math_mean_world, pisa_read_th], axis=1)
pisa_th_world
```

Out[]:

	Country Name	Country Code	Indicator Name	Indicator Code	Year	Value
27405503	Thailand	THA	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2000-01-01	432.000000
30066293	Thailand	THA	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2003-01-01	416.977960
32727083	Thailand	THA	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2006-01-01	417.072614
35387873	Thailand	THA	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2009-01-01	418.583935
38048663	Thailand	THA	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2012-01-01	426.737491
40709453	Thailand	THA	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2015-01-01	415.463800
0	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2000-01-01	470.226190
1	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2003-01-01	485.760587
2	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2006-01-01	467.643993
3	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2009-01-01	461.926995
			PISA: Mean performance			

4	World	WRD	on the mathematics scale	LO.PISA.MAT	2012-01-01	473.109462
5	World	WRD	PISA: Mean performance on the mathematics scale	LO.PISA.MAT	2015-01-01	459.571028
27405506	Thailand	THA	PISA: Mean performance on the reading scale	LO.PISA.REA	2000-01-01	430.683873
30066296	Thailand	THA	PISA: Mean performance on the reading scale	LO.PISA.REA	2003-01-01	419.914829
32727086	Thailand	THA	PISA: Mean performance on the reading scale	LO.PISA.REA	2006-01-01	416.751765
35387876	Thailand	THA	PISA: Mean performance on the reading scale	LO.PISA.REA	2009-01-01	421.374414
38048666	Thailand	THA	PISA: Mean performance on the reading scale	LO.PISA.REA	2012-01-01	441.219934
40709456	Thailand	THA	PISA: Mean performance on the reading scale	LO.PISA.REA	2015-01-01	409.130100
0	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA	2000-01-01	471.349832
1	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA	2003-01-01	480.876088
2	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA	2006-01-01	458.893651
3	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA	2009-01-01	458.963963
4	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA	2012-01-01	473.204464
5	World	WRD	PISA: Mean performance on the reading scale	LO.PISA.REA	2015-01-01	459.393469
27405509	Thailand	THA	PISA: Mean performance on the science scale	LO.PISA.SCI	2000-01-01	436.378537
30066299	Thailand	THA	PISA: Mean performance on the science scale	LO.PISA.SCI	2003-01-01	429.060378
32727089	Thailand	THA	PISA: Mean performance on the science scale	LO.PISA.SCI	2006-01-01	421.011469
35387879	Thailand	THA	PISA: Mean performance on the science scale	LO.PISA.SCI	2009-01-01	425.296285
38048669	Thailand	THA	PISA: Mean performance on the science scale	LO.PISA.SCI	2012-01-01	443.999935
40709459	Thailand	THA	PISA: Mean performance on the science scale	LO.PISA.SCI	2015-01-01	421.337300
0	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI	2000-01-01	473.170070
1	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI	2003-01-01	488.519952
2	World	WRD	PISA: Mean performance	LO.PISA.SCI	2006-	472.159265

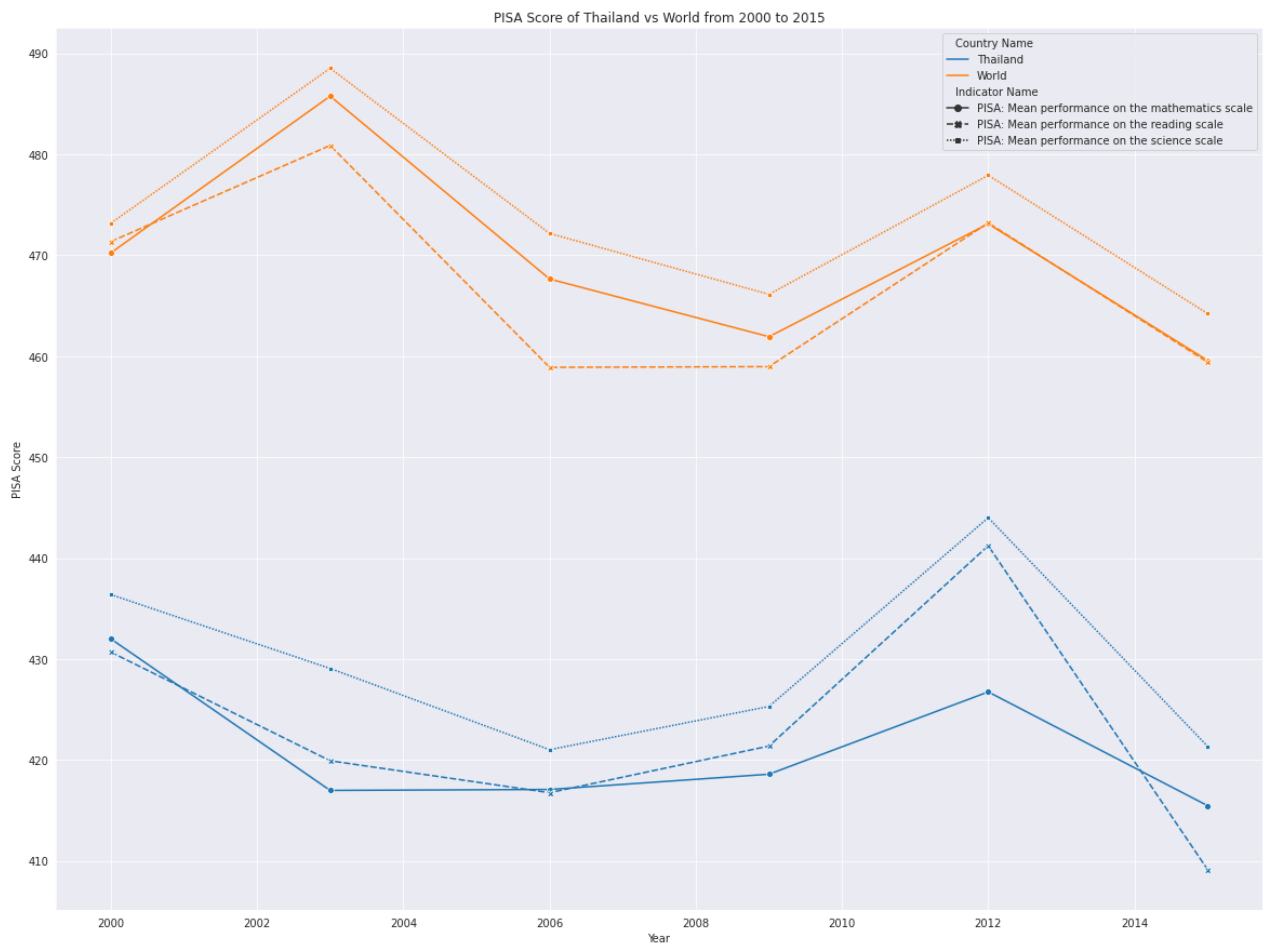
			on the science scale		01-01	
3	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI	2009-01-01	466.125810
4	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI	2012-01-01	477.916935
5	World	WRD	PISA: Mean performance on the science scale	LO.PISA.SCI	2015-01-01	464.213785

```
In [ ]: plt.figure(figsize=(20,15))

ax = sns.lineplot(data=pisa_th_world, x="Year", y="Value", hue='Country Name')

ax.set_title("PISA Score of Thailand vs World from 2000 to 2015")
ax.set_xlabel('Year')
ax.set_ylabel('PISA Score')
```

```
Out[ ]: Text(0, 0.5, 'PISA Score')
```



The Programme for International Student Assessment (PISA) is a worldwide study by the Organisation for Economic Co-operation and Development (OECD) in member and non-member nations intended to evaluate educational systems by measuring 15-year-old school pupils' scholastic performance on mathematics, science, and reading. PISA is held every three years. This graph stated all three types of PISA test's scores, mathematics, reading, and science. This chart comapres the score between Thailand and the average score on the world. The PISA score scaled to fit normal distribution, so maximum and minimum score does not really matter much. In this comparison, Thailand performs poorly compare to the world average. Despite the fact that Thailand performs quite well on the GDP that comparable to the average.

Average Years of Primary Schooling of Thailand vs World

```
In [ ]: primary_schooling = education[education['Indicator Name'] == 'Barro-Lee: A
```

```
In [ ]: primary_schooling_world = primary_schooling.groupby('Year', as_index=False)
primary_schooling_world['Country Name'] = 'World'
primary_schooling_world['Country Code'] = 'WTD'
primary_schooling_world['Indicator Name'] = 'Barro-Lee: Average years of p
primary_schooling_world['Indicator Code'] = 'BAR.PRM.SCHL.15UP'
primary_schooling_world.head()
```

Out []:

	Year	Value	Country Name	Country Code	Indicator Name	Indicator Code
0	1970-01-01	3.101944	World	WTD	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP
1	1975-01-01	3.357847	World	WTD	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP
2	1980-01-01	3.633611	World	WTD	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP
3	1985-01-01	3.890208	World	WTD	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP
4	1990-01-01	4.102153	World	WTD	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP

```
In [ ]: primary_schooling_th = primary_schooling[primary_schooling['Country Name']
primary_schooling_th.head()
```

Out[]:

	Country Name	Country Code	Indicator Name	Indicator Code	Year	Value
795455	Thailand	THA	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP	1970-01-01	2.15
5230105	Thailand	THA	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP	1975-01-01	2.46
9664755	Thailand	THA	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP	1980-01-01	2.87
14099405	Thailand	THA	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP	1985-01-01	3.24
18534055	Thailand	THA	Barro-Lee: Average years of primary schooling,...	BAR.PRM.SCHL.15UP	1990-01-01	3.75

In []:

```
primary_schooling_th_world = pd.concat([primary_schooling_th, primary_schoo
```

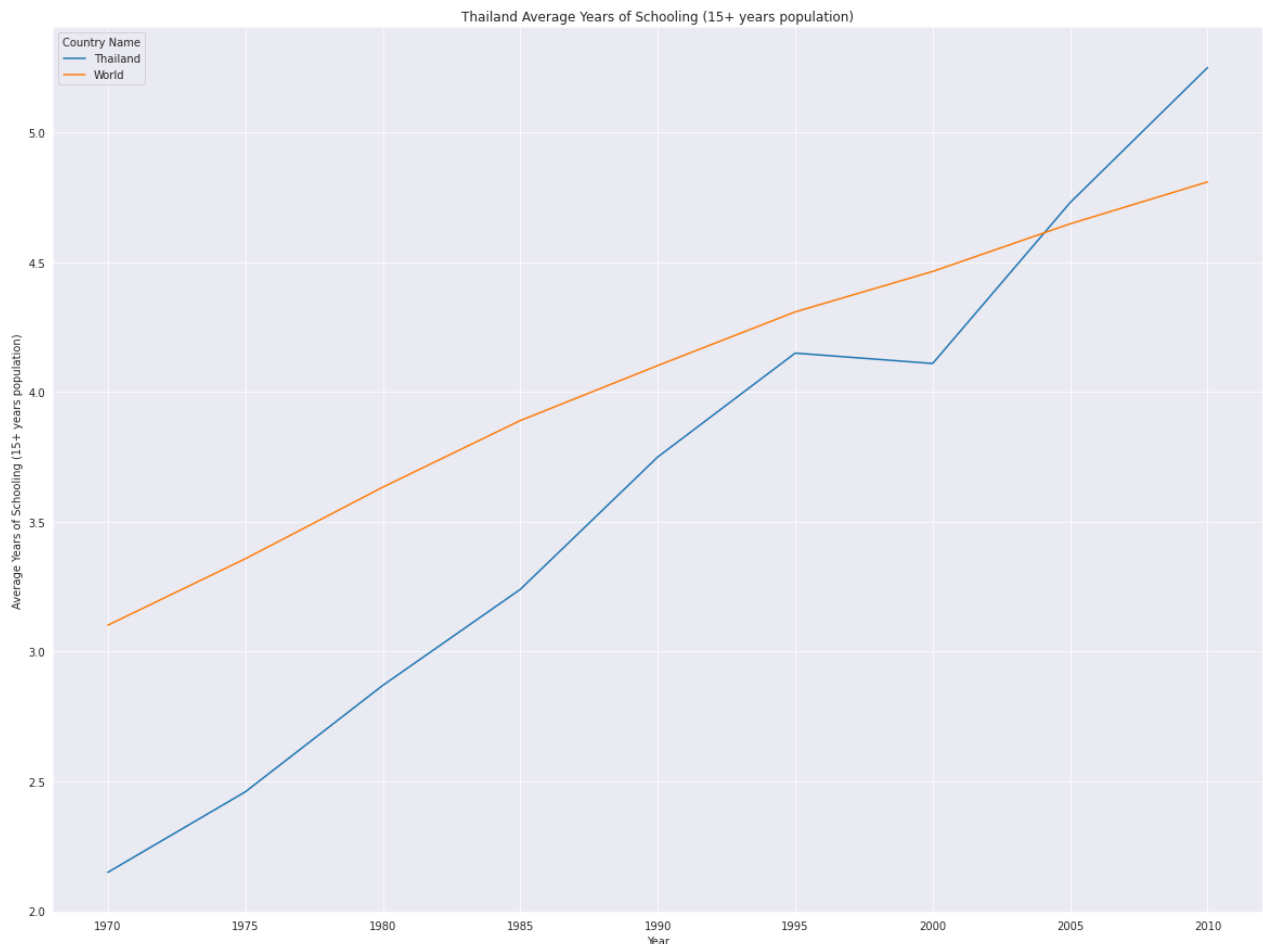
In []:

```
plt.figure(figsize=(20,15))

ax = sns.lineplot(data=primary_schooling_th_world, x="Year", y="Value", hue

ax.set_title("Thailand Average Years of Schooling (15+ years population)")
ax.set_xlabel('Year')
ax.set_ylabel('Average Years of Schooling (15+ years population)')
```

```
Out[ ]: Text(0, 0.5, 'Average Years of Schooling (15+ years population)')
```



This chart shows the average years of schooling of population with age over 15 in Thailand compare to the world average. It can be clearly seen that Thailand's primary schooling years is longer than the average of the world. So, we can roughly say that with longer priamry education time, population maybe better at basic skills such as mathematics and reading. But the trend of Thailand years is not that steep compare to the world average. So, the world puts more effort to accelerate the rate of change in primary schooling years.

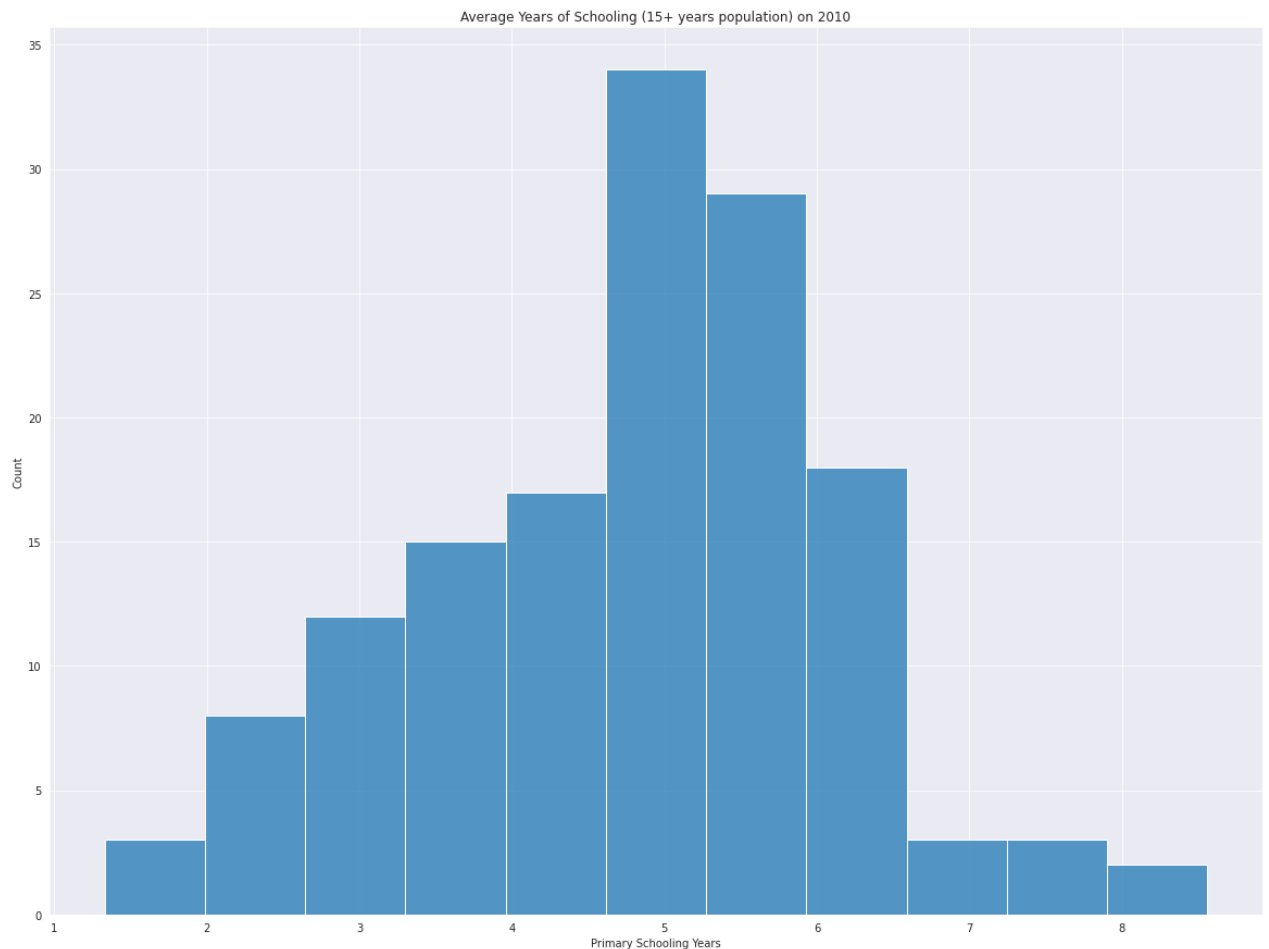
```
In [ ]: primary_schooling_2010 = primary_schooling[primary_schooling['Year'] == '2010']
```

```
In [ ]: plt.figure(figsize=(20,15))

ax = sns.histplot(primary_schooling_2010, x="Value")

ax.set_title("Average Years of Schooling (15+ years population) on 2010")
ax.set_xlabel('Primary Schooling Years')
ax.set_ylabel('Count')
```

```
Out[ ]: Text(0, 0.5, 'Count')
```



This histogram shows that most of the countries population has around 5 to 6 years (Thailand has around this number). This may not be surprising, since primary education usually take around 6 years for most of the countries. However, compare the left side and right side of the mean. It can be seen that most of the countries have less than 5 years of primary schooling.

TIMSS of Thailand vs World

```
In [ ]: timss_math_four = education[education['Indicator Name'] == 'TIMSS: Mean per  
timss_math_eight = education[education['Indicator Name'] == 'TIMSS: Mean pe  
timss_sci_four = education[education['Indicator Name'] == 'TIMSS: Mean per  
timss_sci_eight = education[education['Indicator Name'] == 'TIMSS: Mean pe
```

```
In [ ]: timss_math_four_th = timss_math_four[timss_math_four['Country Name'] == 'Th  
timss_math_eight_th = timss_math_eight[timss_math_eight['Country Name'] ==  
timss_sci_four_th = timss_sci_four[timss_sci_four['Country Name'] == 'Thai  
timss_sci_eight_th = timss_sci_eight[timss_sci_eight['Country Name'] == 'Th
```

```
In [ ]: timss_math_four_world = timss_math_four.groupby('Year', as_index=False).mea  
timss_math_four_world['Country Name'] = 'World'  
timss_math_four_world['Country Code'] = 'WRD'  
timss_math_four_world['Indicator Name'] = 'TIMSS: Mean performance on the r  
timss_math_four_world['Indicator Code'] = 'LO.TIMSS.MAT4'
```

```
In [ ]: timss_math_eight_world = timss_math_eight.groupby('Year', as_index=False).mean
timss_math_eight_world['Country Name'] = 'World'
timss_math_eight_world['Country Code'] = 'WRD'
timss_math_eight_world['Indicator Name'] = 'TIMSS: Mean performance on the
timss_math_eight_world['Indicator Code'] = 'LO.TIMSS.MAT8'
```

```
In [ ]: timss_sci_four_world = timss_sci_four.groupby('Year', as_index=False).mean
timss_sci_four_world['Country Name'] = 'World'
timss_sci_four_world['Country Code'] = 'WRD'
timss_sci_four_world['Indicator Name'] = 'TIMSS: Mean performance on the sc
timss_sci_four_world['Indicator Code'] = 'LO.TIMSS.SCI4'
```

```
In [ ]: timss_sci_eight_world = timss_sci_eight.groupby('Year', as_index=False).mean
timss_sci_eight_world['Country Name'] = 'World'
timss_sci_eight_world['Country Code'] = 'WRD'
timss_sci_eight_world['Indicator Name'] = 'TIMSS: Mean performance on the s
timss_sci_eight_world['Indicator Code'] = 'LO.TIMSS.SCI8'
```

```
In [ ]: timss = pd.concat([timss_math_four_th, timss_math_eight_th, timss_sci_four_th,
timss.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	Year	Value
22971636	Thailand	THA	TIMSS: Mean performance on the mathematics sca...	LO.TIMSS.MAT4	1995-01-01	466.523453
37162516	Thailand	THA	TIMSS: Mean performance on the mathematics sca...	LO.TIMSS.MAT4	2011-01-01	457.975859
22971633	Thailand	THA	TIMSS: Mean performance on the mathematics sca...	LO.TIMSS.MAT8	1995-01-01	516.215545
26519353	Thailand	THA	TIMSS: Mean performance on the mathematics sca...	LO.TIMSS.MAT8	1999-01-01	467.000000
33614793	Thailand	THA	TIMSS: Mean performance on the mathematics sca...	LO.TIMSS.MAT8	2007-01-01	441.000000

```
In [ ]: timss_four_math = pd.concat([timss_math_four_th, timss_math_four_world])
timss_eight_math = pd.concat([timss_math_eight_th, timss_math_eight_world])
timss_four_sci = pd.concat([timss_sci_four_th, timss_sci_four_world])
timss_eight_sci = pd.concat([timss_sci_eight_th, timss_sci_eight_world])
```

In []:

```
plt.figure(figsize=(20,15))

# ax.set_title("TIMSS of Thailand vs World")
# ax.set_xlabel('Year')
# ax.set_ylabel('TIMSS')

fig, axs = plt.subplots(2, 2, figsize=(20, 15))
fig.suptitle('TIMSS Score of Thailand vs World')

sns.lineplot(ax=axs[0, 0], data=timss_four_math, x="Year", y="Value", hue=
axs[0, 0].set_title("TIMSS Mathematics Score of Fourth Grade Student")
axs[0, 0].set_xlabel('Year')
axs[0, 0].set_ylabel('Mathematics Score (4th Grade)')
axs[0, 0].set_ylim(420, 530)

sns.lineplot(ax=axs[0, 1], data=timss_eight_math, x="Year", y="Value", hue=
axs[0, 1].set_title("TIMSS Mathematics Score of Eighth Grade Student")
axs[0, 1].set_xlabel('Year')
axs[0, 1].set_ylabel('Mathematics Score (8th Grade)')
axs[0, 1].set_ylim(420, 530)

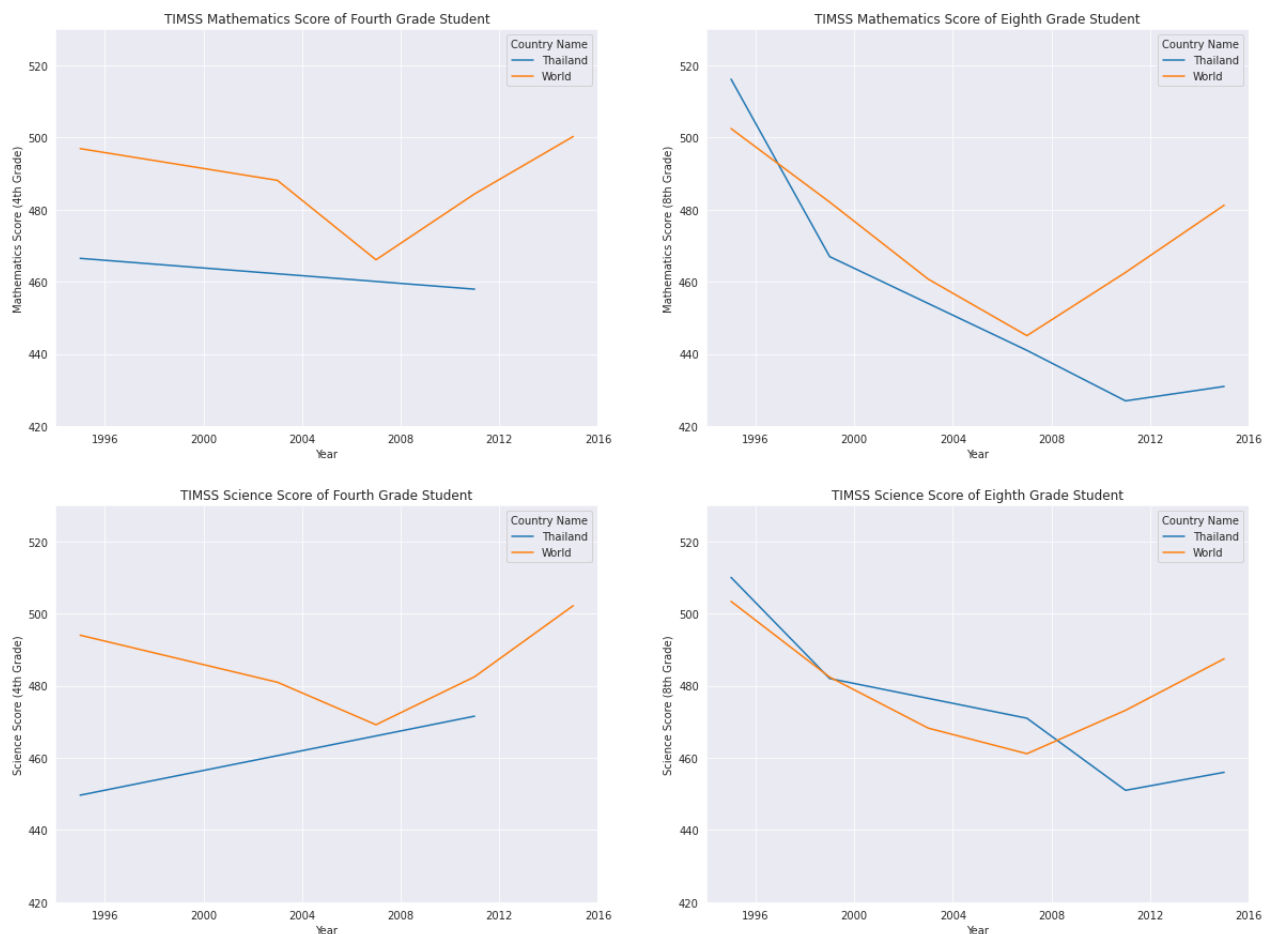
sns.lineplot(ax=axs[1, 0], data=timss_four_sci, x="Year", y="Value", hue='(
axs[1, 0].set_title("TIMSS Science Score of Fourth Grade Student")
axs[1, 0].set_xlabel('Year')
axs[1, 0].set_ylabel('Science Score (4th Grade)')
axs[1, 0].set_ylim(420, 530)

sns.lineplot(ax=axs[1, 1], data=timss_eight_sci, x="Year", y="Value", hue=
axs[1, 1].set_title("TIMSS Science Score of Eighth Grade Student")
axs[1, 1].set_xlabel('Year')
axs[1, 1].set_ylabel('Science Score (8th Grade)')
axs[1, 1].set_ylim(420, 530)
```

```
Out[ ]: (420.0, 530.0)
```

<Figure size 1440x1080 with 0 Axes>

TIMSS Score of Thailand vs World



Trends in International Mathematics and Science Study (TIMSS) is an international assessment focused on mathematics and science. Participants in tests are fourth and eighth grade students (or equivalent). In this visuals, it focuses on Thailand performance compare to the world average. From the graphs, we can see that Thailand performance is acceptable on all subjects in almost every years. While eighth students performing around the average, the score performed by fourth grade student is a little bit lower than the world average. So, this confirms that average performance of students in Thailand on mathematics is on average. And, Thai student are better at mathematics because study heavier and deeper is a myth. It is a truth that Thai students can achieved many academic internations awards. But look at the whole picture of every students in Thailand, that does not seem to be the case. Another interesting trend is that Thai mathematics and science scores except the science score by fourth grade students has decreasing trend, while the world seems to be an increasing trend after 2007. This could show a sign of a problem that we are moving backwards in term of academic performamnce at global scale.

```
In [ ]: timss_math_four_2015 = timss_math_four[timss_math_four['Year'] == '2015-01-01']
timss_math_eight_2015 = timss_math_eight[timss_math_eight['Year'] == '2015-01-01']
timss_sci_four_2015 = timss_sci_four[timss_sci_four['Year'] == '2015-01-01']
timss_sci_eight_2015 = timss_sci_eight[timss_sci_eight['Year'] == '2015-01-01']
```

In []:

```
fig, axs = plt.subplots(2, 2, figsize=(20, 15), sharey=True)
fig.suptitle('2015 TIMSS Score Distribution')

sns.histplot(ax=axs[0, 0], data=timss_math_four_2015, x="Value")
axs[0, 0].set_title("TIMSS Mathematics Score of Fourth Grade Student")
axs[0, 0].set_xlabel('Mathematics Score (4th Grade)')
axs[0, 0].set_ylabel('Count')
axs[0, 0].set_xlim(300, 630)

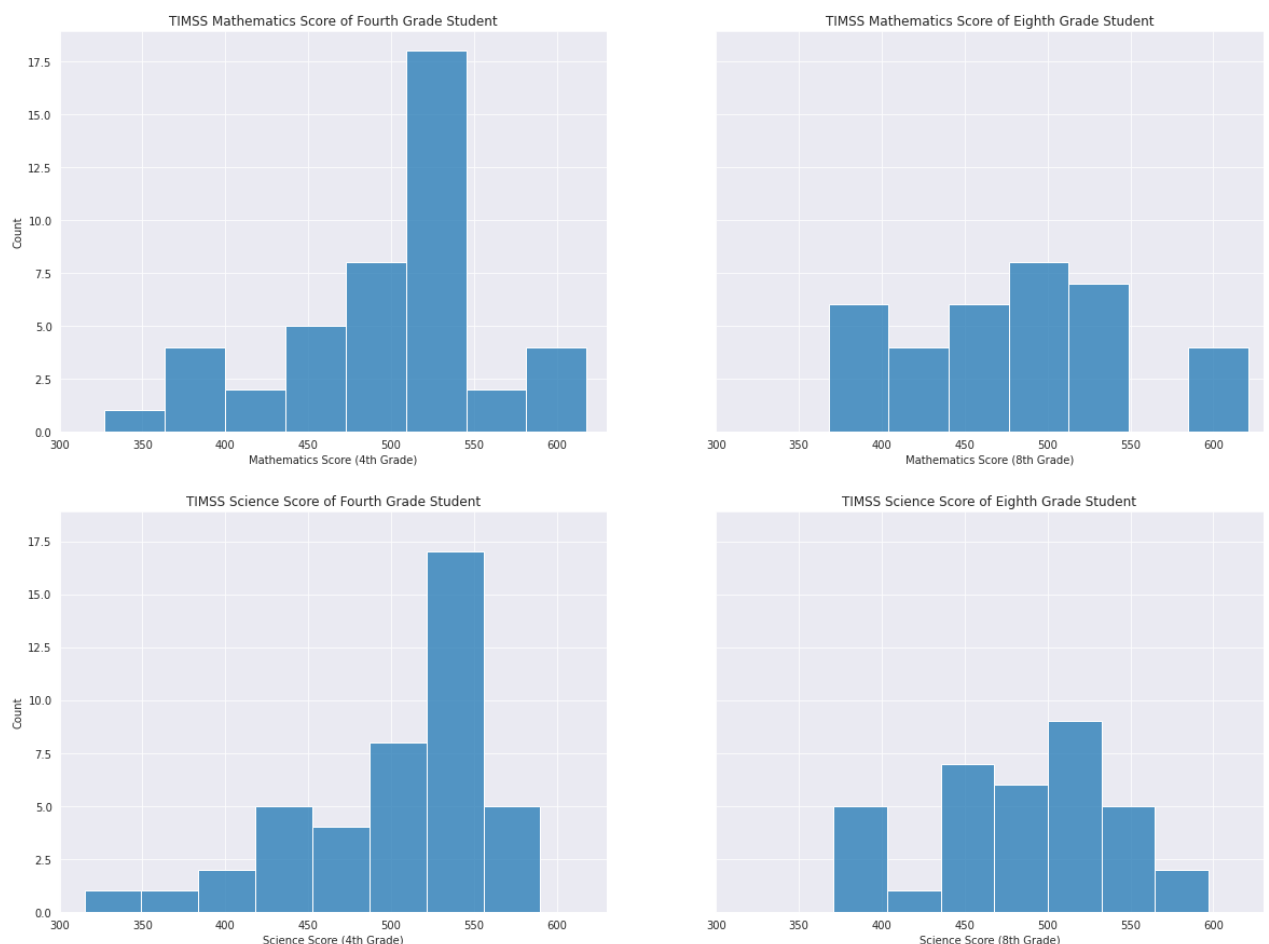
sns.histplot(ax=axs[0, 1], data=timss_math_eight_2015, x="Value")
axs[0, 1].set_title("TIMSS Mathematics Score of Eighth Grade Student")
axs[0, 1].set_xlabel('Mathematics Score (8th Grade)')
axs[0, 1].set_ylabel('Count')
axs[0, 1].set_xlim(300, 630)

sns.histplot(ax=axs[1, 0], data=timss_sci_four_2015, x="Value")
axs[1, 0].set_title("TIMSS Science Score of Fourth Grade Student")
axs[1, 0].set_xlabel('Science Score (4th Grade)')
axs[1, 0].set_ylabel('Count')
axs[1, 0].set_xlim(300, 630)

sns.histplot(ax=axs[1, 1], data=timss_sci_eight_2015, x="Value")
axs[1, 1].set_title("TIMSS Science Score of Eighth Grade Student")
axs[1, 1].set_xlabel('Science Score (8th Grade)')
axs[1, 1].set_ylabel('Count')
axs[1, 1].set_xlim(300, 630)
```

Out[]: (300.0, 630.0)

2015 TIMSS Score Distribution



In this chart, it shows the distribution of 2015 TIMSS scores. While the score performing by the eighth students show that mostly flat distributed. The score performed by fourth grade student crowded around 500 - 550.

In []:

```
fig, axs = plt.subplots(2, 2, figsize=(20, 15))
fig.suptitle('2015 TIMSS Score Top Ten Countries')

sns.barplot(ax=axs[0, 0], data=timss_math_four_2015.nlargest(10, 'Value'),
axs[0, 0].set_title("TIMSS Mathematics Score of Fourth Grade Student")
axs[0, 0].set_xlabel('Mathematics Score (4th Grade)')
axs[0, 0].set_ylabel('')

sns.barplot(ax=axs[0, 1], data=timss_math_eight_2015.nlargest(10, 'Value'),
axs[0, 1].set_title("TIMSS Mathematics Score of Eighth Grade Student")
axs[0, 1].set_xlabel('Mathematics Score (8th Grade)')
axs[0, 1].set_ylabel('')

sns.barplot(ax=axs[1, 0], data=timss_sci_four_2015.nlargest(10, 'Value'),
axs[1, 0].set_title("TIMSS Science Score of Fourth Grade Student")
axs[1, 0].set_xlabel('Science Score (4th Grade)')
axs[1, 0].set_ylabel('')

sns.barplot(ax=axs[1, 1], data=timss_sci_eight_2015.nlargest(10, 'Value'),
axs[1, 1].set_title("TIMSS Science Score of Eighth Grade Student")
axs[1, 1].set_xlabel('Science Score (8th Grade)')
axs[1, 1].set_ylabel('')
```

Out[]: Text(0, 0.5, '')

2015 TIMSS Score Top Ten Countries



The charts show the top ten countries with highest 2015 score in each category. The highest scores in all categories achieved by Singapore. So, it can said that Singapore students is very strong in mathematics and science skills. Korea Republic, Hong Kong, and Japan seems to be very competitive in term of the score as well. The top three in each categories do not have a large gap in the score, but rather very close cut. Surprisingly, Kazakhstan which has an image of third world country also performing very well and can be included in a top ten in very categories.

Data Wrangling for R/Orange

```
In [ ]: education[education['Indicator Name'] == 'Gross enrolment ratio, primary, l
```

Out[]:		Country Name	Country Code	Indicator Name	Indicator Code	Year	Value
	1331	Arab World	ARB	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	1970-01-01	66.621872
	4996	East Asia & Pacific	EAS	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	1970-01-01	102.804962
	8661	East Asia & Pacific (excluding high income)	EAP	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	1970-01-01	102.708969
	12326	Euro area	EMU	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	1970-01-01	107.150627
	15991	Europe & Central Asia	ECS	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	1970-01-01	105.036568

	41397506	Nepal	NPL	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	2016-01-01	135.862137
	41437821	Palau	PLW	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	2016-01-01	99.641830
	41492796	Sao Tome and Principe	STP	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	2016-01-01	114.956253
	41588086	Tajikistan	TJK	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	2016-01-01	100.468292
	41654056	Uzbekistan	UZB	Gross enrolment ratio, primary, both sexes (%)	SE.PRM.ENRR	2016-01-01	104.397751

8083 rows × 6 columns

```
In [ ]: education[education['Indicator Name'] == 'Gross enrolment ratio, primary, 1
```

```
In [ ]: education[education['Indicator Name'] == 'Barro-Lee: Percentage of populat
```

Out[]:		Country Name	Country Code	Indicator Name	Indicator Code	Year	Value
	92000	Afghanistan	AFG	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	1970-01-01	0.41
	95665	Albania	ALB	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	1970-01-01	23.49
	99330	Algeria	DZA	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	1970-01-01	5.23
	117655	Argentina	ARG	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	1970-01-01	30.77
	121320	Armenia	ARM	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	1970-01-01	19.17

	36338850	Venezuela, RB	VEN	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	2010-01-01	26.67
	36342515	Vietnam	VNM	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	2010-01-01	14.29
	36353510	Yemen, Rep.	YEM	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	2010-01-01	10.56
	36357175	Zambia	ZMB	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	2010-01-01	27.45
	36360840	Zimbabwe	ZWE	Barro-Lee: Percentage of population age 15+ wi...	BAR.PRM.CMPT.15UP.ZS	2010-01-01	26.68

1296 rows x 6 columns

In []:

```
education[education['Indicator Name'] == 'Barro-Lee: Percentage of populat:
```