

▼ ข้อสอบเก็บคะแนนรายวิชาการแสดงผลข้อมูลด้วยภาพ (กลางภาคเรียน)

คำอธิบาย

1. ข้อสอบมีทั้งหมด **XX** ข้อ ให้ผู้เรียนอ่านโจทย์แล้วดำเนินการตามคำสั่งให้ครบทุกข้อ รวมเก็บคะแนนทั้งสิ้น **20** คะแนน
2. ข้อสอบเป็นข้อสอบแบบเปิดตำรา
3. เมื่อผู้เรียนทำข้อสอบเสร็จให้ผู้เรียนดำเนินการบันทึกโดยเลือกเมนูคำสั่ง **File --> Save**
4. ผู้เรียนดำเนินการโหลดไฟล์ .ipynb โดยการคลิกที่เมนูคำสั่ง **File --> Download .ipynb** แล้วupload ใน **Google Class Room** ในหัวข้อ ข้อสอบเก็บคะแนนรายวิชาการแสดงผลข้อมูลด้วยภาพ (กลางภาคเรียน) ทั้งนี้ต้องให้ทันเวลาที่กำหนด
5. กำหนดเวลาสอบ **09.00-12.00** น. รวมเวลา Upload ไฟล์

: โปรดระบุรหัสนักศึกษา ตัวอย่างเช่น ** 056250204001-6**

เขียนตอบ--> 056250204018-0

: โปรดระบุ คำนามชื่อ ชื่อ นามสกุลนักศึกษา ตัวอย่างเช่น นางสาวน้ำทิพย์ บุศบัน

เขียนตอบ--> นายจิรพัฒน์ ศิริวสินชัย

▼ เริ่มต้นการทำข้อสอบ

กำหนดให้ผู้เรียน Run คำสั่งด้านล่างนี้เพื่อบันทึกเวลาเริ่มต้นการทำข้อสอบ

```
import pandas as pd
pd.Timestamp.now()

Timestamp('2022-01-06 03:46:54.335653')
```

▼ ข้อที่ 1

จงเขียนโปรแกรมเพื่อ ดำเนินการอ่านไฟล์ข้อมูลจาก Github ตามลิงก์

<https://github.com/pjveerawan/dataset/raw/DataviZ/WHO%20COVID->

[19%20global%20table%20data%20February%201st%202021%20at%203.04.47%20PM.csv](https://github.com/pjveerawan/dataset/raw/DataviZ/WHO%20COVID-19%20global%20table%20data%20February%201st%202021%20at%203.04.47%20PM.csv)

โครงสร้างข้อมูลประกอบด้วย

1. Name : ชื่อประเทศ
2. WHO Region : ทวีปตาม WHO กำหนด
3. Cases - cumulative total : ยอดผู้ป่วย COVID-19 สะสมทั้งหมด
4. Cases - cumulative total per 1 million population : ยอดผู้ป่วย COVID-19 สะสมต่อประชากร 1 ล้านคน
5. Cases - newly reported in last 7 days : ผู้ป่วยใหม่ COVID-19 ใน 7 วันที่ผ่านมา
6. Cases - newly reported in last 24 hours : ผู้ป่วย COVID-19 ใน 24 ชม. ที่ผ่านมา
7. Deaths - cumulative total : ยอดผู้เสียชีวิตจาก COVID-19 สะสมทั้งหมด
8. Deaths - cumulative total per 1 million population : ยอดผู้เสียชีวิตจาก COVID-19 ต่อประชากร 1 ล้านคน
9. Deaths - newly reported in last 7 days : ยอดผู้เสียชีวิตจาก COVID-19 ใน 7 วันที่ผ่านมา
10. Deaths - newly reported in last 24 hours : ยอดผู้เสียชีวิตจาก COVID-19 ใน 24 ชม. ที่ผ่านมา
11. Transmission Classification : กลุ่มของการเจ็บป่วย COVID-19

จะได้ผลลัพธ์ข้อมูลดังนี้

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 1 million population	Cases - newly reported in last 7 days	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 1 million population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 24 hours	Transmission Classification
0	United States of America	Americas	25676612	77572.22	1072287	164415	433173	1308.67	22506	3521	Community transmission
1	India	South-East Asia	10746183	7787.06	91650	13052	154274	111.79	935	127	Clusters of cases
2	Brazil	Americas	9118513	42898.65	364593	59826	222666	1047.55	7423	1119	Community transmission
3	Russian Federation	Europe	3850439	26384.71	131039	18359	73182	501.47	3720	485	Clusters of cases
4	The United Kingdom	Europe	3743738	55147.41	126275	0	103126	1519.11	5797	0	Community transmission

```
df = pd.read_csv('https://github.com/pjveerawan/dataset/raw/DataviZ/WHO%20COVID-19%20global%20table%20data%20February%201st%202021%20at%203.04.47%20PM.csv')
df
```

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 1 million population	Cases - newly reported in last 7 days	Cases - newly reported in last 24 hours	Deaths - cumulative total
0	United States of America	Americas	25676612	77572.22	1072287	164415	433173
1	India	South-East Asia	10746183	7787.06	91650	13052	154274
2	Brazil	Americas	9118513	42898.65	364593	59826	222666
3	Russian Federation	Europe	3850439	26384.71	131039	18359	73182
4	The United Kingdom	Europe	3743738	55147.41	126275	0	103126
...

▼ ข้อที่ 2

จงเขียนโปรแกรมเพื่อสร้าง Data Frame ชื่อ dfTest โดยมีข้อมูลทั้งหมด 4 คอลัมน์ประกอบด้วย

1. Name
2. WHO Region
3. Cases - cumulative total
4. Deaths - cumulative total

พร้อมแสดงผลข้อมูลจากแถวแรกถึงแถวที่ 20

ผลลัพธ์ข้อมูลแสดงดังภาพ

	Name	WHO Region	Cases - cumulative total	Deaths - cumulative total
0	United States of America	Americas	25676612	433173
1	India	South-East Asia	10746183	154274
2	Brazil	Americas	9118513	222666
3	Russian Federation	Europe	3850439	73182
4	The United Kingdom	Europe	3743738	103126
5	France	Europe	3126351	75466
6	Spain	Europe	2705001	57806
7	Italy	Europe	2541783	88279
8	Turkey	Europe	2470901	25865
9	Germany	Europe	2216363	56945
10	Colombia	Americas	2077633	53284
11	Argentina	Americas	1915362	47775
12	Mexico	Americas	1841893	156579
13	Poland	Europe	1513385	37180
14	South Africa	Africa	1449236	43951
15	Iran (Islamic Republic of)	Eastern Mediterranean	1411731	57889
16	Ukraine	Europe	1219455	22707
17	Peru	Americas	1125875	
18	Indonesia	South-East Asia	1066313	
19	Czechia	Europe	984774	

```
dfTest = df[['Name', 'WHO Region', 'Cases - cumulative total', 'Deaths - cumulative total']]
dfTest.head(20)
```

	Name	WHO Region	Cases - cumulative total	Deaths - cumulative total
0	United States of America	Americas	25676612	433173
1	India	South-East Asia	10746183	154274
2	Brazil	Americas	9118513	222666
3	Russian Federation	Europe	3850439	73182
4	The United Kingdom	Europe	3743738	103126
5	France	Europe	3126351	75466
6	Spain	Europe	2705001	57806
7	Italy	Europe	2541783	88279
8	Turkey	Europe	2470901	25865

▼ ข้อที่ 3

จงเขียนโปรแกรมเพื่อ ดำเนินการเปลี่ยนชื่อ Column ใน Data Frame: **dfTest** รายละเอียดดังนี้

- WHO Region เปลี่ยนเป็น **Region**
- Cases - cumulative total เปลี่ยนเป็น **Cases_total**
- Deaths - cumulative total เปลี่ยนเป็น **Deaths_total**

พร้อมแสดงผลข้อมูลจากท้ายข้อมูลจำนวน 10 แถว

จะได้ผลลัพธ์ข้อมูลดังนี้

	Name	Region	Cases_total	Deaths_total
227	Micronesia (Federated States of)	Western Pacific	0	0
228	Nauru	Western Pacific	0	0
229	Niue	Western Pacific	0	0
230	Palau	Western Pacific	0	0
231	Pitcairn Islands	Western Pacific	0	0
232	Saint Helena	Africa	0	0
233	Tokelau	Western Pacific	0	0
234	Tonga	Western Pacific	0	0
235	Turkmenistan	Europe	0	0
236	Tuvalu	Western Pacific	0	0

```
dfTest.rename(columns={'WHO Region': 'Region'}, inplace=True)
dfTest.rename(columns={'Cases - cumulative total': 'Cases_total'}, inplace=True)
dfTest.rename(columns={'Deaths - cumulative total': 'Deaths_total'}, inplace=True)
dfTest.tail(10)
```

/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4308: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/errors.html>, errors=errors,

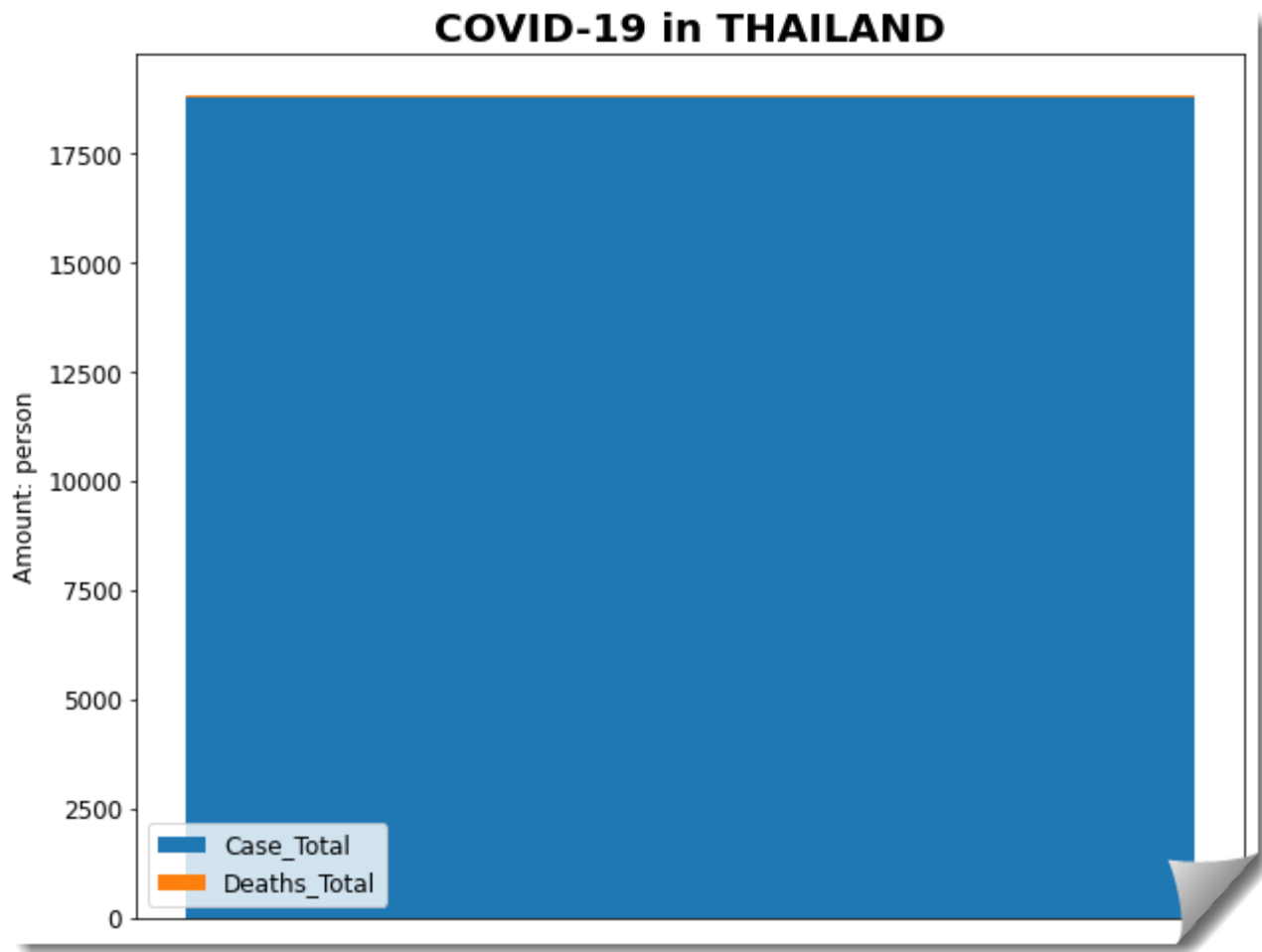
	Name	Region	Cases_total	Deaths_total	
227	Micronesia (Federated States of)	Western Pacific	0	0	
228	Nauru	Western Pacific	0	0	
229	Niue	Western Pacific	0	0	
230	Palau	Western Pacific	0	0	
231	Pitcairn Islands	Western Pacific	0	0	
232	Saint Helena	Africa	0	0	
233	Tokelau	Western Pacific	0	0	
234	Tonga	Western Pacific	0	0	
235	Turkmenistan	Europe	0	0	
236	Tuvalu	Western Pacific	0	0	

▼ ข้อที่ 4

จงเขียนโปรแกรมเพื่อ

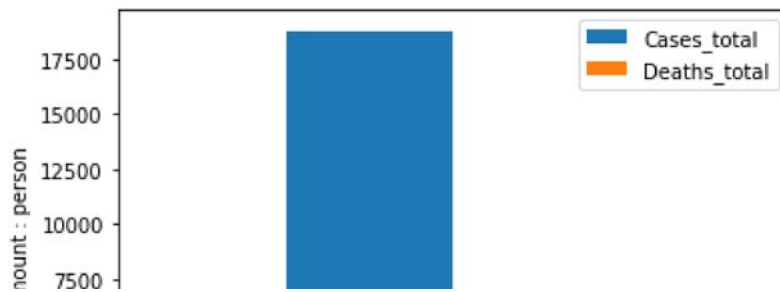
1. ดำเนินการเลือกข้อมูลเฉพาะประเทศไทย
2. สร้างกราฟข้อมูลด้วย matplotlib library

รายละเอียดกราฟผลลัพธ์ดังภาพด้านล่างนี้



```
import matplotlib.pyplot as plt
x = ['Cases_total', 'Deaths_total']
df4 = dfTest[dfTest['Name'] == 'Thailand']
df4.plot(kind = 'bar', ylabel = 'Amount : person')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6523fd0550>

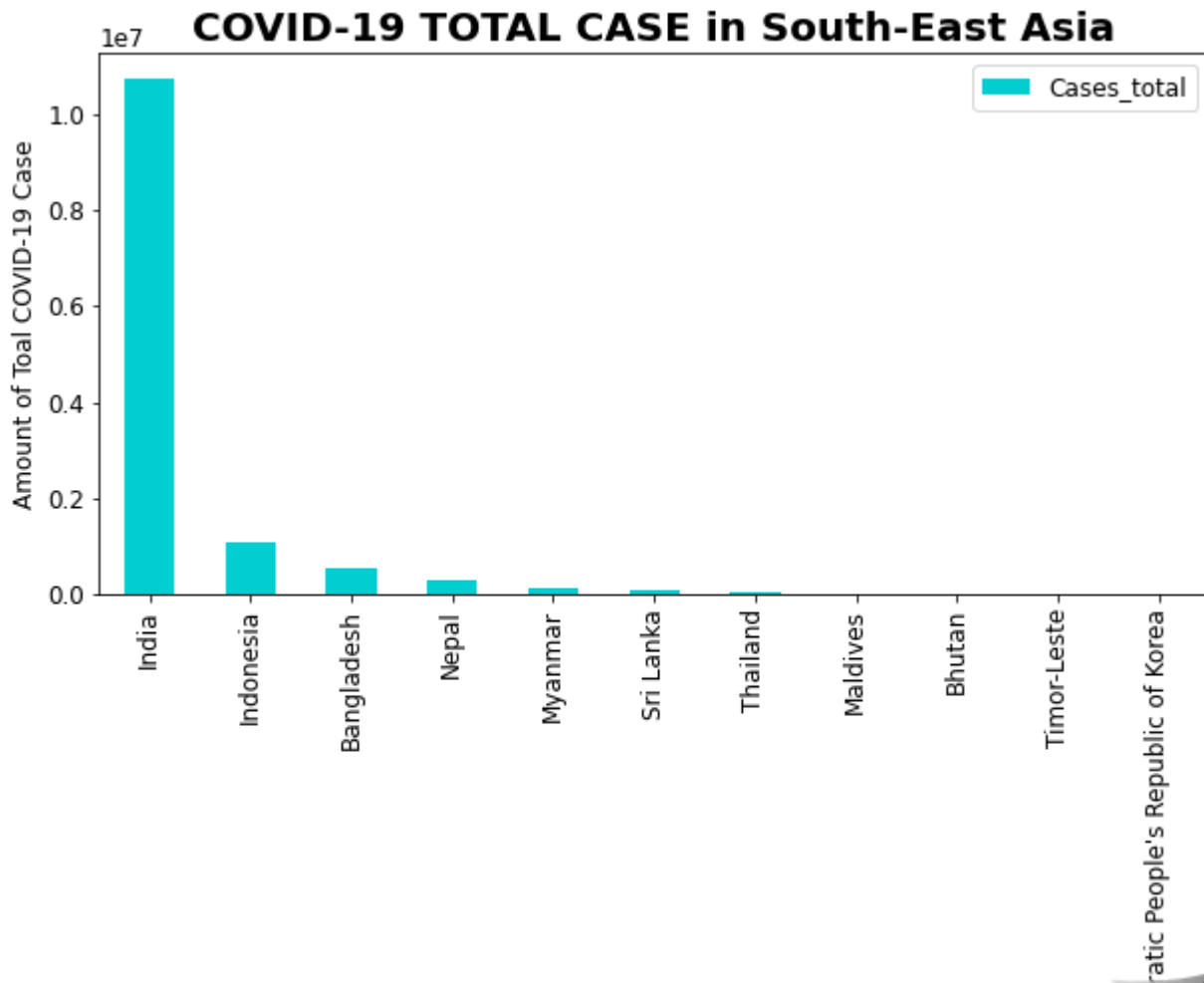


▼ ข้อที่ 5

จงเขียนโปรแกรมเพื่อ เลือกข้อมูลที่สนใจ คือ ข้อมูลยอดผู้ติดเชื้อในประเทศกลุ่ม South-East Asia ซึ่งประกอบด้วยประเทศต่าง ๆ ดังนี้

1. India
2. Indonesia
3. Bangladesh
4. Nepal
5. Myanmar
6. Sri Lanka
7. Thailand
8. Maldives
9. Bhutan
10. Timor-Leste
11. Korea

จากนั้นเขียนโปรแกรมเพื่อสร้างกราฟข้อมูลด้วยแพ็คเกจ pandas
รายละเอียดภาพกราฟผลลัพธ์แสดงดังภาพด้านล่างนี้



#ก่อนอื่นให้ผู้เรียนลองพิมพ์คำสั่งนี้เพื่อดูว่าในคอลัมน์ Region นั้น WHO แบ่งพื้นที่ออกเป็นกี่ Class
`dfTest['Region'].unique()`

```
array(['Americas', 'South-East Asia', 'Europe', 'Africa',  
      'Eastern Mediterranean', 'Western Pacific', 'Other'], dtype=object)
```

```
option = ['India',  
          'Indonesia',  
          'Bangladesh',  
          'Nepal',  
          'Myanmar',  
          'Sri Lanka',  
          'Thailand',  
          'Maldives',  
          'Bhutan',  
          'Timor-Leste',  
          'Korea']  
newdata = dfTest[dfTest['Name'].isin(option)]
```

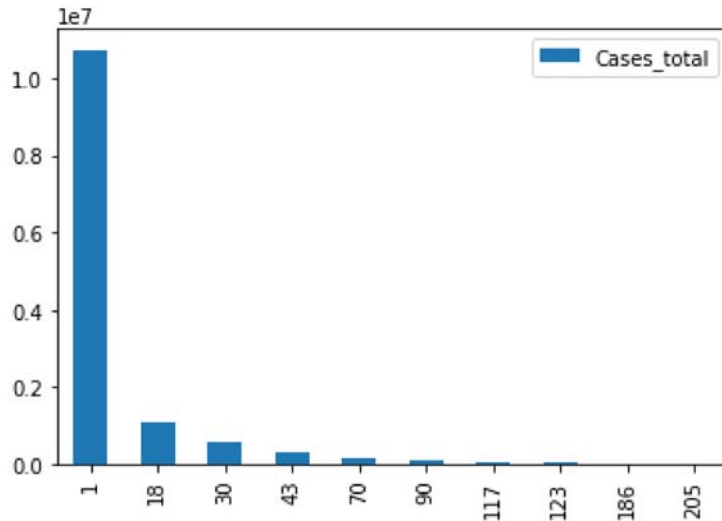
```
newdata.drop(columns=['Deaths_total'], inplace=True)
```

```
newdata.plot(kind = 'bar')
```

/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4174: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/errors.html>,

<matplotlib.axes._subplots.AxesSubplot at 0x7f6523ec2a10>



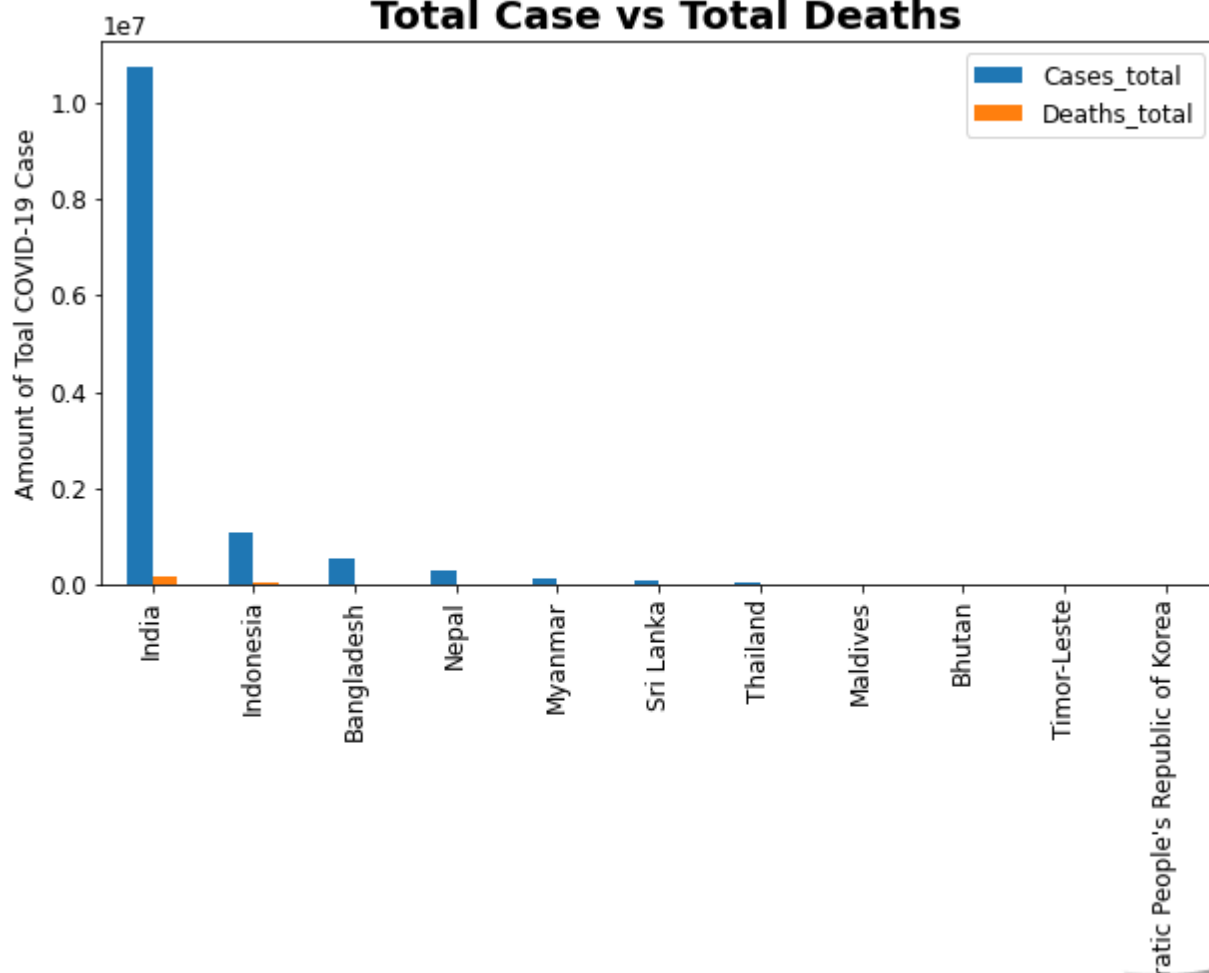
▼ ข้อที่ 6

จากข้อมูลในข้อที่ 5

จงเขียนโปรแกรมเพื่อ สร้างกราฟแสดงผลข้อมูลเพื่อเปรียบเทียบยอดผู้ป่วย และยอดผู้เสียชีวิต ในประเทศกลุ่ม South-East Asia

รายละเอียดของกราฟแสดงภาพด้านล่าง

COVID-19 in South-East Asia Total Case vs Total Deaths



```
option = ['India',
          'Indonesia',
          'Bangladesh',
          'Nepal',
          'Myanmar',
          'Sri Lanka',
          'Thailand',
          'Maldives',
          'Bhutan',
          'Timor-Leste',
          'Korea']
```

```
newdata = dfTest[dfTest['Name'].isin(option)]
newdata.plot(kind = 'bar')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6523995390>

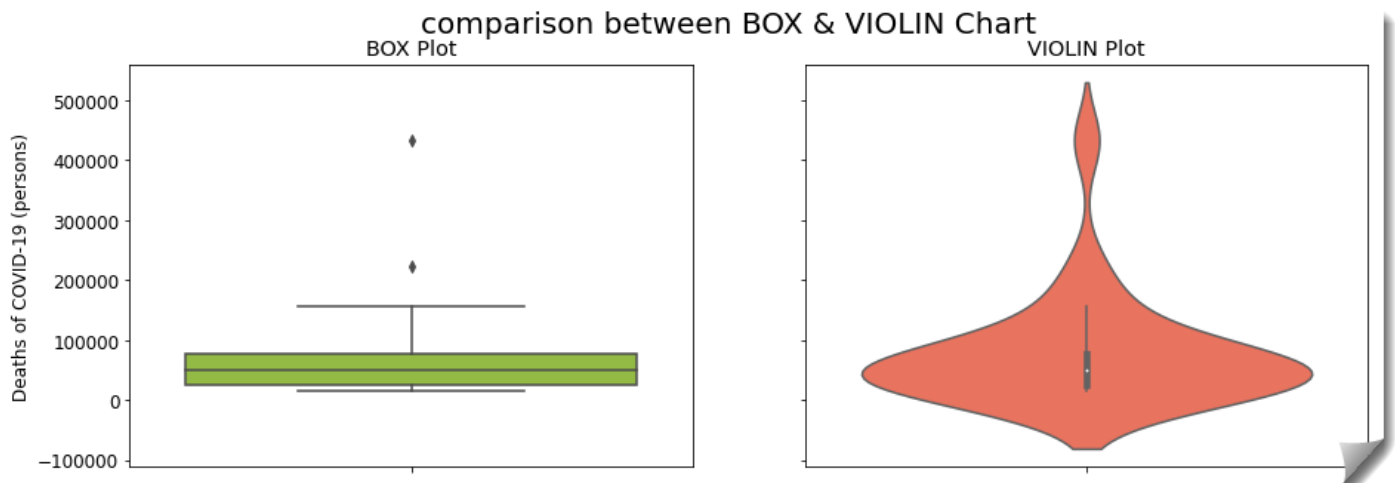


▼ ข้อที่ 7

พิจารณาข้อมูลจาก Data Frame : dfTest

จงเขียนโปรแกรมเพื่อเลือกแถวข้อมูลที่มีจำนวนผู้เสียชีวิตมากกว่า 15000 ราย

จงเขียนโปรแกรมเพื่อสร้างภาพกราฟข้อมูลให้มีรายละเอียดผลลัพธ์ดังภาพด้านล่างนี้



```
df7 = dfTest[dfTest['Deaths_total'] > 15000]
df7
```

	Name	Region	Cases_total	Deaths_total
0	United States of America	Americas	25676612	433173
1	India	South-East Asia	10746183	154274
2	Brazil	Americas	9118513	222666
3	Russian Federation	Europe	3850439	73182
4	The United Kingdom	Europe	3743738	103126
5	France	Europe	3126351	75466
6	Spain	Europe	2705001	57806
7	Italy	Europe	2541783	88279
8	Turkey	Europe	2470901	25865
9	Germany	Europe	2216363	56945
10	Colombia	Americas	2077633	53284
11	Argentina	Americas	1915362	47775
12	Mexico	Americas	1841893	156579
13	Poland	Europe	1513385	37180
14	South Africa	Africa	1449236	43951
15	Iran (Islamic Republic of)	Eastern Mediterranean	1411731	57889
16	Ukraine	Europe	1219455	22707
17	Peru	Americas	1125875	40686

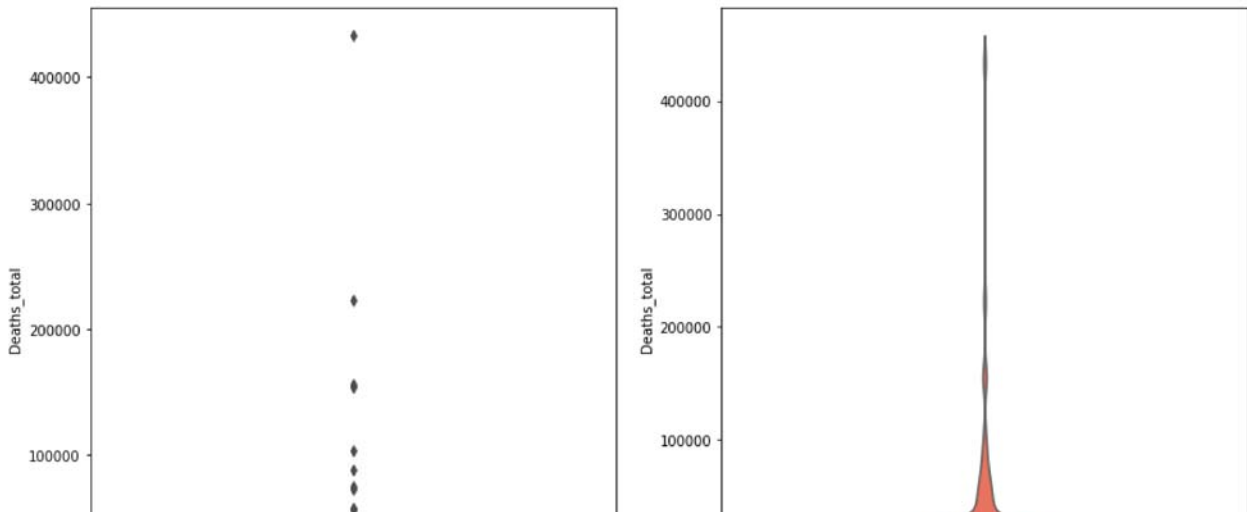


```
import seaborn as sns
df7 = dfTest[dfTest['Deaths_total'] > 15000]
df7

fig, axes = plt.subplots(1, 2, figsize=(15, 8))

sns.boxplot(y=dfTest['Deaths_total'], color='yellowgreen', ax=axes[0])
sns.violinplot(y=dfTest['Deaths_total'], color='tomato', ax=axes[1])
```

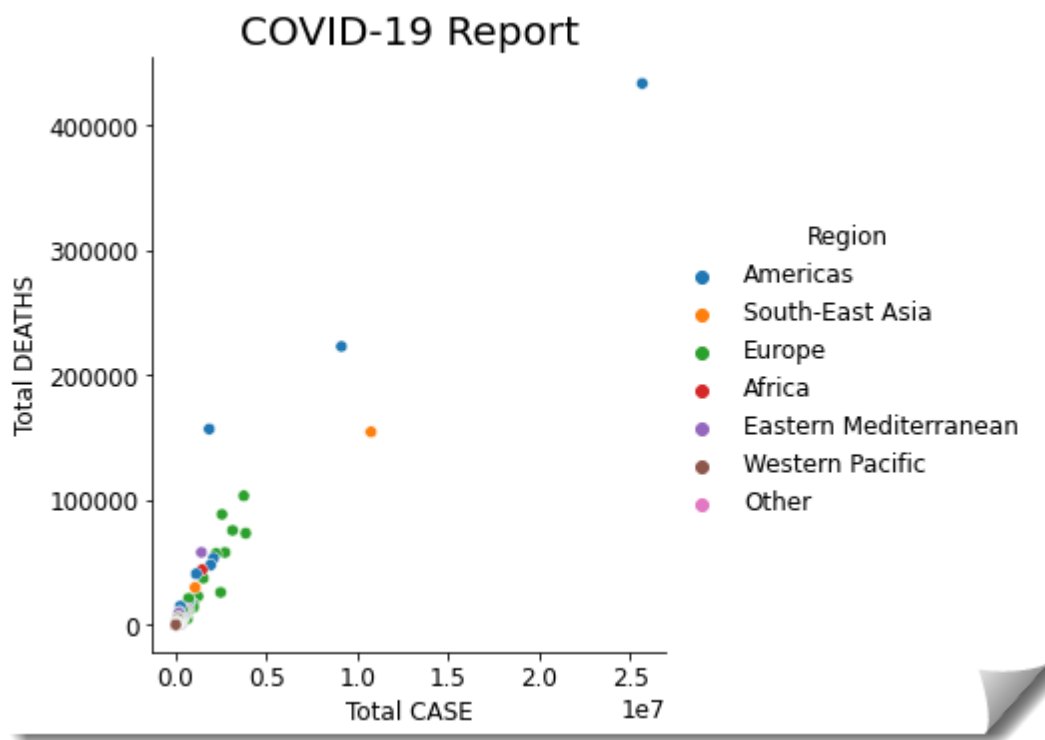
<matplotlib.axes._subplots.AxesSubplot at 0x7f6514461c90>




▼ ข้อที่ 8

พิจารณาข้อมูลจาก Data Frame : dfTest

จงเขียนโปรแกรมเพื่อสร้างภาพกราฟข้อมูลให้มีรายละเอียดผลลัพธ์ดังภาพด้านล่างนี้



ให้ผู้เรียนตรวจสอบข้อมูลจาก Data Frame : dfTest ก่อน
dfTest



	Name	Region	Cases_total	Deaths_total
0	United States of America	Americas	25676612	433173
1	India	South-East Asia	10746183	154274
2	Brazil	Americas	9118513	222666
3	Russian Federation	Europe	3850439	73182
4	The United Kingdom	Europe	3743738	103126
...
232	Saint Helena	Africa	0	0
233	Tokelau	Western Pacific	0	0
234	Tonga	Western Pacific	0	0
235	Turkmenistan	Europe	0	0

```
df8 = dfTest['Cases_total'] + dfTest['Deaths_total']
df8
```

```
0      26109785
1      10900457
2       9341179
3       3923621
4       3846864
...
232         0
233         0
234         0
235         0
236         0
Length: 237, dtype: int64
```

```
fig = plt.figure(figsize=(10,5))
sns.relplot(data=dfTest,
            x='Cases_total',
            y='Deaths_total',
            hue='Region')
```

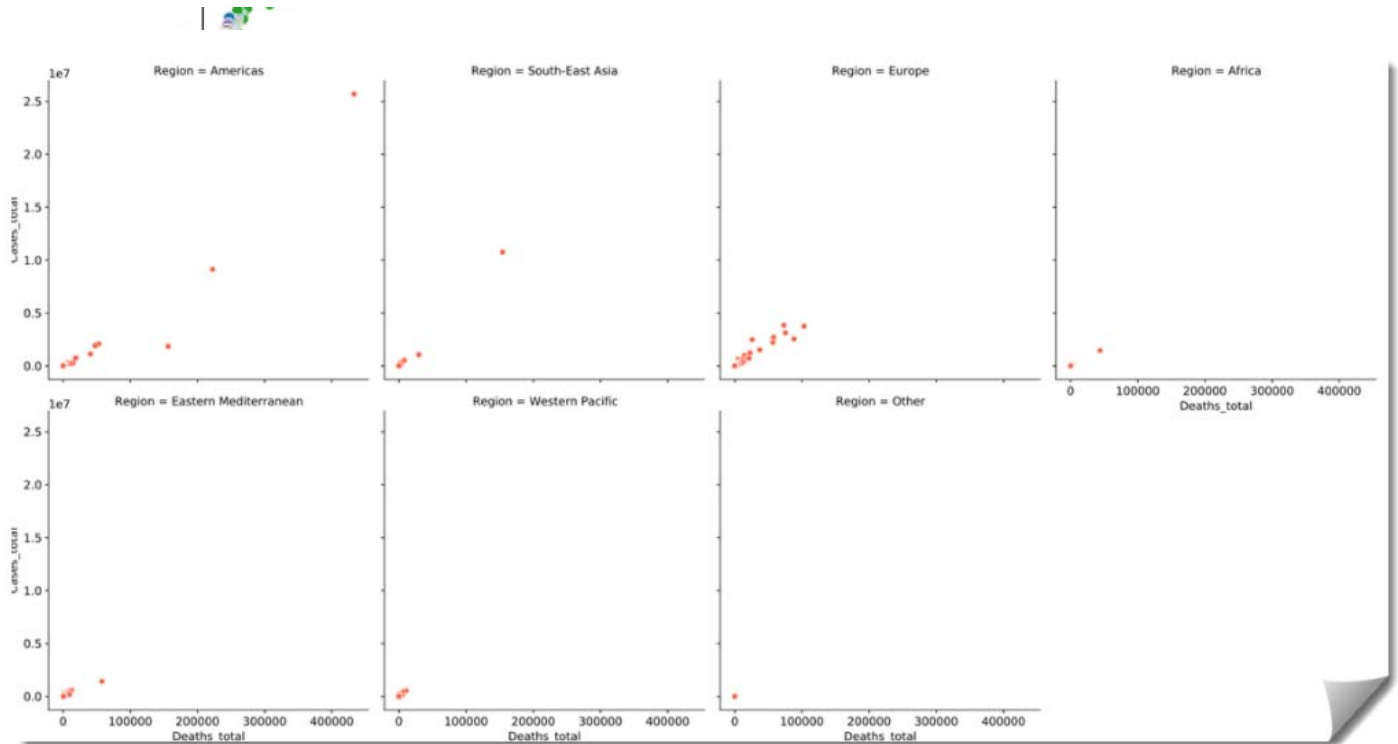
```
<seaborn.axisgrid.FacetGrid at 0x7f6514b85bd0>  
<Figure size 720x360 with 0 Axes>
```



▼ ข้อที่ 9

พิจารณาข้อมูลจาก Data Frame : dfTest

จงเขียนโปรแกรมเพื่อสร้างภาพกราฟข้อมูลให้มีรายละเอียดผลลัพธ์ดังภาพด้านล่างนี้

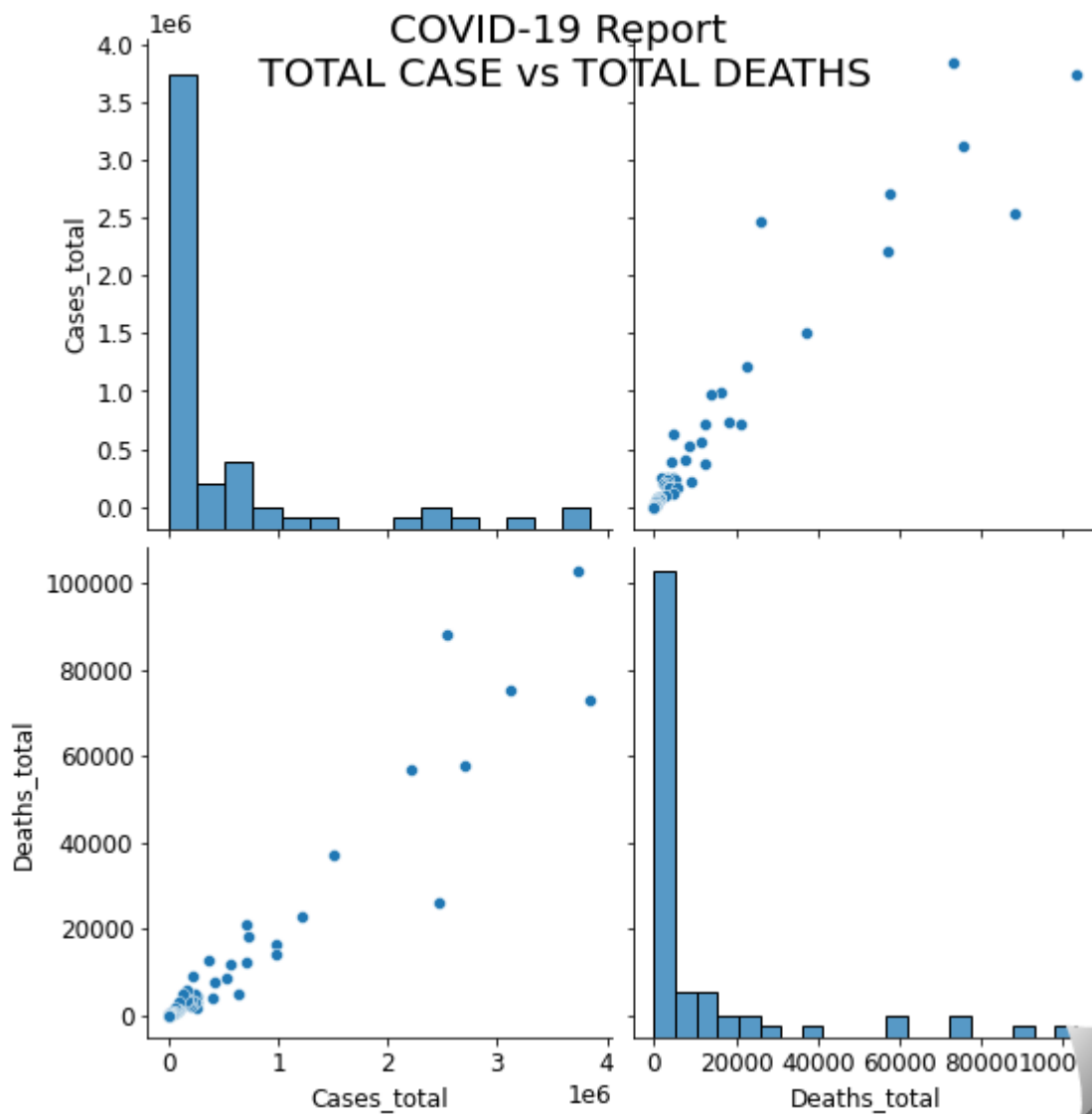


```
sns.relplot(data=tips,  
            x='total_bill',  
            y='tip',  
            hue='time',  
            col='day',  
            col_wrap=2)
```

▼ ข้อที่ 10

พิจารณาข้อมูลจาก Data Frame : dfTest

2. จงเขียนโปรแกรมเพื่อสร้างกราฟข้อมูล ตามภาพด้านล่างนี้ กรณีเลือกพื้นที่ Region = South-East Asia



ก่อนอื่นให้ผู้เรียนตรวจสอบข้อมูล Class ในคอลัมน์ข้อมูล Region ว่าประกอบด้วย Class อะไรบ้าง เพื่อนำไปสร้างราย
dfTest['Region'].unique()

```
array(['Americas', 'South-East Asia', 'Europe', 'Africa',  
      'Eastern Mediterranean', 'Western Pacific', 'Other'], dtype=object)
```

```
sns.pairplot(dfTest, vars=['Case_total', 'Deaths_total'],  
             hue='Region',  
             markers=['o', '*'],  
             height=4)
```

```
'''data=dfTest,  
    x='Cases_total'
```

```
x= cases_total ,  
y='Deaths_total',  
hue='Region''''
```

File "<ipython-input-17-de6ed6ce050a>", line 9
hue='Region''''

^

SyntaxError: EOL while scanning string literal

SEARCH STACK OVERFLOW

▼ สิ้นสุดการทำข้อสอบ

กำหนดให้ผู้เรียน Run คำสั่งด้านล่างนี้เพื่อบันทึกเวลาสิ้นสุดการทำข้อสอบ

```
import pandas as pd  
pd.Timestamp.now()
```

```
Timestamp('2022-01-06 03:48:58.518648')
```

✓ 0s completed at 10:48 AM

● ✕