

Họ tên: **Nguyễn Hữu Hiệu**

MSSV: **20520506**

Lớp: **DS103.N21**

BÀI THỰC HÀNH 1

Mục lục:

4. Bài tập	3
a) Code lại các ví dụ trong Phần 3	3
• import các thư viện python được sử dụng trong bài lab:	3
• Đọc dữ liệu từ file csv và gán vào biến df (dataframe) và in 10 dòng đầu:	3
• Liệt kê tên và số cột trong bảng dữ liệu:	3
• Liệt kê số dòng:	4
• Tạo biến countryCorona lưu giá trị là các quốc gia có dịch Corona	4
• Liệt kê số lượng ca lây nhiễm được xác nhận (biến Confirmed) nhiều nhất và lưu vào	5
• biến maxConfirmedCases	5
• Liệt kê các dữ liệu về covid-19 tại quốc gia Trung Quốc đại lục (Mainland China) và	5
• lưu vào biến coronaChina	5
• Tìm quốc gia (Country.Region) có số ca lây nhiễm nhiều nhất:	5
• Tìm tỉnh (Province.State) có số ca lây nhiễm nhiều nhất	6
• Lấy dữ liệu theo ngày tháng; Định dạng lại cột dữ liệu ngày tháng:	6
• Lấy dữ liệu trong tháng 1/2020: bắt đầu từ 01/01/2020 đến 31/01/2020	6
b) Tìm dữ liệu về số ca lây nhiễm tại Vietnam (Country.Region == 'Vietnam') và lưu vào biến coronaVietnam	7
c) In ra số ca lây nhiễm nhiều nhất tại Việt Nam	7
d) Tìm dữ liệu về số ca lây nhiễm tại Việt Nam trong tháng 02 năm 2021	8

e) In ra số dữ liệu về ca lây nhiễm nhiều nhất trong khoảng tháng 01 và 02 tại Việt Nam (Lấy năm 2021).....	9
f) Thực hiện tương tự câu e) cho Indonesia và Philipine.....	9
g) In ra dữ liệu về ca nhiễm ghi nhận (Confirmed) của Trung Quốc trong khoảng thời gian từ 01/02/2021 cho đến 15/02/2021.....	10
h) Thống kê số lượng record theo từng tỉnh của Trung Quốc trong tháng 02/2021	11
i) Đếm số lượng ca nhiễm theo từng tỉnh của Trung Quốc trong tháng 02/2021...	12
k) Tìm dữ liệu ca tử vong của Trung Quốc trong khoảng thời gian từ 01/02/2021 cho đến 15/02/2021.....	13
l) *Có nhận xét gì về số ca nhiễm mới tại Việt Nam giữa tháng 05/2020 và tháng 05/2021. Vẽ biểu đồ đường thể hiện số ca nhiễm mới trong 2 tháng trên:.....	14
m) * Vẽ biểu đồ về số ca lây nhiễm nhiều nhất của 3 quốc gia: Vietnam, Indonesia và Philippine trong 2 tháng gồm 01 và tháng 02 năm 2021.....	16

4. Bài tập

a) Code lại các ví dụ trong Phần 3.

- import các thư viện python được sử dụng trong bài lab:

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

- Đọc dữ liệu từ file csv và gán vào biến df (dataframe) và in 10 dòng đầu:

```
# Loading the csv file
df = pd.read_csv('data/covid_19_data.csv', index_col=0)

# showing the content of the first 10 records in the dataframe
df.head(10)
```

	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
SNo							
1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14.0	0.0	0.0
3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6.0	0.0	0.0
4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
6	01/22/2020	Guangdong	Mainland China	1/22/2020 17:00	26.0	0.0	0.0
7	01/22/2020	Guangxi	Mainland China	1/22/2020 17:00	2.0	0.0	0.0
8	01/22/2020	Guizhou	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
9	01/22/2020	Hainan	Mainland China	1/22/2020 17:00	4.0	0.0	0.0
10	01/22/2020	Hebei	Mainland China	1/22/2020 17:00	1.0	0.0	0.0

- Liệt kê tên và số cột trong bảng dữ liệu:

```
print(f"Coumms' name: {df.columns.tolist()}")
print(f"Number of columns in the dataframe: {len(df. columns)}")
```

```
Coumms' name: ['ObservationDate', 'Province/State', 'Country/Region', 'Last Update', 'Confirmed', 'Deaths', 'Recovered']
Number of columns in the dataframe: 7
```

- Liệt kê số dòng:

```
len(df.index)
```

```
306429
```

- Tạo biến countryCorona lưu giá trị là các quốc gia có dịch Corona

```
countryCorona = df["Country/Region"]
```

```
df['Country/Region'].unique()
```

```
array(['Mainland China', 'Hong Kong', 'Macau', 'Taiwan', 'US', 'Japan',
      'Thailand', 'South Korea', 'China', 'Kiribati', 'Singapore',
      'Philippines', 'Malaysia', 'Vietnam', 'Australia', 'Mexico',
      'Brazil', 'Colombia', 'France', 'Nepal', 'Canada', 'Cambodia',
      'Sri Lanka', 'Ivory Coast', 'Germany', 'Finland',
      'United Arab Emirates', 'India', 'Italy', 'UK', 'Russia', 'Sweden',
      'Spain', 'Belgium', 'Others', 'Egypt', 'Iran', 'Israel', 'Lebanon',
      'Iraq', 'Oman', 'Afghanistan', 'Bahrain', 'Kuwait', 'Austria',
      'Algeria', 'Croatia', 'Switzerland', 'Pakistan', 'Georgia',
      'Greece', 'North Macedonia', 'Norway', 'Romania', 'Denmark',
      'Estonia', 'Netherlands', 'San Marino', 'Azerbaijan', 'Belarus',
      'Iceland', 'Lithuania', 'New Zealand', 'Nigeria', 'North Ireland',
      'Ireland', 'Luxembourg', 'Monaco', 'Qatar', 'Ecuador',
      'Azerbaijan', 'Czech Republic', 'Armenia', 'Dominican Republic',
      'Indonesia', 'Portugal', 'Andorra', 'Latvia', 'Morocco',
      'Saudi Arabia', 'Senegal', 'Argentina', 'Chile', 'Jordan',
      'Ukraine', 'Saint Barthelemy', 'Hungary', 'Faroe Islands',
      'Gibraltar', 'Liechtenstein', 'Poland', 'Tunisia', 'Palestine',
      'Bosnia and Herzegovina', 'Slovenia', 'South Africa', 'Bhutan',
      'Cameroon', 'Costa Rica', 'Peru', 'Serbia', 'Slovakia', 'Togo',
      'Vatican City', 'French Guiana', 'Malta', 'Martinique',
      'Republic of Ireland', 'Bulgaria', 'Maldives', 'Bangladesh',
      'Moldova', 'Paraguay', 'Albania', 'Cyprus', 'St. Martin', 'Brunei',
      'occupied Palestinian territory', "('St. Martin',)",
      'Burkina Faso', 'Channel Islands', 'Holy See', 'Mongolia',
      'Panama', 'Bolivia', 'Honduras', 'Congo (Kinshasa)', 'Jamaica',
      'Reunion', 'Turkey', 'Cuba', 'Guyana', 'Kazakhstan',
      'Cayman Islands', 'Guadeloupe', 'Ethiopia', 'Sudan', 'Guinea',
      'Antigua and Barbuda', 'Aruba', 'Kenya', 'Uruguay', 'Ghana',
      'Jersey', 'Namibia', 'Seychelles', 'Trinidad and Tobago',
      'Venezuela', 'Curacao', 'Eswatini', 'Gabon', 'Guatemala',
      'Guernsey', 'Mauritania', 'Rwanda', 'Saint Lucia',
      'Saint Vincent and the Grenadines', 'Suriname', 'Kosovo',
      'Central African Republic', 'Congo (Brazzaville)',
      'Equatorial Guinea', 'Uzbekistan', 'Guam', 'Puerto Rico', 'Benin',
      'Greenland', 'Liberia', 'Mayotte', 'Republic of the Congo',
      'Somalia', 'Tanzania', 'The Bahamas', 'Barbados', 'Montenegro',
      'The Gambia', 'Kyrgyzstan', 'Mauritius', 'Zambia', 'Djibouti',
      'Gambia, The', 'Bahamas, The', 'Chad', 'El Salvador', 'Fiji',
      'Nicaragua', 'Madagascar', 'Haiti', 'Angola', 'Cabo Verde',
      'Niger', 'Papua New Guinea', 'Zimbabwe', 'Cape Verde',
      'East Timor', 'Eritrea', 'Uganda', 'Bahamas', 'Dominica', 'Gambia',
      'Grenada', 'Mozambique', 'Samoa', 'Syria', 'Timor-Leste', 'Belize',
      'Laos', 'Libya', 'Diamond Princess', 'Guinea-Bissau', 'Mali',
      'Saint Kitts and Nevis', 'West Bank and Gaza', 'Burma',
      'MS Zaandam', 'Botswana', 'Burundi', 'Sierra Leone', 'Malawi',
      'South Sudan', 'Sao Tome and Principe', 'Yemen', 'Comoros',
      'Tajikistan', 'Lesotho', 'Solomon Islands', 'Marshall Islands',
      'Vanuatu', 'Micronesia'], dtype=object)
```

- Liệt kê số lượng ca lây nhiễm được xác nhận (biến Confirmed) nhiều nhất và lưu vào
- biến maxConfirmedCases

```
maxConfirmedCases = max(df["Confirmed"])
print(maxConfirmedCases)
```

5863138.0

- Liệt kê các dữ liệu về covid-19 tại quốc gia Trung Quốc đại lục (Mainland China) và
- lưu vào biến coronaChina.

```
coronaChina = df[(df['Country/Region'] == "Mainland China")]
print(coronaChina)
```

SNo	ObservationDate	Province/State	Country/Region	Last Update	\
1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	
2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	
3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	
4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	
5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	
...	
306341	05/29/2021	Tibet	Mainland China	2021-05-30 04:20:55	
306369	05/29/2021	Unknown	Mainland China	2021-05-30 04:20:55	
306413	05/29/2021	Xinjiang	Mainland China	2021-05-30 04:20:55	
306421	05/29/2021	Yunnan	Mainland China	2021-05-30 04:20:55	
306427	05/29/2021	Zhejiang	Mainland China	2021-05-30 04:20:55	

SNo	Confirmed	Deaths	Recovered
1	1.0	0.0	0.0
2	14.0	0.0	0.0
3	6.0	0.0	0.0
4	1.0	0.0	0.0
5	0.0	0.0	0.0
...
306341	1.0	0.0	1.0
306369	0.0	0.0	10.0
306413	980.0	3.0	977.0
306421	352.0	2.0	331.0
306427	1364.0	1.0	1324.0

[15758 rows x 7 columns]

- Tìm quốc gia (Country.Region) có số ca lây nhiễm nhiều nhất:

```
maxCountryConfirmedCorona = df[df["Confirmed"]==maxConfirmedCases]['Country/Region'].values[0]
print(f"The country has the most confirmed cases is: {maxCountryConfirmedCorona}")
```

The country has the most confirmed cases is: France

- Tìm tỉnh (Province/State) có số ca lây nhiễm nhiều nhất

```
maxStateConfirmedCorona = df[df["Confirmed"]==maxConfirmedCases]['Province/State'].values[0]
print(f"The state has the most confirmed cases is: {maxStateConfirmedCorona}")
```

The state has the most confirmed cases is: nan

- Lấy dữ liệu theo ngày tháng; Định dạng lại cột dữ liệu ngày tháng:

```
df["ObservationDate"] = pd.to_datetime(df["ObservationDate"])
df["Last Update"] = pd.to_datetime(df["Last Update"])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 306429 entries, 1 to 306429
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ObservationDate        306429 non-null  datetime64[ns]
1   Province/State         228329 non-null  object
2   Country/Region         306429 non-null  object
3   Last Update            306429 non-null  datetime64[ns]
4   Confirmed              306429 non-null  float64
5   Deaths                306429 non-null  float64
6   Recovered              306429 non-null  float64
dtypes: datetime64[ns](2), float64(3), object(2)
memory usage: 18.7+ MB
```

- Lấy dữ liệu trong tháng 1/2020: bắt đầu từ 01/01/2020 đến 31/01/2020.

```
df[(df["ObservationDate"] >= "2020-01-01") & (df["ObservationDate"] <= "2020-01-31")]
```

b) Tìm dữ liệu về số ca lây nhiễm tại Vietnam (Country.Region == 'Vietnam') và lưu vào biến coronaVietnam

```
coronaVietnam = df[(df['Country/Region'] == "Vietnam")]
print(coronaVietnam)
```

SNo	ObservationDate	Province/State	Country/Region	Last Update	\
82	2020-01-23	NaN	Vietnam	2020-01-23 17:00:00	
128	2020-01-24	NaN	Vietnam	2020-01-24 17:00:00	
171	2020-01-25	NaN	Vietnam	2020-01-25 17:00:00	
219	2020-01-26	NaN	Vietnam	2020-01-26 16:00:00	
268	2020-01-27	NaN	Vietnam	2020-01-27 23:59:00	
...	
302772	2021-05-25	NaN	Vietnam	2021-05-26 04:20:35	
303537	2021-05-26	NaN	Vietnam	2021-05-27 04:20:59	
304302	2021-05-27	NaN	Vietnam	2021-05-28 04:20:36	
305067	2021-05-28	NaN	Vietnam	2021-05-29 04:20:41	
305832	2021-05-29	NaN	Vietnam	2021-05-30 04:20:55	

SNo	Confirmed	Deaths	Recovered
82	2.0	0.0	0.0
128	2.0	0.0	0.0
171	2.0	0.0	0.0
219	2.0	0.0	0.0
268	2.0	0.0	0.0
...
302772	5931.0	44.0	2794.0
303537	6086.0	45.0	2853.0
304302	6356.0	46.0	2853.0
305067	6396.0	47.0	2896.0
305832	6908.0	47.0	2896.0

[493 rows x 7 columns]

c) In ra số ca lây nhiễm nhiều nhất tại Việt Nam

```
maxVietnamConfirmedCases = max(coronaVietnam["Confirmed"])
print(f"The most cases in Vietnam are: {maxVietnamConfirmedCases}")
```

The most cases in Vietnam are: 6908.0

d) Tìm dữ liệu về số ca lây nhiễm tại Việt Nam trong tháng 02 năm 2021

```
cases_of_Vietnam_in_Feb2021 = coronaVietnam[(coronaVietnam["ObservationDate"].dt.year == 2021)
                                              & (coronaVietnam["ObservationDate"].dt.month == 2)]
print(cases_of_Vietnam_in_Feb2021)
```

	ObservationDate	Province/State	Country/Region	Last Update	\
SNo					
216327	2021-02-01	NaN	Vietnam	2021-04-02 15:13:53	
217092	2021-02-02	NaN	Vietnam	2021-04-02 15:13:53	
217857	2021-02-03	NaN	Vietnam	2021-04-02 15:13:53	
218622	2021-02-04	NaN	Vietnam	2021-04-02 15:13:53	
219387	2021-02-05	NaN	Vietnam	2021-04-02 15:13:53	
220152	2021-02-06	NaN	Vietnam	2021-04-02 15:13:53	
220917	2021-02-07	NaN	Vietnam	2021-04-02 15:13:53	
221682	2021-02-08	NaN	Vietnam	2021-04-02 15:13:53	
222447	2021-02-09	NaN	Vietnam	2021-04-02 15:13:53	
223212	2021-02-10	NaN	Vietnam	2021-04-02 15:13:53	
223977	2021-02-11	NaN	Vietnam	2021-04-02 15:13:53	
224742	2021-02-12	NaN	Vietnam	2021-04-02 15:13:53	
225507	2021-02-13	NaN	Vietnam	2021-04-02 15:13:53	
226272	2021-02-14	NaN	Vietnam	2021-04-02 15:13:53	
227037	2021-02-15	NaN	Vietnam	2021-04-02 15:13:53	
227802	2021-02-16	NaN	Vietnam	2021-04-02 15:13:53	
228567	2021-02-17	NaN	Vietnam	2021-04-02 15:13:53	
229332	2021-02-18	NaN	Vietnam	2021-04-02 15:13:53	
230097	2021-02-19	NaN	Vietnam	2021-04-02 15:13:53	
230862	2021-02-20	NaN	Vietnam	2021-04-02 15:13:53	
231627	2021-02-21	NaN	Vietnam	2021-04-02 15:13:53	
232392	2021-02-22	NaN	Vietnam	2021-04-02 15:13:53	
233157	2021-02-23	NaN	Vietnam	2021-04-02 15:13:53	
233922	2021-02-24	NaN	Vietnam	2021-04-02 15:13:53	
234687	2021-02-25	NaN	Vietnam	2021-04-02 15:13:53	
235452	2021-02-26	NaN	Vietnam	2021-04-02 15:13:53	
236217	2021-02-27	NaN	Vietnam	2021-04-02 15:13:53	
236982	2021-02-28	NaN	Vietnam	2021-04-02 15:13:53	

	Confirmed	Deaths	Recovered
SNo			
216327	1850.0	35.0	1460.0
217092	1882.0	35.0	1460.0
217857	1948.0	35.0	1461.0
218622	1957.0	35.0	1465.0
219387	1976.0	35.0	1465.0
220152	1985.0	35.0	1468.0
220917	2001.0	35.0	1472.0
221682	2050.0	35.0	1472.0
222447	2064.0	35.0	1472.0
223212	2091.0	35.0	1480.0
223977	2140.0	35.0	1528.0
224742	2142.0	35.0	1529.0
225507	2195.0	35.0	1529.0
226272	2228.0	35.0	1532.0
227037	2269.0	35.0	1541.0
227802	2311.0	35.0	1573.0
228567	2329.0	35.0	1574.0
229332	2347.0	35.0	1605.0
230097	2362.0	35.0	1627.0
230862	2368.0	35.0	1627.0
231627	2383.0	35.0	1702.0
232392	2392.0	35.0	1717.0
233157	2403.0	35.0	1760.0
233922	2412.0	35.0	1804.0
234687	2421.0	35.0	1804.0
235452	2426.0	35.0	1839.0
236217	2432.0	35.0	1844.0
236982	2448.0	35.0	1876.0

e) In ra số dữ liệu về ca lây nhiễm nhiều nhất trong khoảng tháng 01 và 02 tại Việt Nam (Lấy năm 2021)

```
corona_inJanFeb_atVietnam = coronaVietnam[(coronaVietnam["ObservationDate"] >= "1-1-2021")
& (coronaVietnam["ObservationDate"] <= "2-28-2021")]
print(
    f"The most cases in Vietnam between January and February in 2021 were: {max(corona_inJanFeb_atVietnam['Confirmed'])}"
)
# max_cases_of_Vietnam_in_JanFeb2021[max_cases_of_Vietnam_in_JanFeb2021["Confirmed"] == max(max_cases_of_Vietnam_in_JanFeb2021["Confirmed"])]
The most cases in Vietnam between January and February in 2021 were: 2448.0
```

f) Thực hiện tương tự câu e) cho Indonesia và Philipine

Show the data about the most cases in Indonesia between January and Feb 2021

```
corona_inJanFeb_atIndonesia = df[
    (df["Country/Region"] == "Indonesia")
    & (df["ObservationDate"] >= "1-1-2021")
    & (df["ObservationDate"] <= "2-28-2021")
]
max_cases_of_Indo_in_JanFeb2021 = max(corona_inJanFeb_atIndonesia["Confirmed"])
print(
    f"The most cases in Indonesia between January and February in 2021 were: {max_cases_of_Indo_in_JanFeb2021}"
)
# corona_inJanFeb_atIndonesia[corona_inJanFeb_atIndonesia["Confirmed"] == max_cases_of_Indo_in_JanFeb2021]
```

The most cases in Indonesia between January and February in 2021 were: 1334634.0

Show the data about the most cases in Philippines between January and Feb 2021

```
corona_inJanFeb_atPhilippines = df[
    (df["Country/Region"] == "Philippines")
    & (df["ObservationDate"] >= "1-1-2021")
    & (df["ObservationDate"] <= "2-28-2021")
]
max_cases_of_Philip_in_JanFeb2021 = max(corona_inJanFeb_atPhilippines["Confirmed"])
print(
    f"The most cases in Philippines between January and February in 2021 were: {max_cases_of_Philip_in_JanFeb2021}"
)
```

The most cases in Philippines between January and February in 2021 were: 576352.0

g) In ra dữ liệu về ca nhiễm ghi nhận (Confirmed) của Trung Quốc trong khoảng thời gian từ 01/02/2021 cho đến 15/02/2021

```
coronaChina_half_Feb = df[
    (df["Country/Region"] == "Mainland China" ) | (df["Country/Region"] == "China")
    & (df["ObservationDate"] >= "2-1-2021")
    & (df["ObservationDate"] <= "2-15-2021")
]
coronaChina_half_Feb
```

	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
SNo							
1	2020-01-22	Anhui	Mainland China	2020-01-22 17:00:00	1.0	0.0	0.0
2	2020-01-22	Beijing	Mainland China	2020-01-22 17:00:00	14.0	0.0	0.0
3	2020-01-22	Chongqing	Mainland China	2020-01-22 17:00:00	6.0	0.0	0.0
4	2020-01-22	Fujian	Mainland China	2020-01-22 17:00:00	1.0	0.0	0.0
5	2020-01-22	Gansu	Mainland China	2020-01-22 17:00:00	0.0	0.0	0.0
...
306341	2021-05-29	Tibet	Mainland China	2021-05-30 04:20:55	1.0	0.0	1.0
306369	2021-05-29	Unknown	Mainland China	2021-05-30 04:20:55	0.0	0.0	10.0
306413	2021-05-29	Xinjiang	Mainland China	2021-05-30 04:20:55	980.0	3.0	977.0
306421	2021-05-29	Yunnan	Mainland China	2021-05-30 04:20:55	352.0	2.0	331.0
306427	2021-05-29	Zhejiang	Mainland China	2021-05-30 04:20:55	1364.0	1.0	1324.0

15758 rows × 7 columns

h) Thống kê số lượng record theo từng tỉnh của Trung Quốc trong tháng 02/2021

```
df_China_in_Feb = df[(df["Country/Region"] == "Mainland China" )
                    & (df["ObservationDate"].dt.month == 2)
                    & (df["ObservationDate"].dt.year == 2021)]

df_China_in_Feb_groupby_Province = df_China_in_Feb['Province/State']
print(df_China_in_Feb_groupby_Province.value_counts())
```

Anhui	28
Beijing	28
Yunnan	28
Xinjiang	28
Unknown	28
Tibet	28
Tianjin	28
Sichuan	28
Shanxi	28
Shanghai	28
Shandong	28
Shaanxi	28
Qinghai	28
Ningxia	28
Liaoning	28
Jilin	28
Jiangxi	28
Jiangsu	28
Inner Mongolia	28
Hunan	28
Hubei	28
Henan	28
Heilongjiang	28
Hebei	28
Hainan	28
Guizhou	28
Guangxi	28
Guangdong	28
Gansu	28
Fujian	28
Chongqing	28
Zhejiang	28

Name: Province/State, dtype: int64

i) Đếm số lượng ca nhiễm theo từng tỉnh của Trung Quốc trong tháng 02/2021

```
def get_confirm_by_state(df_China_in_Feb):  
    lastConfirmedInFeb = df_China_in_Feb[df_China_in_Feb["ObservationDate"]  
                                         == df_China_in_Feb["ObservationDate"].max()][ "Confirmed"].values[0]  
    firstConfirmedInFeb = df_China_in_Feb[df_China_in_Feb["ObservationDate"]  
                                           == df_China_in_Feb["ObservationDate"].min()][ "Confirmed"].values[0]  
    return lastConfirmedInFeb - firstConfirmedInFeb
```

```
i = ChinaProvince_inFeb2021.apply(get_confirm_by_state)  
i = i.reset_index()  
i = i.rename(columns={0: "confirmed_count"})  
print(i)
```

	Province/State	confirmed_count
0	Anhui	0.0
1	Beijing	10.0
2	Chongqing	0.0
3	Fujian	10.0
4	Gansu	0.0
5	Guangdong	85.0
6	Guangxi	0.0
7	Guizhou	0.0
8	Hainan	0.0
9	Hebei	4.0
10	Heilongjiang	19.0
11	Henan	1.0
12	Hubei	1.0
13	Hunan	4.0
14	Inner Mongolia	1.0
15	Jiangsu	7.0
16	Jiangxi	0.0
17	Jilin	18.0
18	Liaoning	4.0
19	Ningxia	0.0
20	Qinghai	0.0
21	Shaanxi	11.0
22	Shandong	3.0
23	Shanghai	91.0
24	Shanxi	1.0
25	Sichuan	29.0
26	Tianjin	12.0
27	Tibet	0.0
28	Unknown	0.0
29	Xinjiang	0.0
30	Yunnan	2.0
31	Zhejiang	5.0

k) Tìm dữ liệu ca tử vong của Trung Quốc trong khoảng thời gian từ 01/02/2021 cho đến 15/02/2021

```
# Filter the data to include only records between February 1 and February 15
china_feb_1_to_15 = df_China_in_Feb[(df_China_in_Feb['ObservationDate'] >= '2021-02-01')
                                     & (df_China_in_Feb['ObservationDate'] <= '2021-02-15')]

print(f"China's death data from February 1, 2021 to February 15, 2021: {china_feb_1_to_15}")
```

China's death data from February 1, 2021 to February 15, 2021:				ObservationDate	Province/State	Country/Region	Last Update
\	SNo						
216354	2021-02-01	Anhui	Mainland China	2021-04-02 15:13:53			
216388	2021-02-01	Beijing	Mainland China	2021-04-02 15:13:53			
216438	2021-02-01	Chongqing	Mainland China	2021-04-02 15:13:53			
216480	2021-02-01	Fujian	Mainland China	2021-04-02 15:13:53			
216485	2021-02-01	Gansu	Mainland China	2021-04-02 15:13:53			
...			
227546	2021-02-15	Tibet	Mainland China	2021-04-02 15:13:53			
227574	2021-02-15	Unknown	Mainland China	2021-04-02 15:13:53			
227618	2021-02-15	Xinjiang	Mainland China	2021-04-02 15:13:53			
227626	2021-02-15	Yunnan	Mainland China	2021-04-02 15:13:53			
227632	2021-02-15	Zhejiang	Mainland China	2021-04-02 15:13:53			

	Confirmed	Deaths	Recovered
SNo			
216354	994.0	6.0	987.0
216388	1039.0	9.0	977.0
216438	591.0	6.0	584.0
216480	541.0	1.0	525.0
216485	187.0	2.0	181.0
...
227546	1.0	0.0	1.0
227574	0.0	0.0	0.0
227618	980.0	3.0	977.0
227626	231.0	2.0	228.0
227632	1320.0	1.0	1306.0

[480 rows x 7 columns]

```
# Group the data by province/state and sum up the number of deaths
China_province_deaths = china_feb_1_to_15.groupby('Province/State')['Deaths'].sum()
print(China_province_deaths)
```

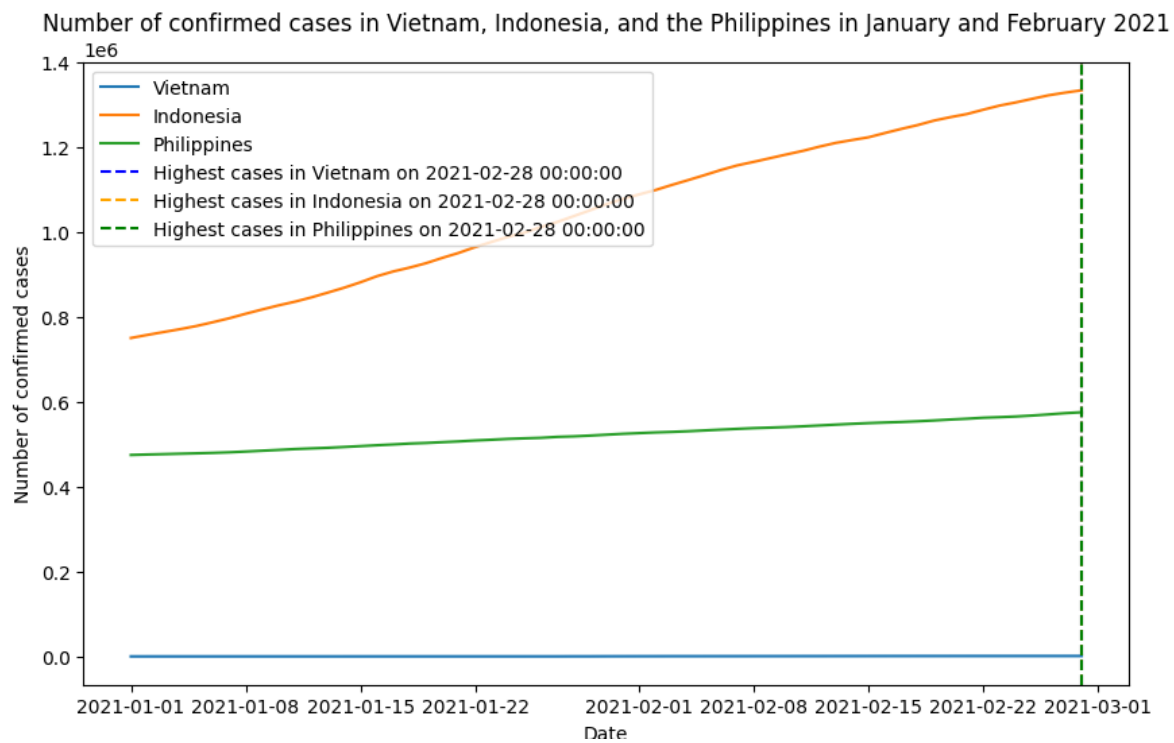
```
Province/State
Anhui          90.0
Beijing        135.0
Chongqing      90.0
Fujian         15.0
Gansu          30.0
Guangdong     120.0
Guangxi        30.0
Guizhou        30.0
Hainan         90.0
Hebei          105.0
Heilongjiang   195.0
Henan          330.0
Hubei          67680.0
Hunan          60.0
Inner Mongolia 15.0
Jiangsu        0.0
Jiangxi        15.0
Jilin          45.0
Liaoning       30.0
Ningxia        0.0
Qinghai        0.0
Shaanxi        45.0
Shandong       105.0
Shanghai       105.0
Shanxi         0.0
Sichuan        45.0
Tianjin        45.0
Tibet          0.0
Unknown        0.0
Xinjiang       45.0
Yunnan         30.0
Zhejiang       15.0
Name: Deaths, dtype: float64
```

l) *Có nhận xét gì về số ca nhiễm mới tại Việt Nam giữa tháng 05/2020 và tháng 05/2021. Vẽ biểu đồ đường thể hiện số ca nhiễm mới trong 2 tháng trên:

```
# Group the data by date and sum up the number of confirmed cases for each country
vietnam_cases = corona_inJanFeb_atVietnam.groupby('ObservationDate')['Confirmed'].sum()
indonesia_cases = corona_inJanFeb_atIndonesia.groupby('ObservationDate')['Confirmed'].sum()
philippines_cases = corona_inJanFeb_atPhilippines.groupby('ObservationDate')['Confirmed'].sum()

# Find the dates with the highest number of confirmed cases for each country
vietnam_highest_date = vietnam_cases.idxmax()
indonesia_highest_date = indonesia_cases.idxmax()
philippines_highest_date = philippines_cases.idxmax()
```

```
# Draw the graph
plt.figure(figsize=(10, 6))
plt.plot(vietnam_cases.index, vietnam_cases, label='Vietnam')
plt.plot(indonesia_cases.index, indonesia_cases, label='Indonesia')
plt.plot(philippines_cases.index, philippines_cases, label='Philippines')
plt.xlabel('Date')
plt.ylabel('Number of confirmed cases')
plt.title('Number of confirmed cases in Vietnam, Indonesia, and the Philippines in January and February 2021')
plt.axvline(vietnam_highest_date, color='blue', linestyle='--',
            label=f'Highest cases in Vietnam on {vietnam_highest_date}')
plt.axvline(indonesia_highest_date, color='orange', linestyle='--',
            label=f'Highest cases in Indonesia on {indonesia_highest_date}')
plt.axvline(philippines_highest_date, color='green', linestyle='--',
            label=f'Highest cases in Philippines on {philippines_highest_date}')
plt.legend()
plt.show()
```



m) * Vẽ biểu đồ về số ca lây nhiễm nhiều nhất của 3 quốc gia: Vietnam, Indonesia và Philippine trong 2 tháng gồm 01 và tháng 02 năm 2021

```
# Filter the data to include only records from May 2020 and May 2021
df_may2020 = df[(df['ObservationDate'].dt.year == 2020) & (df['ObservationDate'].dt.month == 5)]
df_may2021 = df[(df['ObservationDate'].dt.year == 2021) & (df['ObservationDate'].dt.month == 5)]

# Group the data by date and sum up the number of confirmed cases for each day
daily_cases_2020 = df_may2020.groupby('ObservationDate')['Confirmed'].sum().diff()
daily_cases_2021 = df_may2021.groupby('ObservationDate')['Confirmed'].sum().diff()

# Plot the graph
plt.figure(figsize=(10, 6))
plt.plot(daily_cases_2020.index, daily_cases_2020, label='May 2020')
plt.plot(daily_cases_2021.index, daily_cases_2021, label='May 2021')
plt.xlabel('Date')
plt.ylabel('Number of new infections')
plt.title('Comparison of new infections in May 2020 and May 2021')
plt.legend()
plt.show()
```

