

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



## CẤU TRÚC RỜI RẠC CHO KHMT (CO1007)

---

### BÁO CÁO BÀI TẬP LỚN PHÂN TÍCH & THỐNG KÊ DỮ LIỆU COVID-19

---

GVHD: Huỳnh Tường Nguyên  
Nguyễn Ngọc Lê

SV thực hiện: Nguyễn Duy Tùng – 2115232  
Trần Thiện Nhân – 2111913  
Đậu Xuân Thành – 2014486  
Lâm Tấn Thịnh – 2110559

Tp. Hồ Chí Minh, Tháng 04/2022



## Mục lục

<b>1</b>	<b>Động cơ nghiên cứu</b>	<b>2</b>
<b>2</b>	<b>Mục tiêu</b>	<b>2</b>
<b>3</b>	<b>Mô tả dữ liệu</b>	<b>2</b>
<b>4</b>	<b>Kiến thức quan trọng cần chuẩn bị</b>	<b>3</b>
4.1	Giá trị trung bình . . . . .	3
4.2	Giá trị trung vị . . . . .	3
4.3	Giá trị tứ phân vị . . . . .	3
4.4	Giá trị độ lệch chuẩn . . . . .	3
4.5	Outliers (dữ liệu ngoại lai) . . . . .	4
4.6	Sự tương quan . . . . .	4
<b>5</b>	<b>Bài làm của nhóm</b>	<b>5</b>
i	Nhóm câu hỏi liên quan đến tổng quát dữ liệu . . . . .	5
ii	Nhóm câu hỏi liên quan đến mô tả thống kê cơ bản dữ liệu . . . . .	12
iii	Nhóm câu hỏi liên quan đến dữ liệu thể hiện thu thập dữ liệu . . . . .	18
iv	Nhóm câu hỏi liên quan đến trực quan dữ liệu . . . . .	23
v	Nhóm câu hỏi liên quan đến trực quan dữ liệu theo thời gian là tháng . . . . .	29
vi	Nhóm câu hỏi liên quan đến trực quan dữ liệu theo trung bình 7 ngày gần nhất . . . . .	50
vii	Nhóm câu hỏi liên quan đến tất cả quốc gia theo thời gian là tháng . . . . .	84
viii	Nhóm câu hỏi liên quan tất cả quốc gia theo trung bình 7 ngày gần nhất . . . . .	96
ix	Nhóm câu hỏi liên quan đến sự tương quan giữa nhiễm bệnh và tử vong . . . . .	108
x	Nhóm câu hỏi riêng . . . . .	122
<b>6</b>	<b>Kết luận</b>	<b>125</b>
i	Ngôn ngữ R . . . . .	125
ii	Phân tích và thống kê . . . . .	125



## 1 Động cơ nghiên cứu

Bệnh Corona do virus gây ra còn gọi là COVID-19 đã tạo ra những tác động tiêu cực đến nền đời sống của cư dân trên thế giới. Các đợt bùng phát của COVID-19 hay những biến thể virus đã mang đến những thách thức chưa từng có và được dự báo sẽ có tác động đáng kể đến sự phát triển kinh tế. Nhiều thông tin, tin tức về tình hình dịch bệnh cũng như dữ liệu về COVID-19 được phổ biến rộng rãi trong đời sống hay trên internet để giúp cho mọi người quan sát, phân tích, nghiên cứu được cập nhật hàng ngày.

Phân tích & thống kê dữ liệu về COVID-19 giúp cho ta thấy được số ca nhiễm bệnh, tử vong của một quốc gia, so sánh tình trạng của các quốc gia trong khu vực hay diễn biến dịch trên thế giới. Từ số liệu được báo cáo mỗi chúng ta muốn biết các ca nhiễm bệnh có xu hướng tăng lên hay giảm xuống quy mô các đợt bùng phát ở mỗi quốc gia. Dữ liệu dùng cho bài tập lớn có tham khảo từ nguồn có thể xử lý trước với một vài thống kê cơ bản trước khi nó được truyền đi để khai thác dữ liệu thông minh sâu hơn.

## 2 Mục tiêu

Trong bài tập lớn này, các sinh viên sẽ bắt đầu với các bài toán thống kê đơn giản từ những dữ liệu được cung cấp. Qua đó, các em sẽ tìm ra những con số thú vị, có ý nghĩa đối với các dữ liệu thực tế từ tình hình dịch corona. Những kết quả mà các em tìm ra sẽ là bước khởi đầu cho việc khai phá nguồn dữ liệu của hệ thống sau này, nhằm đạt tới mục tiêu nâng cao kỹ năng lập trình, kỹ năng giải quyết vấn đề cho người học, kỹ năng làm việc nhóm cũng như hướng tới mục tiêu cao hơn là đam mê trong làm việc, học tập và nghiên cứu.

## 3 Mô tả dữ liệu

Dữ liệu gồm các thuộc tính chính “iso\_code, continent, location, date, new\_cases, new\_deaths” được lưu trong file csv.

1. *iso\_code*: Định danh đất nước
2. *continent*: Tên châu lục
3. *location*: Tên quốc gia
4. *date*: Ngày quan sát với định dạng Month-Day-Year
5. *new\_cases*: Số trường hợp COVID-19 mới được xác nhận
6. *new\_deaths*: Số tử vong mới do COVID-19



## 4 Kiến thức quan trọng cần chuẩn bị

### 4.1 Giá trị trung bình

Giá trị trung bình chính là tổng của tất cả các số rồi chia cho số lượng số.

Ví dụ: Cho dãy số sau (6, 5, 8, 7, 12, 13, 15, 14, 2, 200, 1), ở đây số lượng số là 11 số, như vậy giá trị trung bình cộng là  $\frac{6+5+8+7+12+13+15+14+2+200+1}{11} = 25.72$ .

Sử dụng câu lệnh trong R: mean(na.omit(data)). Trong đó: na.omit dùng để loại bỏ các giá trị NA.

### 4.2 Giá trị trung vị

Giá trị trung vị là số ở giữa trong một danh sách các số được sắp xếp tăng dần hoặc giảm dần và có thể mô tả nhiều hơn về tập dữ liệu so với giá trị trung bình. Đặc điểm cơ bản của giá trị trung vị trong việc mô tả dữ liệu so với giá trị trung bình là nó không bị sai lệch bởi một tỷ lệ nhỏ các giá trị cực lớn hoặc cực nhỏ, và do đó cung cấp một đại diện tốt hơn về giá trị đặc trưng.

Với dãy số trên khi sắp xếp lại theo thứ tự tăng dần ta được: 1, 2, 5, 6, 7, 8, 12, 13, 14, 15, 200. Vậy số trung vị là 8. Trước và sau số trung vị có 50% quan sát.

Nếu dãy số có số lượng phần tử là số chẵn thì số trung vị là trung bình cộng của 2 số ở giữa.

### 4.3 Giá trị tứ phân vị

Điểm tứ phân vị (quartile) là giá trị bằng số phân chia một nhóm các kết quả quan sát bằng số thành bốn phần, mỗi phần có số liệu quan sát bằng nhau (= 25% số kết quả quan sát). Tứ phân vị có 3 giá trị, đó là tứ phân vị thứ nhất (Q1), thứ nhì (Q2) và thứ ba (Q3). Ba giá trị này chia một tập hợp dữ liệu (đã sắp xếp dữ liệu theo trật tự từ bé đến lớn) thành 4 phần có số lượng quan sát đều nhau.

Xem lại dãy số 11 số ở trên của chúng ta (1, 2, 5, 6, 7, 8, 12, 13, 14, 15, 200)

Giá trị tứ phân vị thứ nhất Q1 bằng trung vị phần dưới, phần dưới là các số (1, 2, 5, 6, 7), là số 5

Giá trị tứ phân vị thứ hai Q2 chính bằng giá trị trung vị, chính là số 8

Giá trị tứ phân vị thứ ba Q3 bằng trung vị phần trên(12, 13, 14, 15, 200), là số 14

- Sử dụng câu lệnh trong R:

+ Tứ phân vị thứ nhất: quantile(na.omit(data),c(0.25))

+ Tứ phân vị thứ hai: quantile(na.omit(data),c(0.5))

+ Tứ phân vị thứ ba: quantile(na.omit(data),c(0.75))

### 4.4 Giá trị độ lệch chuẩn

Trong thống kê mô tả, độ lệch chuẩn là thước đo độ phân tán của một tập hợp các giá trị so với giá trị trung bình của chúng. Độ lệch chuẩn của 1 giá trị càng thấp nghĩa là giá trị đó càng gần với giá trị trung bình của tập hợp.

Công thức toán học:  $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$

Trong đó:

+  $s$ : độ lệch chuẩn của mẫu.

+  $n$ : tổng số thành phần của mẫu.

+  $x_i$ : phần tử thứ i của mẫu.

+  $\bar{x}$ : giá trị trung bình của mẫu.

Sử dụng câu lệnh trong R: sd(na.omit(data)).



#### 4.5 Outliers (dữ liệu ngoại lai)

Hiểu đơn giản thì Outliers là một hoặc nhiều cá thể khác hẳn đối với các thành viên còn lại của nhóm. Sự khác biệt này có thể dựa trên nhiều tiêu chí khác nhau như giá trị hạch toán tính.Trong bài này, các giá trị được xem là Outliers được xác định trong công thức ở mục ii-5.

$$IQR = Q3 - Q1$$
$$\text{outliers} < Q1 - 1.5 * IQR \text{ hoặc outliers} > Q3 + 1.5 * IQR$$

Sử dụng câu lệnh trong R: dùm hàm sum để đếm các Outliers, kết hợp với công thức và các toán tử điều kiện.

#### 4.6 Sự tương quan

Hệ số tương quan là chỉ số thống kê đo lường mức độ mạnh yếu của mối quan hệ giữa hai biến số.

Trong đó: hệ số tương quan có giá trị từ -1.0 đến 1.0. Kết quả được tính ra lớn hơn 1.0 hoặc nhỏ hơn -1.0 có nghĩa là có lỗi trong phép đo tương quan.

- Hệ số tương quan có giá trị âm cho thấy hai biến có mối quan hệ nghịch biến hoặc tương quan âm (nghịch biến tuyệt đối khi giá trị bằng -1)
- Hệ số tương quan có giá trị dương cho thấy mối quan hệ đồng biến hoặc tương quan dương (đồng biến tuyệt đối khi giá trị bằng 1)
- Tương quan bằng 0 cho hai biến độc lập với nhau.

Có nhiều loại hệ số tương quan khác nhau nhưng trong bài tập lớn này, ta chỉ đề cập đến hệ số tương quan Pearson. Vì tương quan pearson được biết đến như là phương pháp tốt nhất để đo lường mối liên hệ giữa các biến quan tâm bởi vì nó dựa trên phương pháp hiệp phương sai. Nó cung cấp thông tin về mức độ quan trọng của mối liên hệ, hoặc mối tương quan, cũng như hướng của mối quan hệ. Ngoài ra, việc kiểm tra hệ số tương quan pearson còn giúp chúng ta sớm nhận diện được sự xảy ra của vấn đề đa cộng tuyến khi các biến độc lập có sự tương quan mạnh với nhau.

Cho hai biến số x và y từ n mẫu, hệ số tương quan Pearson được ước tính bằng công thức sau đây:

$$r = \frac{\Sigma(x - \bar{x})(y - \bar{y})}{\sqrt{\Sigma(x - \bar{x})^2 \Sigma(y - \bar{y})^2}}$$

Trong R chúng ta sử dụng hàm `cor(x, y)` để tính hệ số này. Xét giá trị tuyệt đối của hệ số tương quan, ta được mức độ tương quan giữa 2 biến:

- $|r| \leq 0.2$  : tương quan rất yếu.
- $0.2 < |r| \leq 0.5$  : tương quan yếu.
- $0.5 < |r| \leq 0.7$  : tương quan vừa.
- $0.7 < |r| \leq 0.9$  : tương quan mạnh.
- $0.9 < |r|$  : tương quan rất mạnh.



## 5 Bài làm của nhóm

Vì MADE của nhóm là 1204 nên:  $kq = (1 + 2 + 0 + 4) \% 6 = 1$

STT	đất nước	STT	đất nước
1	Kenya	10	Canada
2	Lesotho	11	Greenland
3	Morocco	12	United States
4	Indonesia	13	Australia
5	Japan	14	New Caledonia
6	Vietnam	15	New Zealand
7	Andorra	16	Brazil
8	Slovenia	17	Chile
9	United Kingdom	18	Venezuela

$kq = 1$  nên nhóm sẽ thống kê dữ liệu về COVID-19 liên quan đến các nước có stt là 4,5,6 (Indonesia, Japan, Vietnam).

### i Nhóm câu hỏi liên quan đến tổng quát dữ liệu

Dùng tập dữ liệu để trả lời các câu hỏi và trình bày theo định dạng

```
data = read.csv("owid-covid-data.csv")
```

1) Tập mẫu thể hiện thu thập dữ liệu vào các năm nào

```
temp = data %>% select(date)
rDate = strftime(temp$date, format = "%m/%d/%Y")
year = unique(format(rDate, "%Y"))
cat(year)
```

Kết quả:

```
2020 2021 2022
```

2) Số lượng đất nước và định danh của mỗi đất nước (hiển thị 10 đất nước đầu tiên).

```
i2 = cbind(unique(data %>% select(iso_code)),
           unique(data %>% select(location)))
i2 = i2[1:10,]
colnames(i2) = c("iso_code:", "Country")
i2[11,] = c("Count", 10)
rownames(i2) = NULL
```

Kết quả:

iso_code:	Country
AFG	Afghanistan
OWID_AFR	Africa
ALB	Albania
DZA	Algeria
AND	Andorra
AGO	Angola
AIA	Anguilla
ATG	Antigua and Barbuda
ARG	Argentina
ARM	Armenia
Count	10

3) Số lượng chủng tộc trong tập mẫu



```
i3 = cbind(unique(data %>% select(continent) %>% filter(str_length(continent)
!= 0)))
i3 = arrange(i3, continent)
i3 = cbind(i3, rbind("Châu Phi", "Châu Á", "Châu Âu", "Châu Bắc Mỹ", "Châu Đại
Đường", "Châu Nam Mỹ"))
colnames(i3) = c("Continent:", nrow(i3))
```

Kết quả:

Continent :	6
Africa	Châu Phi
Asia	Châu Á
Europe	Châu Âu
North America	Châu Bắc Mỹ
Oceania	Châu Đại Dương
South America	Châu Nam Mỹ

4) Số lượng dữ liệu thu nhập được trong từng châu lục và tổng số

```
i4 = select(i3, -2)
x = table(data$continent)
i4 = cbind(i4, rbind(x[[2]], x[[3]], x[[4]], x[[5]], x[[6]], x[[7]]))
colnames(i4) = c("Continent:", "Observations")
a = colSums(i4[2])
i4[7,] = c("Tổng:", a)
```

Kết quả:

Continent:	Observations
Africa	38647
Asia	35528
Europe	36375
North America	24438
Oceania	8993
South America	9335
Tổng:	153316

5) Số lượng dữ liệu thu thập được trong từng đất nước (hiển thị 10 đất nước cuối cùng) và tổng số

```
i5 = cbind(unique(data %>% select(iso_code)), unique(data %>% select(location
)))
i5 = i5[(nrow(i5)-9):nrow(i5),]
y = table(data$iso_code)
for (i in 1:10){
  for (k in 1:length(y)){
    if (rownames(y)[k] == i5[i,1]) i5[i,2] = y[[k]]
  }
}
colnames(i5) = c("iso_code", "Observations")
i5[[2]] = as.numeric(i5[[2]])
a = colSums(i5[2])
i5[11,] = c("Tổng:", a)
rownames(i5) = NULL
```

Kết quả:



	iso_code	Observations
	UZB	707
	VUT	467
	VAT	716
	VEN	708
	VNM	759
	WLF	489
	OWID_WRL	760
	YEM	681
	ZMB	704
	ZWE	702
	Tổng:	6693

6) Cho biết các châu lục nào có lượng dữ liệu thu thập nhỏ nhất và giá trị nhó nhất đó?

```
i6 = table(data$continent)
for (i in 1:length(i6)){
  if (i6[[i]] == min(i6)) print(i6[i])
}
```

Kết quả:

```
Oceania
8993
```

7) Cho biết các châu lục nào có lượng dữ liệu thu thập lớn nhất và giá trị lớn nhất đó?

```
i6 = table(data$continent)
for (i in 1:length(i6)){
  if (i6[[i]] == max(i6)) print(i6[i])
}
```

Kết quả:

```
Africa
38647
```

8) Cho biết các nước nào có lượng dữ liệu thu thập nhỏ nhất và giá trị nhó nhất đó?

```
i8 = table(data$location)
for(i in 1:length(i8)){
  if(i8[[i]] == min(i8)) print(i8[i])
}
```

Kết quả:

```
Pitcairn
85
```

9) Cho biết các nước nào có lượng dữ liệu thu thập lớn nhất và giá trị lớn nhất đó?

```
i8 = table(data$location)
for(i in 1:length(i8)){
  if(i8[[i]] == max(i8)) print(i8[i])
}
```

Kết quả:

```
Argentina Mexico
781
```

10) Cho biết các date nào có lượng dữ liệu thu thập nhỏ nhất và giá trị nhó nhất đó?



```
i10 = table(data$date)
for(i in 1:length(i10)){
  if(i10[[i]] == min(i10)) print(i10[i])
}
```

Kết quả:

```
1/1/2020 - 1/3/2020
2
```

- 11) Cho biết các date nào có lượng dữ liệu thu thập lớn nhất và giá trị lớn nhất đó?

```
i10 = table(data$date)
for(i in 1:length(i10)){
  if(i10[[i]] == max(i10)) print(i10[i])
}
```

Kết quả:

```
8/22/2021 - 8/29/2021
238
```

- 12) Cho biết số lượng dữ liệu thu thập được theo date và châu lục.

```
i12 = table(data$date, data$continent)
i12 = i12[order(as.Date(rownames(i12), format="%m/%d/%Y")),]
```

Kết quả:



	Africa	Asia	Europe	North America	Oceania	South America	
1/1/2020	0	0	0	0	1	0	1
1/2/2020	0	0	0	0	1	0	1
1/3/2020	0	0	0	0	1	0	1
1/4/2020	0	0	1	0	1	0	1
1/5/2020	0	0	1	0	1	0	1
1/6/2020	0	0	1	3	1	0	1
1/7/2020	0	0	1	0	1	0	1
1/8/2020	0	0	1	0	1	0	1
1/9/2020	0	0	1	0	1	0	1
1/10/2020	0	0	1	0	1	0	1
1/11/2020	0	0	1	0	1	0	1
1/12/2020	0	0	1	0	1	0	1
1/13/2020	0	0	1	0	1	0	1
1/14/2020	0	0	1	0	1	0	1
1/15/2020	0	0	1	0	1	0	1
1/16/2020	0	0	2	0	1	0	1
1/17/2020	0	0	2	0	1	0	1
1/18/2020	0	0	2	3	1	0	1
1/19/2020	0	0	2	0	1	0	1
1/20/2020	0	0	2	0	1	0	1
1/21/2020	0	0	2	0	1	0	1
1/22/2020	6	0	6	0	2	0	1
1/23/2020	8	0	9	0	2	0	1
1/24/2020	8	0	10	1	2	0	1
1/25/2020	9	0	11	1	2	0	1
1/26/2020	9	0	11	1	3	1	1
1/27/2020	9	0	13	2	3	1	1
1/28/2020	9	0	13	2	3	1	1
1/29/2020	9	0	14	3	3	1	1
1/30/2020	9	0	16	3	3	2	1
1/31/2020	9	0	16	6	3	2	1
2/1/2020	9	0	16	8	3	2	1
2/2/2020	9	0	16	10	3	2	1
2/3/2020	9	0	17	10	3	2	1
2/4/2020	9	0	17	11	3	2	1
2/5/2020	9	0	17	12	3	2	1
2/6/2020	9	0	17	12	3	2	1
2/7/2020	10	2	17	12	3	2	1
2/8/2020	10	2	17	12	3	2	1
2/9/2020	10	2	17	12	3	2	1
2/10/2020	10	2	17	12	3	2	1
2/11/2020	10	2	17	12	3	2	1
2/12/2020	10	2	17	15	3	2	1
2/13/2020	11	2	17	12	3	2	1
2/14/2020	11	3	17	12	3	2	1
2/15/2020	11	3	17	12	3	2	1

Hình 1: Số lượng dữ liệu thu nhập được theo date và châu lục

- 13) Cho biết số lượng dữ liệu thu thập được là lớn nhất theo date và châu lục.

```
cat(max(i12))
```

Kết quả:

```
55
```

- 14) Cho biết số lượng dữ liệu thu thập được là nhỏ nhất theo date và châu lục.

```
cat(min(i12))
```

Kết quả:

```
0
```

- 15) Với một date là k và châu lục t cho trước, hãy cho biết số lượng dữ liệu thể hiện thu thập dữ liệu được.

Các date được viết theo định dạng tháng/ngày/năm từ 1/1/2020 đến 2/19/2022.

Các châu lục sắp theo thứ tự bảng chữ cái tiếng Anh.

Ta có:



1 <= k <= 781 và k = 1 ứng với 1/1/2020  
1 <= t <= 6 và t = 1 ứng với châu Phi (Africa)

Nhóm chọn k = 456 và t = 3.

```
cat(i12[456,3])
```

Kết quả:

```
50
```

- 16) Có đất nước nào mà số lượng dữ liệu thu thập được là bằng nhau không? Hãy cho biết các iso\_code của đất nước đó.

```
i16 = table(data %>% select(iso_code) %>% filter(str_length(iso_code) <= 3))
i16 = sort(i16, decreasing = FALSE)
i = 1
while (i < nrow(i16)){
  temp = c()
  if (i16[[i]] == i16[[i+1]]) {
    temp = cbind(temp, i16[[i]])
    temp = cbind(temp, rownames(i16)[i])
    while (i16[[i]] == i16[[i+1]]){
      i = i + 1
      temp = cbind(temp, rownames(i16)[i])
      if (i == nrow(i16)) break
    }
    cat(temp, '\n')
  }
  i = i + 1
}
```

Kết quả:



686 SPM SSD  
691 BDI SLE  
694 AIA TCA VGB  
697 GNB KNA MLI  
700 DMA GRD MOZ SYR TLS  
701 ERI UGA  
702 AGO CPV HTI IMN MDG NER PNG ZWE  
703 BMU NCL NIC SLV TCD  
704 DJI KGZ MSR MUS ZMB  
705 BRB GMB LBR MNE  
706 BEN BHS GRL SOM TZM  
707 CAF COG GNQ SYC UZB  
708 CUW GAB GHA LCA MRT NAM RWA SUR SWZ TTO VCT VEN  
709 ABW ATG CYM ETH GIN KAZ PYF SDN URY  
710 CUB GUY  
711 BOL CIV COD HND JAM TUR  
712 BFA MNG  
713 BRN PAN  
714 BGR CYP MDA MDV  
716 BTN CMR COL CRI KEN PER SVK VAT  
717 BIH MLT PSE  
718 FRO GIB LBY LIE POL TGO TUN  
719 BDG HUN JOR UKR  
720 AND IDN SAU  
721 AZE CZE DOM ECU PRT  
722 IRL LTU MCO QAT SMR  
723 BLR ISL NGA NZL SEN  
725 BRA GEO LVA MKD ROU SRB  
726 ALB AUT CHE DZA HRV NOR PAK  
727 AFG BHR IRQ KWT LUX OMN  
728 CHL GRC  
744 MAR ZAF  
748 ARM EST  
749 DNK SVN  
750 ESP SWE  
751 GBR ITA RUS  
752 FJI IND PHL  
753 ARE FIN  
755 DEU KHM LKA  
756 AUS CAN  
758 FRA MYS  
759 HKG SGP VNM  
760 CHN JPN KOR MAC USA  
781 ARG MEX

**Hình 2:** Các quốc gia có số lượng dữ liệu thu nhập được là bằng nhau

- 17) Liệt kê iso\_code, tên đất nước mà chiều dài iso\_code lớn hơn 3.

```
i17 = unique(data %>% select(iso_code, location) %>% filter(str_length(iso_code) >3))
```

Kết quả:



iso_code	location
OWID_AFR	Africa
OWID_ASIA	Asia
OWID_EUR	Europe
OWID_EUN	European Union
OWID_HIC	High income
OWID_INT	International
OWID_KOS	Kosovo
OWID_LIC	Low income
OWID_LMC	Lower middle income
OWID_NAM	North America
OWID_CYN	Northern Cyprus
OWID_OCE	Oceania
OWID_SAM	South America
OWID_UMC	Upper middle income
OWID_WRL	World

Hình 3: Các quốc gia có chiều dài iso\_code lớn hơn 3

## ii Nhóm câu hỏi liên quan đến mô tả thống kê cơ bản dữ liệu

Với mỗi quốc gia mà thuộc về nhóm cần tính số liệu thống kê lần lượt cho nhiễm và tử vong do coronavirus được báo cáo mới:

Chọn các nước nằm trong nhóm cần tính (LINK MADE 1204 -> Indonesia, Japan, Vietnam)

Đổi định dạng ngày, tháng, năm từ Character sang Date, đồng thời cung cấp cho R biết định dạng theo kiểu tháng/ngày-năm.

Đổi các giá trị âm thành giá trị dương trong mục New\_Cases và Deaths\_Cases.

```
data$date = as.Date(data$date, format = "%m/%d/%Y")
data$new_cases = abs(data$new_cases)
data$new_deaths = abs(data$new_deaths)

indo = subset(data, data$location == "Indonesia")
japan = subset(data, data$location == "Japan")
vietnam = subset(data, data$location == "Vietnam")
```

Sau đó tính riêng từng thành phần cho ca nhiễm, tử vong

1) Tính giá trị nhỏ nhất, lớn nhất.

Dùng câu lệnh na.omit để lọc bỏ các dữ liệu NA.

Đối với số ca nhiễm:

```
indo.cases.MIN = min(na.omit(indo$new_cases))
indo.cases.MAX = max(na.omit(indo$new_cases))
japan.cases.MIN = min(na.omit(japan$new_cases))
japan.cases.MAX = max(na.omit(japan$new_cases))
vietnam.cases.MIN = min(na.omit(vietnam$new_cases))
vietnam.cases.MAX = max(na.omit(vietnam$new_cases))
```

Đối với số ca tử vong:

```
indo.deaths.MIN = min(na.omit(indo$new_deaths))
indo.deaths.MAX = max(na.omit(indo$new_deaths))
japan.deaths.MIN = min(na.omit(japan$new_deaths))
japan.deaths.MAX = max(na.omit(japan$new_deaths))
vietnam.deaths.MIN = min(na.omit(vietnam$new_deaths))
vietnam.deaths.MAX = max(na.omit(vietnam$new_deaths))
```

2) Tính tứ phân vị thứ nhất(Q1), thứ hai(Q2), thứ ba(Q3)



```
indo.cases.Q1 = quantile(na.omit(indo$new_cases),c(0.25))
indo.cases.Q2 = quantile(na.omit(indo$new_cases),c(0.5))
indo.cases.Q3 = quantile(na.omit(indo$new_cases),c(0.75))
indo.deaths.Q1 = quantile(na.omit(indo$new_deaths),c(0.25))
indo.deaths.Q2 = quantile(na.omit(indo$new_deaths),c(0.5))
indo.deaths.Q3 = quantile(na.omit(indo$new_deaths),c(0.75))

japan.cases.Q1 = quantile(na.omit(japan$new_cases),c(0.25))
japan.cases.Q2 = quantile(na.omit(japan$new_cases),c(0.5))
japan.cases.Q3 = quantile(na.omit(japan$new_cases),c(0.75))
japan.deaths.Q1 = quantile(na.omit(japan$new_deaths),c(0.25))
japan.deaths.Q2 = quantile(na.omit(japan$new_deaths),c(0.5))
japan.deaths.Q3 = quantile(na.omit(japan$new_deaths),c(0.75))

vietnam.cases.Q1 = quantile(na.omit(vietnam$new_cases),c(0.25))
vietnam.cases.Q2 = quantile(na.omit(vietnam$new_cases),c(0.5))
vietnam.cases.Q3 = quantile(na.omit(vietnam$new_cases),c(0.75))
vietnam.deaths.Q1 = quantile(na.omit(vietnam$new_deaths),c(0.25))
vietnam.deaths.Q2 = quantile(na.omit(vietnam$new_deaths),c(0.5))
vietnam.deaths.Q3 = quantile(na.omit(vietnam$new_deaths),c(0.75))
```

3) Tính giá trị trung bình (Avg)

```
indo.cases.avg = mean(na.omit(indo$new_cases))
indo.deaths.avg = mean(na.omit(indo$new_deaths))

japan.cases.avg = mean(na.omit(japan$new_cases))
japan.deaths.avg = mean(na.omit(japan$new_deaths))

vietnam.cases.avg = mean(na.omit(vietnam$new_cases))
vietnam.deaths.avg = mean(na.omit(vietnam$new_deaths))
```

4) Tính giá trị độ lệch chuẩn (Std)

```
indo.cases.std = sd(na.omit(indo$new_cases))
indo.deaths.std = sd(na.omit(indo$new_deaths))

japan.cases.std = sd(na.omit(japan$new_cases))
japan.deaths.std = sd(na.omit(japan$new_deaths))

vietnam.cases.std = sd(na.omit(vietnam$new_cases))
vietnam.deaths.std = sd(na.omit(vietnam$new_deaths))
```

5) Dếm xem có bao nhiêu outliers, một quan sát mà giá trị của nó nằm trong khoảng sau:

$$IQR = Q3 - Q1$$
$$\text{outliers} < Q1 - 1.5 * IQR \text{ hoặc } \text{outliers} > Q3 + 1.5 * IQR$$

```
indo.cases.IQR = indo.cases.Q3 - indo.cases.Q1

indo.cases.outlier = sum(na.omit
    (indo$new_cases < indo.cases.Q1 - 1.5*indo.cases.IQR
    | indo$new_cases > indo.cases.Q3 + 1.5*indo.cases.IQR))

indo.deaths.IQR = indo.deaths.Q3 - indo.deaths.Q1

indo.deaths.outlier = sum(na.omit
    (indo$new_deaths < indo.deaths.Q1 - 1.5*indo.deaths.IQR
    | indo$new_deaths > indo.deaths.Q3 + 1.5*indo.deaths.IQR))

japan.cases.IQR = japan.cases.Q3 - japan.cases.Q1

japan.cases.outlier = sum(na.omit
```



```
(japan$new_cases < japan.cases.Q1 - 1.5*japan.cases.IQR  
| japan$new_cases > japan.cases.Q3 + 1.5*japan.cases.IQR))  
  
japan.deaths.IQR = japan.deaths.Q3 - japan.deaths.Q1  
  
japan.deaths.outlier = sum(na.omit  
(japan$new_deaths < japan.deaths.Q1 - 1.5*japan.deaths.IQR  
| japan$new_deaths > japan.deaths.Q3 + 1.5*japan.deaths.IQR))  
  
vietnam.cases.IQR = vietnam.cases.Q3 - vietnam.cases.Q1  
  
vietnam.cases.outlier = sum(na.omit  
(vietnam$new_cases < vietnam.cases.Q1 - 1.5*vietnam.cases.IQR  
| vietnam$new_cases > vietnam.cases.Q3 + 1.5*vietnam.cases.IQR))  
  
vietnam.deaths.IQR = vietnam.deaths.Q3 - vietnam.deaths.Q1  
  
vietnam.deaths.outlier = sum(na.omit  
(vietnam$new_deaths < vietnam.deaths.Q1 - 1.5*vietnam.deaths.IQR  
| vietnam$new_deaths > vietnam.deaths.Q3 + 1.5*vietnam.deaths.IQR))
```

6) Lập bảng mô tả số liệu thống kê cho từng đất nước thuộc về nhóm:

```
print ("New_Cases(infections)")  
data.frame(  
Country = c("Indonesia","Japan","Vietnam"),  
Min = c(indo.cases.MIN,japan.cases.MIN,vietnam.cases.MIN),  
Q1 = c(indo.cases.Q1,japan.cases.Q1,vietnam.cases.Q1),  
Q2 = c(indo.cases.Q2,japan.cases.Q2,vietnam.cases.Q2),  
Q3 = c(indo.cases.Q3,japan.cases.Q3,vietnam.cases.Q3),  
Max = c(indo.cases.MAX,japan.cases.MAX,vietnam.cases.MAX),  
Avg = c(indo.cases.avg,japan.cases.avg,vietnam.cases.avg),  
Std = c(indo.cases.std,japan.cases.std,vietnam.cases.std),  
Outlier = c(indo.cases.outlier,japan.cases.outlier, vietnam.cases.outlier)  
)
```

Chạy đoạn code trên ta được kết quả như sau:

New\_Cases(infections)

Countries	Min	Q1	Q2	Q3	Max	Avg	Std	Outlier
Indonesia	0	766	3874	6816.5	64718	7078.772	10904.261	80
Japan	0	225	1032	3342.5	104345	5822.466	16231.866	87
Vietnam	0	1	10	4758.0	54830	3610.399	6917.646	102

```
print ("New_Deaths(deaths)")  
data.frame(  
Country = c("Indonesia","Japan","Vietnam"),  
Min = c(indo.deaths.MIN,japan.deaths.MIN,vietnam.deaths.MIN),  
Q1 = c(indo.deaths.Q1,japan.deaths.Q1,vietnam.deaths.Q1),  
Q2 = c(indo.deaths.Q2,japan.deaths.Q2,vietnam.deaths.Q2),  
Q3 = c(indo.deaths.Q3,japan.deaths.Q3,vietnam.deaths.Q3),  
Max = c(indo.deaths.MAX,japan.deaths.MAX,vietnam.deaths.MAX),  
Avg = c(indo.deaths.avg,japan.deaths.avg,vietnam.deaths.avg),  
Std = c(indo.deaths.std,japan.deaths.std,vietnam.deaths.std),  
Outlier = c(indo.deaths.outlier,japan.deaths.outlier,vietnam.deaths.outlier)  
)
```

Chạy đoạn code trên ta được kết quả như sau:

New\_Deaths(deaths)

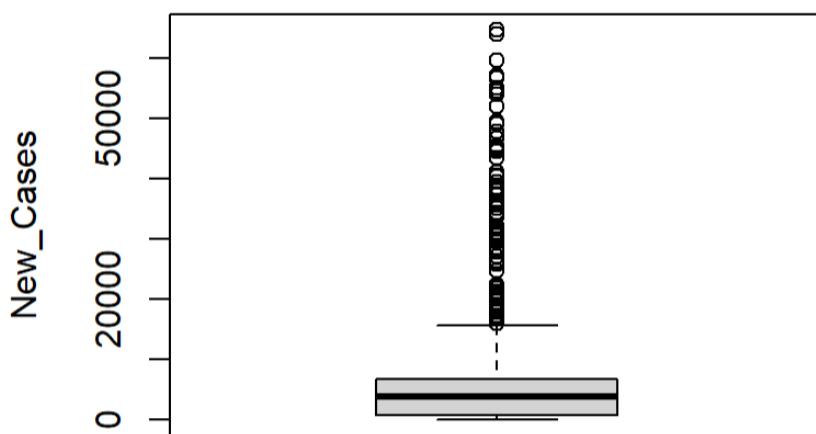
Countries	Min	Q1	Q2	Q3	Max	Avg	Std	Outlier
Indonesia	0	33	100	187	2069	205.62869	348.46457	74
Japan	0	4	14	46	271	29.38347	36.63266	27
Vietnam	0	0	0	113	804	69.28822	116.45448	36

7) Vẽ biểu đồ boxplot hay còn được gọi là box-and-whisker cho nhiễm coronavirus

```
boxplot(indo$new_cases, ylab="New_Cases", main="Indonesia New Cases Boxplot")
```

Kết quả:

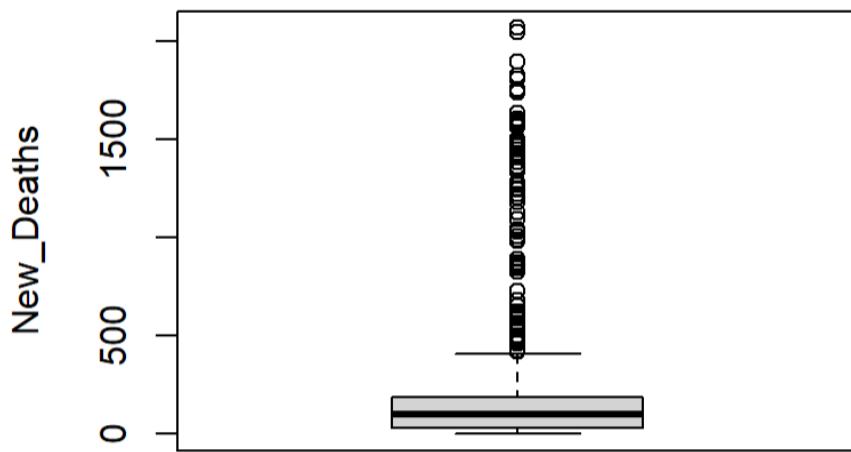
**Indonesia New Cases Boxplot**



```
boxplot(indo$new_deaths, ylab="New_Deaths", main="Indonesia New Deaths Boxlot")
```

Kết quả:

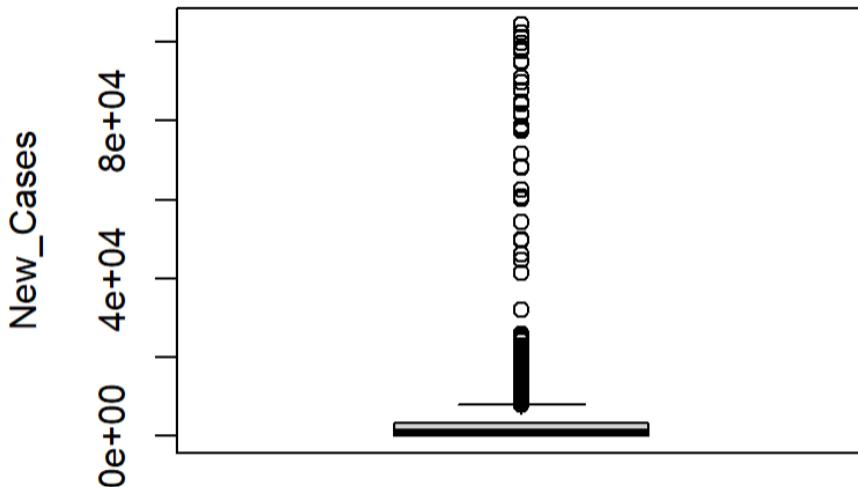
## Indonesia New Deaths Boxplot



```
boxplot(japan$new_cases, ylab="New_Cases", main="Japan New Cases Boxplot")
```

Kết quả:

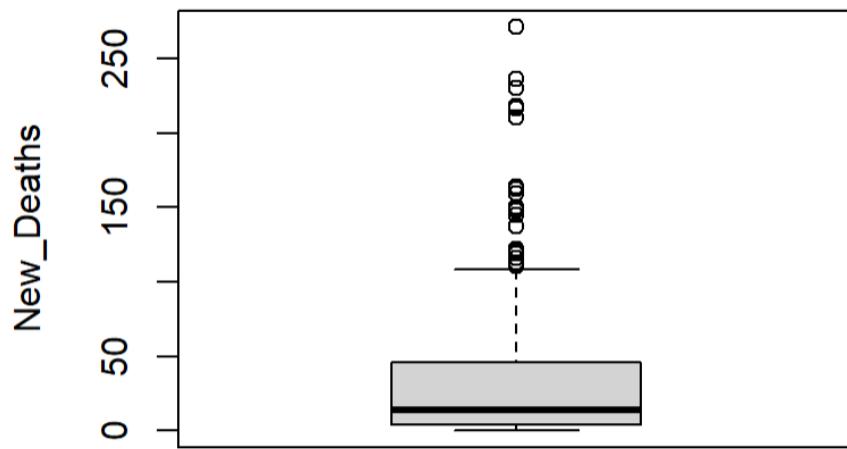
## Japan New Cases Boxplot



```
boxplot(japan$new_deaths, ylab="New_Deaths", main="Japan New Deaths Boxplot")
```

Kết quả:

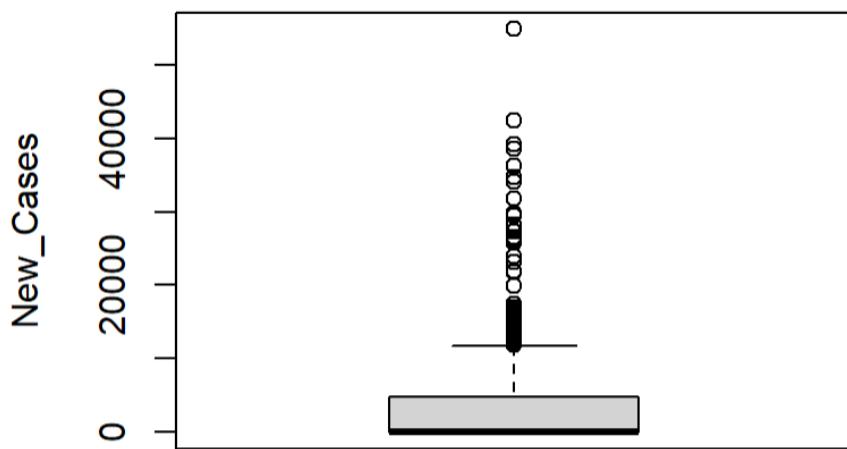
### Japan New Deaths Boxplot



```
boxplot(vietnam$new_cases, ylab="New_Cases", main="Vietnam New Cases Boxplot")
```

Kết quả:

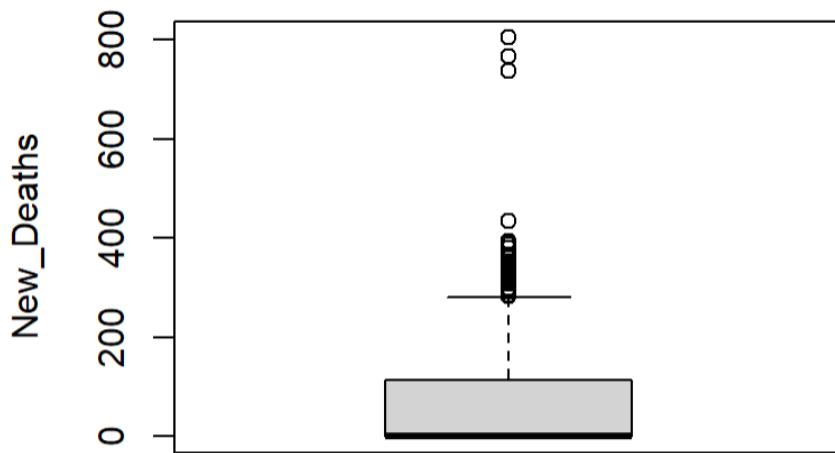
### Vietnam New Cases Boxplot



```
boxplot(vietnam$new_deaths, ylab="New_Deaths", main="Vietnam New Deaths Boxplot")
```

Kết quả:

## Vietnam New Deaths Boxplot



### iii Nhóm câu hỏi liên quan đến dữ liệu thể hiện thu thập dữ liệu

Với mỗi quốc gia mà thuộc về nhóm cần tính số liệu thống kê lần lượt cho nhiễm và tử vong do coronavirus:

Cần phân biệt rõ các khái niệm sau:

+Dữ liệu không được báo cáo mới: giá trị bằng NA hoặc 0.

+Không có dữ liệu được báo cáo: giá trị bằng NA.

+Không có người nhiễm bệnh mới: giá trị bằng 0.

- 1) Có bao nhiêu ngày có số lần dữ liệu không được báo cáo mới.

Ta lọc số liệu chọn các hàng có New\_Cases hoặc New\_Deaths bằng 0 hoặc bằng NA

```
indo.cases.No.Report = sum(indo$new_cases==0|is.na(indo$new_cases))
indo.deaths.No.Report = sum(indo$new_deaths==0|is.na(indo$new_deaths))
japan.cases.No.Report = sum(japan$new_cases==0|is.na(japan$new_cases))
japan.deaths.No.Report = sum(japan$new_deaths==0|is.na(japan$new_deaths))
vietnam.cases.No.Report = sum(vietnam$new_cases==0|is.na(vietnam$new_cases))
)
vietnam.deaths.No.Report = sum(vietnam$new_deaths==0 |
is.na(vietnam$new_deaths))
```

- 2) Có bao nhiêu ngày có số ca nhiễm/ tử vong là thấp nhất được báo cáo mới.

```
indo.cases.Report = subset(indo,indo$new_cases!=0)
indo.cases.Report.MIN = min(na.omit(indo.cases.Report$new_cases))
indo.count.day.cases.MIN = sum(na.omit(indo.cases.Report$new_cases==
indo.cases.Report.MIN))

indo.deaths.Report = subset(indo,indo$new_deaths!=0)
indo.deaths.Report.MIN = min(na.omit(indo.deaths.Report$new_deaths))
indo.count.day.deaths.MIN = sum(na.omit(indo$new_deaths==
indo.deaths.Report.MIN))

japan.cases.Report = subset(japan,japan$new_cases!=0)
japan.cases.Report.MIN = min(na.omit(japan.cases.Report$new_cases))
japan.count.day.cases.MIN = sum(na.omit(japan.cases.Report$new_cases==
japan.cases.Report.MIN))

japan.deaths.Report = subset(japan,japan$new_deaths!=0)
japan.deaths.Report.MIN = min(na.omit(japan.deaths.Report$new_deaths))
```



```
japan.count.day.deaths.MIN = sum(na.omit(japan$new_deaths==  
japan.deaths.Report.MIN))  
  
vietnam.cases.Report = subset(vietnam,vietnam$new_cases!=0)  
vietnam.cases.Report.MIN = min(na.omit(vietnam.cases.Report$new_cases))  
vietnam.count.day.cases.MIN = sum(na.omit(vietnam.cases.Report$new_cases==  
vietnam.cases.Report.MIN))  
  
vietnam.deaths.Report = subset(vietnam,vietnam$new_deaths!=0)  
vietnam.deaths.Report.MIN = min(na.omit(vietnam.deaths.Report$new_deaths))  
vietnam.count.day.deaths.MIN = sum(na.omit(vietnam$new_deaths==  
vietnam.deaths.Report.MIN))
```

- 3) Có bao nhiêu ngày có số ca nhiễm/ tử vong là cao nhất được báo cáo mới

```
indo.cases.Report.MAX = max(na.omit(indo.cases.Report$new_cases))  
indo.count.day.cases.MAX = sum(na.omit(indo.cases.Report$new_cases==  
indo.cases.Report.MAX))  
  
indo.deaths.Report.MAX = max(na.omit(indo.deaths.Report$new_deaths))  
indo.count.day.deaths.MAX = sum(na.omit(indo$new_deaths==  
indo.deaths.Report.MAX))  
  
japan.cases.Report.MAX = max(na.omit(japan.cases.Report$new_cases))  
japan.count.day.cases.MAX = sum(na.omit(japan.cases.Report$new_cases==  
japan.cases.Report.MAX))  
  
japan.deaths.Report.MAX = max(na.omit(japan.deaths.Report$new_deaths))  
japan.count.day.deaths.MAX = sum(na.omit(japan$new_deaths==  
japan.deaths.Report.MAX))  
  
vietnam.cases.Report.MAX = max(na.omit(vietnam.cases.Report$new_cases))  
vietnam.count.day.cases.MAX = sum(na.omit(vietnam.cases.Report$new_cases==  
vietnam.cases.Report.MAX))  
  
vietnam.deaths.Report.MAX = max(na.omit(vietnam.deaths.Report$new_deaths))  
vietnam.count.day.deaths.MAX = sum(na.omit(vietnam$new_deaths==  
vietnam.deaths.Report.MAX))
```

- 4) Thể hiện bảng số liệu như sau:

```
print("No_Report")  
data.frame(  
  Countries = c("Indonesia", "Japan", "Vietnam"),  
  Infections = c(indo.cases.No.Report, japan.cases.No.Report,  
                 vietnam.cases.No.Report),  
  Deaths = c(indo.deaths.No.Report, japan.deaths.No.Report,  
             vietnam.deaths.No.Report))
```

No\_Report

Countries	Infections	Deaths
Indonesia	8	15
Japan	11	77
Vietnam	139	489

Báo cáo mới:



```
print ("Report_MIN")
data.frame(
Countries = c("Indonesia","Japan","Vietnam"),
Infections = c(indo.count.day.cases.MIN,japan.count.day.cases.MIN,
vietnam.count.day.cases.MIN),
Deaths = c(indo.count.day.deaths.MIN,japan.count.day.deaths.MIN,
vietnam.count.day.deaths.MIN)
)
```

### Report\_MIN

Countries	Infections	Deaths
Indonesia	3	5
Japan	4	54
Vietnam	64	27

```
print ("Report_MAX")
data.frame(
Countries = c("Indonesia","Japan","Vietnam"),
Infections = c(indo.count.day.cases.MAX,japan.count.day.cases.MAX,
vietnam.count.day.cases.MAX),
Deaths = c(indo.count.day.deaths.MAX,japan.count.day.deaths.MAX,
vietnam.count.day.deaths.MAX)
)
```



## Report\_MAX

Countries	Infections	Deaths
Indonesia	1	1
Japan	1	1
Vietnam	1	1

Xây dựng hàm check.day.MIN để tính số ngày ngắn nhất liên tiếp

```
check.day.MIN = function(x){  
  if (length(x)==1) print(1)  
  else if (length(x)<1) print (0)  
  else {  
    min <- length(x)  
    count <- 1  
    for (i in 2:length(x)){  
      if(as.numeric(x[i]-x[i-1])==1){  
        count <- count + 1  
      }  
      else {  
        if (count<min) {  
          min <- count  
        }  
        count <- 1  
      }  
    }  
    min  
  }  
}
```

Xây dựng hàm check.day.MAX để tính số ngày dài nhất liên tiếp

```
check.day.MAX = function(x){  
  if (length(x)==1) print(1)  
  else if (length(x)<1) print (0)  
  else {  
    max <- 1  
    count <- 1  
    for (i in 2:length(x)){  
      if(as.numeric(x[i]-x[i-1])==1){  
        count <- count + 1  
      }  
      else {  
        if (count>max) {  
          max <- count  
        }  
        count <-1  
      }  
      if (count>max) {  
        max <- count  
      }  
    }  
    max  
  }  
}
```

Dùng 2 hàm đã xây dựng ở trên kết hợp với lọc dữ liệu theo từng yêu cầu ta tính được các ý còn lại trong mục iii.

- 5) Cho biết số ngày ngắn nhất liên tiếp mà không có dữ liệu được báo cáo

```
indo.cases.unupdate = subset(indo, is.na(indo$new_cases))  
japan.cases.unupdate = subset(japan, is.na(japan$new_cases))
```



```
vietnam.cases.unupdate = subset(vietnam, is.na(vietnam$new_cases))

indo.deaths.unupdate = subset(indo, is.na(indo$new_deaths))
japan.deaths.unupdate = subset(japan, is.na(japan$new_deaths))
vietnam.deaths.unupdate = subset(vietnam, is.na(vietnam$new_deaths))

check.day.MIN(indo.deaths.unupdate$date)
check.day.MIN(indo.cases.unupdate$date)
check.day.MIN(japan.deaths.unupdate$date)
check.day.MIN(japan.cases.unupdate$date)
check.day.MIN(vietnam.deaths.unupdate$date)
check.day.MIN(vietnam.cases.unupdate$date)
```

Kết quả:

Countries	Min Infections	Min Deaths
Indonesia	1	9
Japan	1	22
Vietnam	0	190

- 6) Cho biết số ngày dài nhất liên tiếp mà không có dữ liệu được báo cáo

```
check.day.MAX(indo.deaths.unupdate$date)
check.day.MAX(indo.cases.unupdate$date)
check.day.MAX(japan.deaths.unupdate$date)
check.day.MAX(japan.cases.unupdate$date)
check.day.MAX(vietnam.deaths.unupdate$date)
check.day.MAX(vietnam.cases.unupdate$date)
```

Kết quả:

Countries	Max Infections	Max Deaths
Indonesia	1	9
Japan	1	22
Vietnam	0	190

- 7) Cho biết số ngày ngắn nhất liên tiếp mà không có người nhiễm bệnh mới

```
indo.zero.cases = subset(indo, indo$new_cases == '0')
japan.zero.cases = subset(japan, japan$new_cases == '0')
vietnam.zero.cases = subset(vietnam, vietnam$new_cases == '0')

check.day.MIN(indo.zero.cases$date)
check.day.MIN(japan.zero.cases$date)
check.day.MIN(vietnam.zero.cases$date)
```

- 8) Cho biết số ngày dài nhất liên tiếp mà không có người nhiễm bệnh mới

```
check.day.MAX(indo.zero.cases$date)
check.day.MAX(japan.zero.cases$date)
check.day.MAX(vietnam.zero.cases$date)
```

Kết quả của phần 7 và 8:

Countries	Min No Infections	Max No Infections
Indonesia	1	3
Japan	1	3
Vietnam	1	22



#### iv Nhóm câu hỏi liên quan đến trực quan dữ liệu

Xử lý số liệu:

- Khai báo thư viện và nhập dữ liệu vào R:

```
pacman::p_load(rio, readr, here, ggplot2, dplyr, tidyverse, extrafont,
                janitor, zoo, skimr, tidyverse, base, lubridate, scales,
                cowplot)
file_raw <- import("owid-covid-data.csv")
```

- Xử lý dữ liệu cơ bản R: Chuyển đổi biến *date* về kiểu dữ liệu ngày tháng năm:

```
file_raw <- file_raw %>% mutate(date = lubridate::mdy(date))
```

- Tạo biến *countryName*, chứa 2 biến *continent* và *location* từ dữ liệu gốc, loại bỏ trùng lặp và các dữ liệu không kỳ vọng. Ta được một tập dữ liệu chỉ chứa 2 cột: Châu lục và quốc gia ứng với châu lục đó.

```
countryName <- file_raw %>% select(continent, location)
countryName <- countryName %>% distinct()
countryName <- countryName %>% filter(continent != "")
```

- Dùng hàm *tabyl()* để thống kê số lượng quốc gia của mỗi châu lục trong tập dữ liệu, và tính tỷ lệ.

```
nCountry <- countryName %>% tabyl(continent)
```

- Xử lý dữ liệu cho câu 3 và câu 4

Dùng hàm *filter()* để chọn hàng theo tên quốc gia, sau đó chọn 7 hàng cuối, tương ứng với tuần cuối cùng.

```
sevVie = file_raw %>% filter(location == "Vietnam")
sevVie = sevVie[(nrow(sevVie)-6):nrow(sevVie), 4:6]
sevInd = file_raw %>% filter(location == "Indonesia")
sevInd = sevInd[(nrow(sevInd)-6):nrow(sevInd), 4:6]
sevJap = file_raw %>% filter(location == "Japan")
sevJap = sevJap[(nrow(sevJap)-6):nrow(sevJap), 4:6]
```

- Xử lý dữ liệu cho 2 câu 5 và 6:

Lọc dữ liệu theo các nhóm và loại bỏ các Missing Value:

```
data_VietNam <- file_raw %>% filter(location == "Vietnam")
data_VietNam <- data_VietNam %>% mutate(new_deaths_dips = replace_na(new_deaths, 0))
data_Indo <- file_raw %>% filter(location == "Indonesia")
data_Indo <- data_Indo %>% mutate(new_deaths_dips = replace_na(new_deaths, 0))
data_Japan <- file_raw %>% filter(location == "Japan")
data_Japan <- data_Japan %>% mutate(new_deaths_dips = replace_na(new_deaths, 0))
```

### Vẽ biểu đồ:

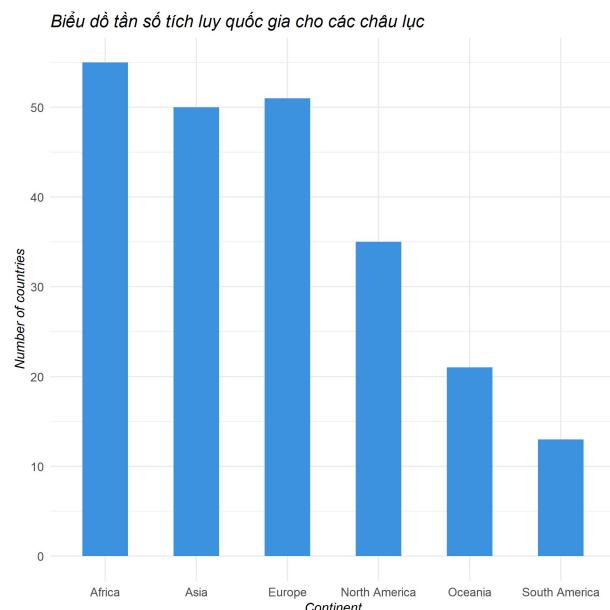
- 1 Vẽ biểu đồ tần số tích lũy quốc gia cho các châu lục.

Ta dùng hàm `ggplot` để vẽ biểu đồ kết hợp với hàm `geom_col` để vẽ biểu đồ cột và gán vào biến `nCountry_plot`

Code:

```
nCountry_plot <- ggplot(data = nCountry, aes(x = continent, y = n)) +  
  geom_col(fill = "#3c92de", width = 0.5) + theme_minimal() +  
  labs(  
    x = "Continent",  
    y = "Number of countries",  
    title = "Bieu do tan so tich luy quoc gia cho cac chau luc"  
) + theme(plot.title = element_text(size = 13, face = "italic"),  
         axis.title.x = element_text(size = 10, face = "italic"),  
         axis.title.y = element_text(size = 10, face = "italic")) +  
  scale_y_continuous(  
    breaks = seq(  
      from = 0,  
      to = 60,  
      by = 10  
)  
)  
nCountry_plot
```

### Kết quả:



- 2 Vẽ biểu đồ tần số tương đối quốc gia cho các châu lục

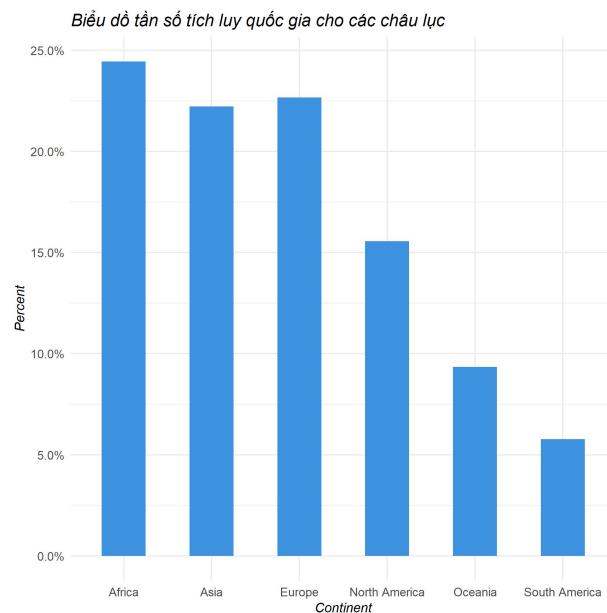
Ở đây, ta chỉ cần thay đổi từ biến số lượng thành biến tỷ lệ.

Code:

```
ratCountry_plot <- ggplot(data = nCountry, aes(x = continent, y = percent)) +  
  geom_col(fill = "#3c92de", width = 0.5) + theme_minimal() +
```

```
labs(  
  x = "Continent",  
  y = "Percent",  
  title = "Bieu do tan so tuong doi tich luy quoc gia cho cac chau luc"  
) + theme(plot.title = element_text(size = 13, face = "italic"),  
           axis.title.x = element_text(size = 10, face = "italic"),  
           axis.title.y = element_text(size = 10, face = "italic"))+  
scale_y_continuous()  
labels = scales::percent  
)
```

Kết quả:

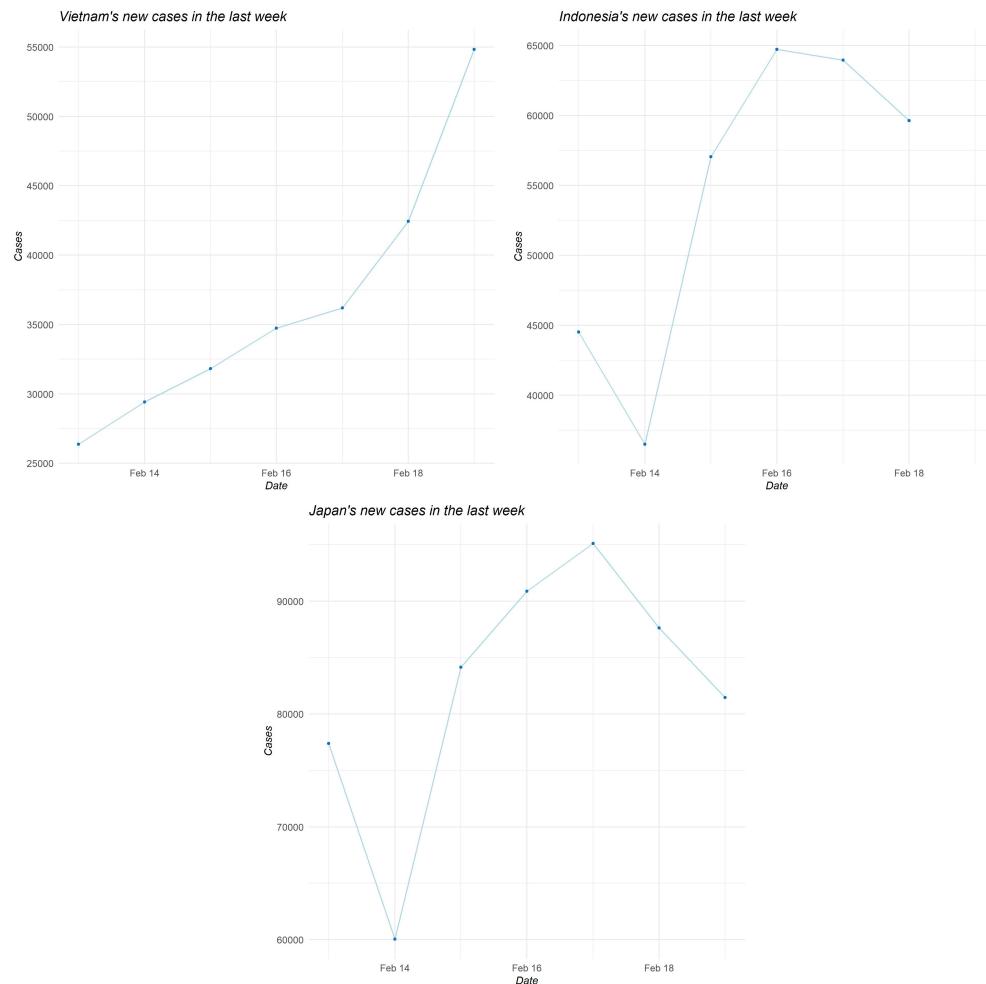


3 Vẽ biểu đồ thể hiện nhiễm bệnh đã báo cáo của các quốc gia mà thuộc về nhóm trong 7 ngày cuối của năm cuối cùng

**Code:** Ở đây, code đại diện cho Việt Nam, các quốc gia còn lại làm tương tự.

```
sevVie_Pcase<-ggplot(data = sevVie, aes(x = date, y = new_cases))+  
  geom_line(color = "lightblue") +  
  geom_point(size = 1, color = "#0871c2") +  
  labs(x = "Date", y = "Cases", title = "Vietnam's new deaths in the last week  
  ") +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 13, face = "italic"),  
        axis.title.x = element_text(size = 10, face = "italic"),  
        axis.title.y = element_text(size = 10, face = "italic"))
```

### Kết quả:

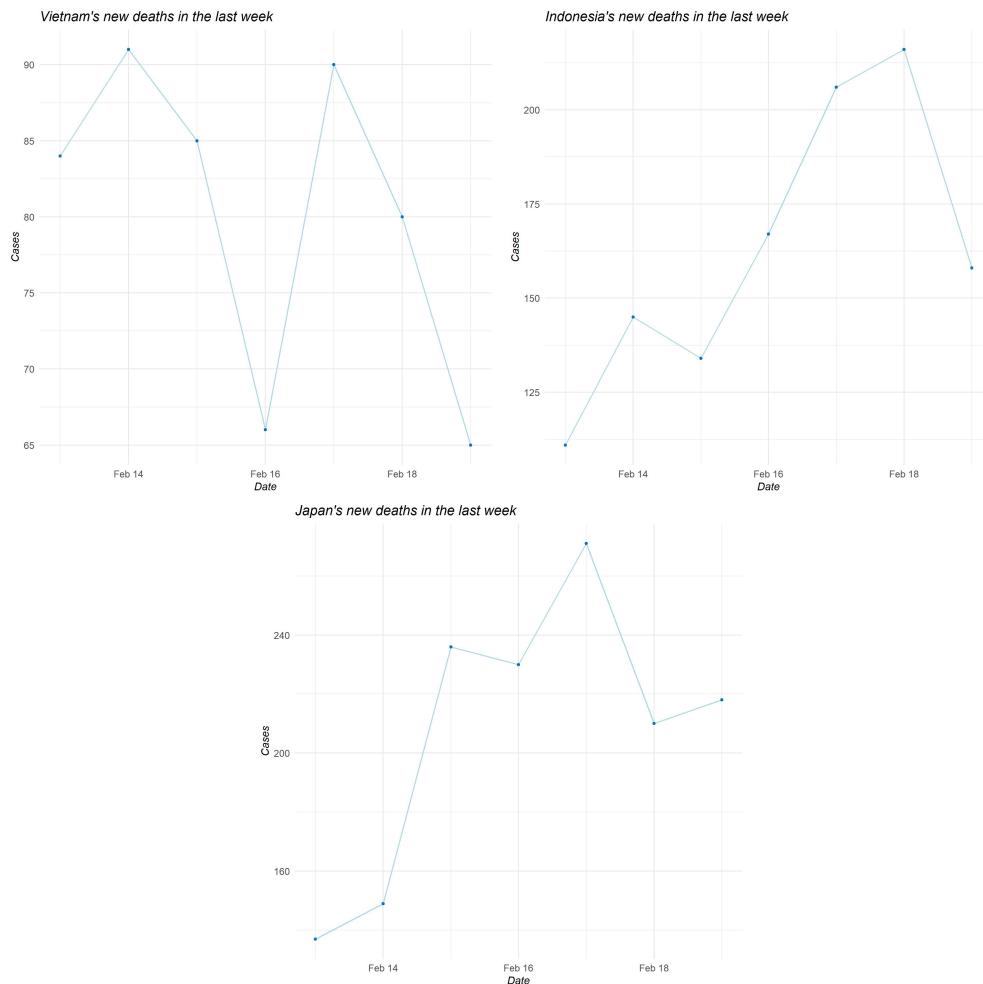


- 4 Vẽ biểu đồ thể hiện tử vong đã báo cáo của các quốc gia mà thuộc về nhóm trong 7 ngày cuối của năm cuối cùng

**Code:** Ở đây, code đại diện cho Việt Nam, các quốc gia còn lại làm tương tự.

```
sevVie_Pdead <- ggplot(data = sevVie, aes(x = date, y = new_deaths))+
  geom_line(color = "lightblue") +
  geom_point(size = 1, color = "#0871c2") +
  labs(x = "Date", y = "Cases", title = "Vietnam's new deaths in the last week
  ") +
  theme_minimal() +
  theme(plot.title = element_text(size = 13, face = "italic"),
        axis.title.x = element_text(size = 10, face = "italic"),
        axis.title.y = element_text(size = 10, face = "italic"))
```

### Kết quả:



5 Vẽ biểu đồ phổ đất nước xuất hiện outliers cho nhiễm bệnh

6 Vẽ biểu đồ phổ đất nước xuất hiện outliers cho tử vong

### Bài giải cho 2 câu 5 và 6:

Sử dụng `ggplot()` và hàm `geom_histogram` để vẽ biểu đồ và lưu biểu đồ vào biến `newcases_Tenquocgia` và `newdeaths_Tenquocgia` để tiện cho việc lưu trữ biểu đồ: **Code:** Ví dụ cho 1 nước, các nước còn lại làm tương tự:

```
newcasesVie <- ggplot(data = data_VietNam, aes(x = new_cases)) + labs(
  x = "New case",
  y = "Count",
  title = "Bieu do pho cua Viet Nam cho nhiem benh"
) +
  theme_minimal() +
  geom_histogram(bins = 20, fill = "#3c92de") +
  scale_y_continuous(
    breaks = seq (
      from = 0,
      to = 600,
      by = 20
    )
  ) +
```

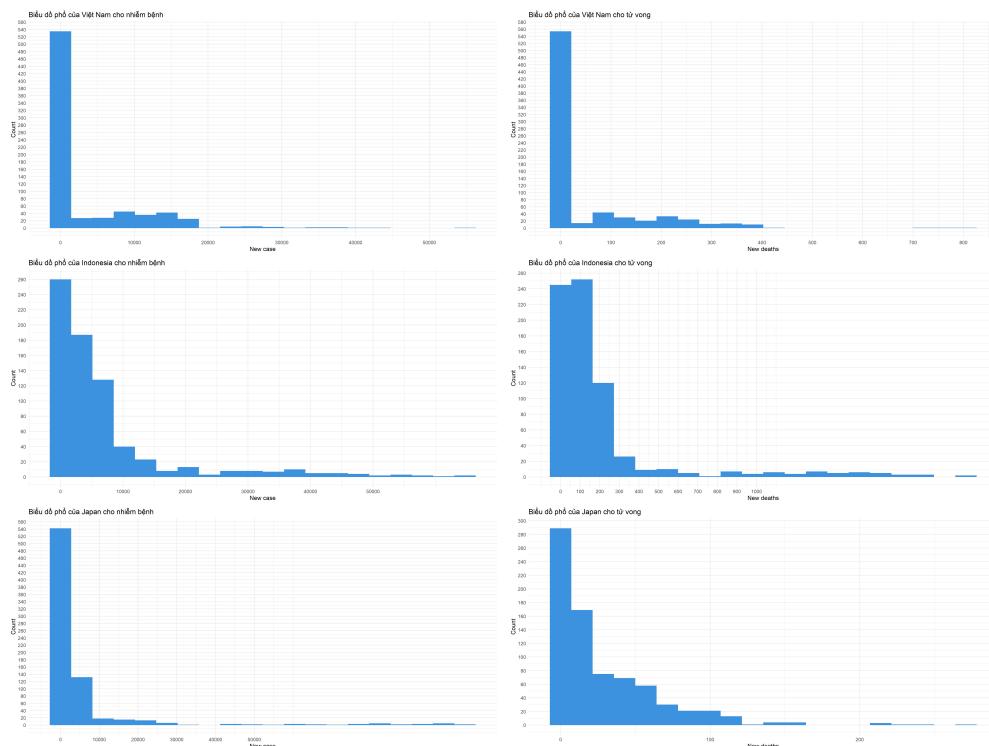
```

scale_x_continuous(
  breaks = seq(
    from = 0,
    to = 55000,
    by = 10000
  )
)

newdeathsVie <- ggplot(data = data_VietNam, aes(x = new_deaths_dips)) + labs(
  x = "New deaths",
  y = "Count",
  title = "Bieu do pho cua Viet Nam cho tu vong"
) +
  theme_minimal() +
  geom_histogram(bins = 20, fill = "#3c92de") +
  scale_y_continuous(
    breaks = seq (
      from = 0,
      to = 600,
      by = 20
    )
  ) +
  scale_x_continuous(
    breaks = seq(
      from = 0,
      to = 1000,
      by = 100
    )
  )
)

```

Kết quả:



## v Nhóm câu hỏi liên quan đến trực quan dữ liệu theo thời gian là tháng

Vì *MADE* của nhóm là 1024 nên ta sẽ phân tích dữ liệu trong các tháng 1, 2, 4, 10

### Xử lý dữ liệu:

- Khai báo thư viện, nhập dữ liệu và chuyển đổi biến **date** thành kiểu ngày tháng năm: **Đã làm ở câu iv**
- Dùng hàm **filter()** chọn các hàng có *iso\_code* là *OWID\_WRL*, đây là dữ liệu COVID-19 thống kê trên toàn thế giới cho vào biến **world**, sau đó chia ra thành các năm 2020, 2021, 2022 tương ứng với các biến **twenty**, **twenty1**, **twenty2** và các tháng trong từng năm.

```
world <- file_raw %>% filter(iso_code == "OWID_WRL")
twenty <- world %>% filter(year(date) == 2020)
twenty1 <- world %>% filter(year(date) == 2021)
twenty2 <- world %>% filter(year(date) == 2022)
#2020
Jan2020 <- twenty %>% filter(month(date) == 1)
Feb2020 <- twenty %>% filter(month(date) == 2)
Apr2020 <- twenty %>% filter(month(date) == 4)
Oct2020 <- twenty %>% filter(month(date) == 10)
#2021
Jan2021 <- twenty1 %>% filter(month(date) == 1)
Feb2021 <- twenty1 %>% filter(month(date) == 2)
Apr2021 <- twenty1 %>% filter(month(date) == 4)
Oct2021 <- twenty1 %>% filter(month(date) == 10)
#2022
Jan2022 <- twenty2 %>% filter(month(date) == 1)
Feb2022 <- twenty2 %>% filter(month(date) == 2)
```

- Xử lý dữ liệu cho 3 câu 4,5,6:

Ta quan tâm đến 2 tháng cuối của mỗi năm, dùng hàm **filter()** để chọn dữ liệu 2 tháng cuối năm (11,12) của 2 năm 2020, 2021 và lưu vào 2 biến **NovDec2020**,**NovDec2021**

```
NovDec2020 <- twenty %>% filter(month(date) == 11 | month(date) == 12)
NovDec2021 <- twenty1 %>% filter(month(date) == 11 | month(date) == 12)
```

- Xử lý dữ liệu cho 2 câu 7 và 8:

Ta dùng hàm **mutate()** để tạo 2 cột mới trong biến **world**, và dùng hàm **cumsum()** để tính tổng tích luỹ cho 2 cột **new\_cases** và **new\_deaths**. Sau đó chọn lại các tháng cần phân tích ứng với 3 năm 2020, 2021, 2022

```
world <- world %>% mutate(
  cumulative_cases = cumsum(new_cases)) %>%
  mutate(
    cumulative_deaths = cumsum(new_deaths))
)
twenty <- world %>% filter(year(date) == 2020)
twenty1 <- world %>% filter(year(date) == 2021)
twenty2 <- world %>% filter(year(date) == 2022)
#2020
Jan2020 <- twenty %>% filter(month(date) == 1)
Feb2020 <- twenty %>% filter(month(date) == 2)
Apr2020 <- twenty %>% filter(month(date) == 4)
Oct2020 <- twenty %>% filter(month(date) == 10)
#2021
Jan2021 <- twenty1 %>% filter(month(date) == 1)
Feb2021 <- twenty1 %>% filter(month(date) == 2)
Apr2021 <- twenty1 %>% filter(month(date) == 4)
Oct2021 <- twenty1 %>% filter(month(date) == 10)
#2022
```



```
Jan2022 <- twenty2 %>% filter(month(date) == 1)
Feb2022 <- twenty2 %>% filter(month(date) == 2)
```

**Vẽ biểu đồ:** Trước khi vẽ biểu đồ, chúng ta tắt các ký hiệu khoa học để dễ cho quan sát:

```
options(scipen = 999) #tat ky hieu khoa hoc
```

1 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh cho từng tháng

Sử dụng hàm **ggplot**, kết hợp **geom\_line** và **geom\_point** để vẽ biểu đồ đường và điểm biểu diễn.

```
cases_0120 <- ggplot(data = Jan2020, aes(x = date, y = new_cases))+
  geom_line(color = "lightblue") +
  geom_point(size = 1, color = "#0871c2") +
  labs(x = "", y = "Cases", title = "New cases in January 2020") +
  theme_minimal()+
  theme(plot.title = element_text(size = 13, face = "italic"),
        axis.title.x = element_text(size = 10, face = "italic"),
        axis.title.y = element_text(size = 10, face = "italic"))
cases_0120
```

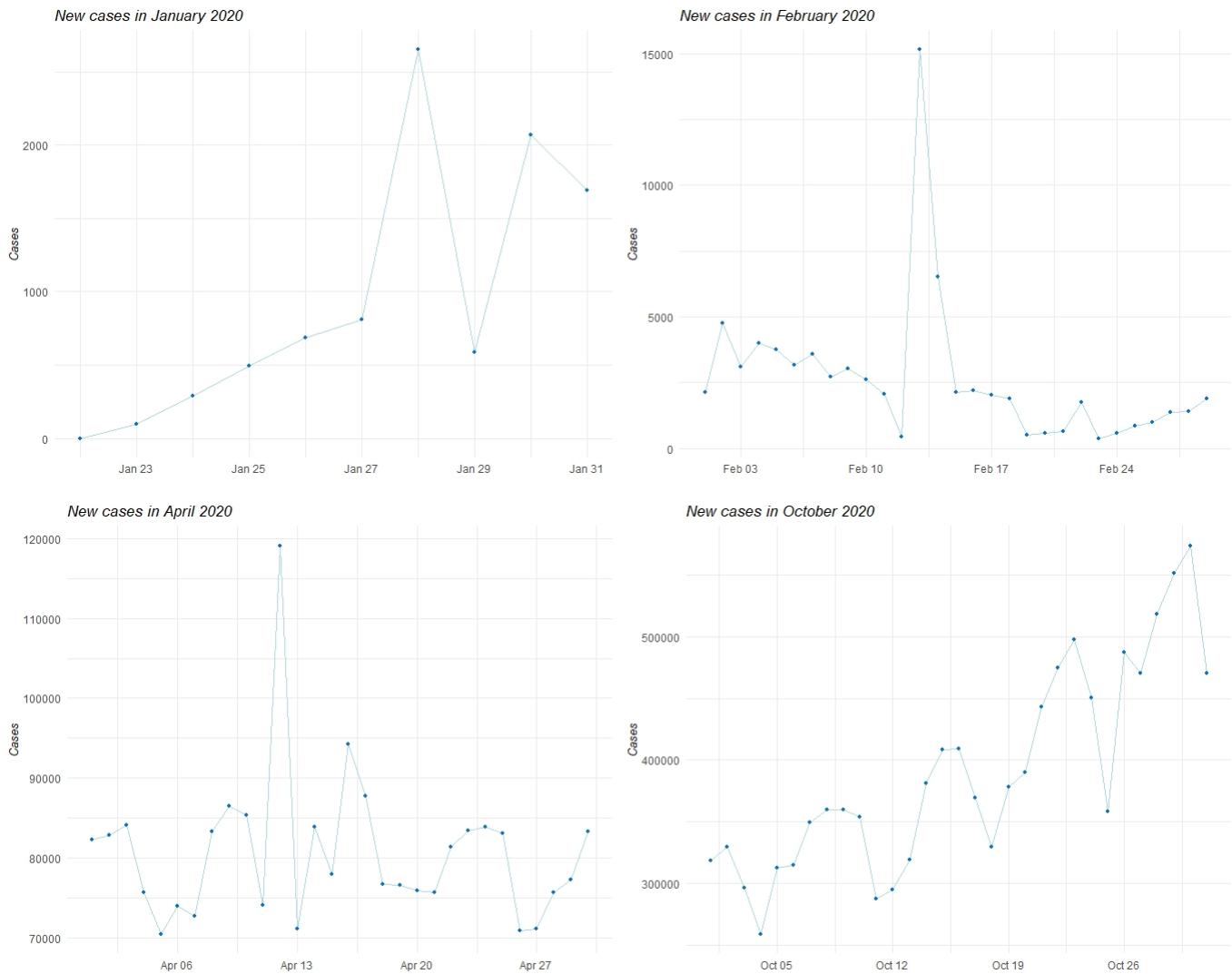
Ở đây, ta chọn biến **Jan2020** (01/2020) làm biến đại diện, làm tương tự cho các biến còn lại.

Ta cũng có thể dùng lệnh **cowplot::plot\_grid** để ghép các biểu đồ lại với nhau:

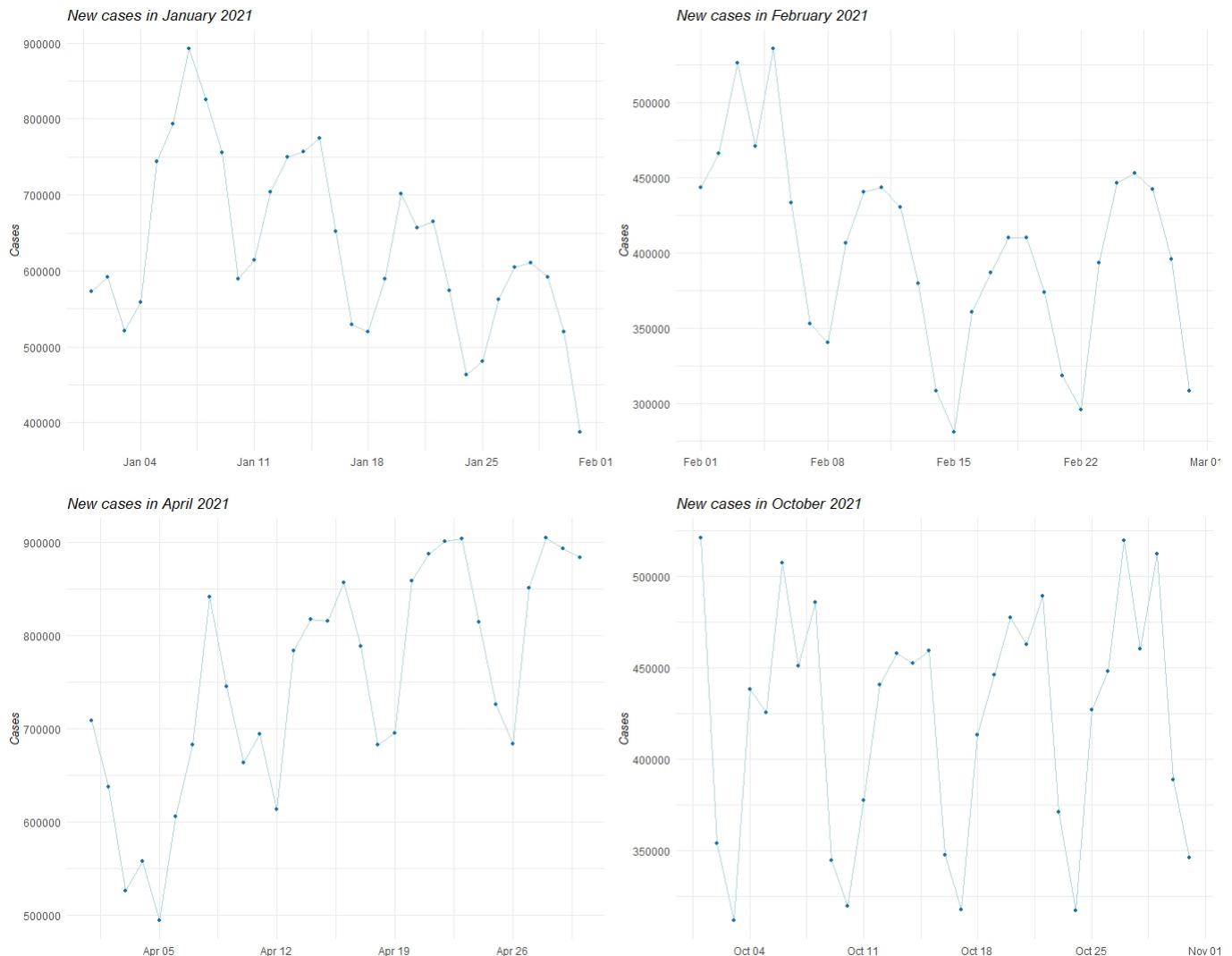
```
cases_plots2020 <- cowplot::plot_grid(cases_0120, cases_0220, cases_0420,
                                         cases_1020)
cases_plots2021 <- cowplot::plot_grid(cases_0121, cases_0221, cases_0421,
                                         cases_1021)
cases_plots2022 <- cowplot::plot_grid(cases_0122, cases_0222)
```

## Kết quả:

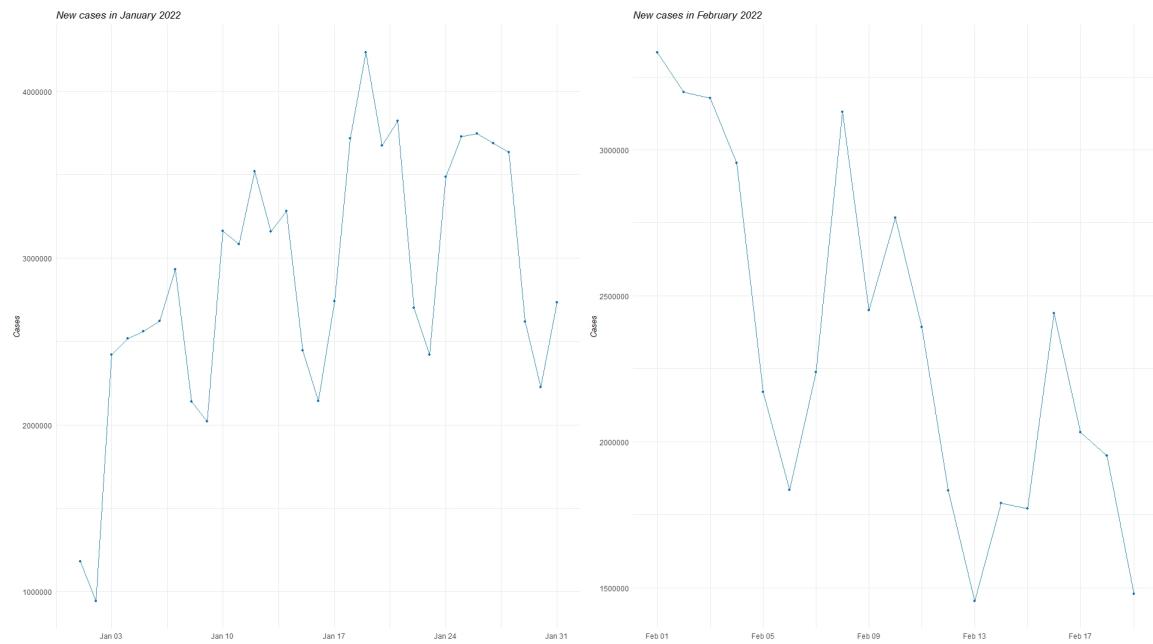
Năm 2020:



Năm 2021:



Năm 2022:



2 Biểu đồ thể hiện thu thập dữ liệu tử vong cho từng tháng

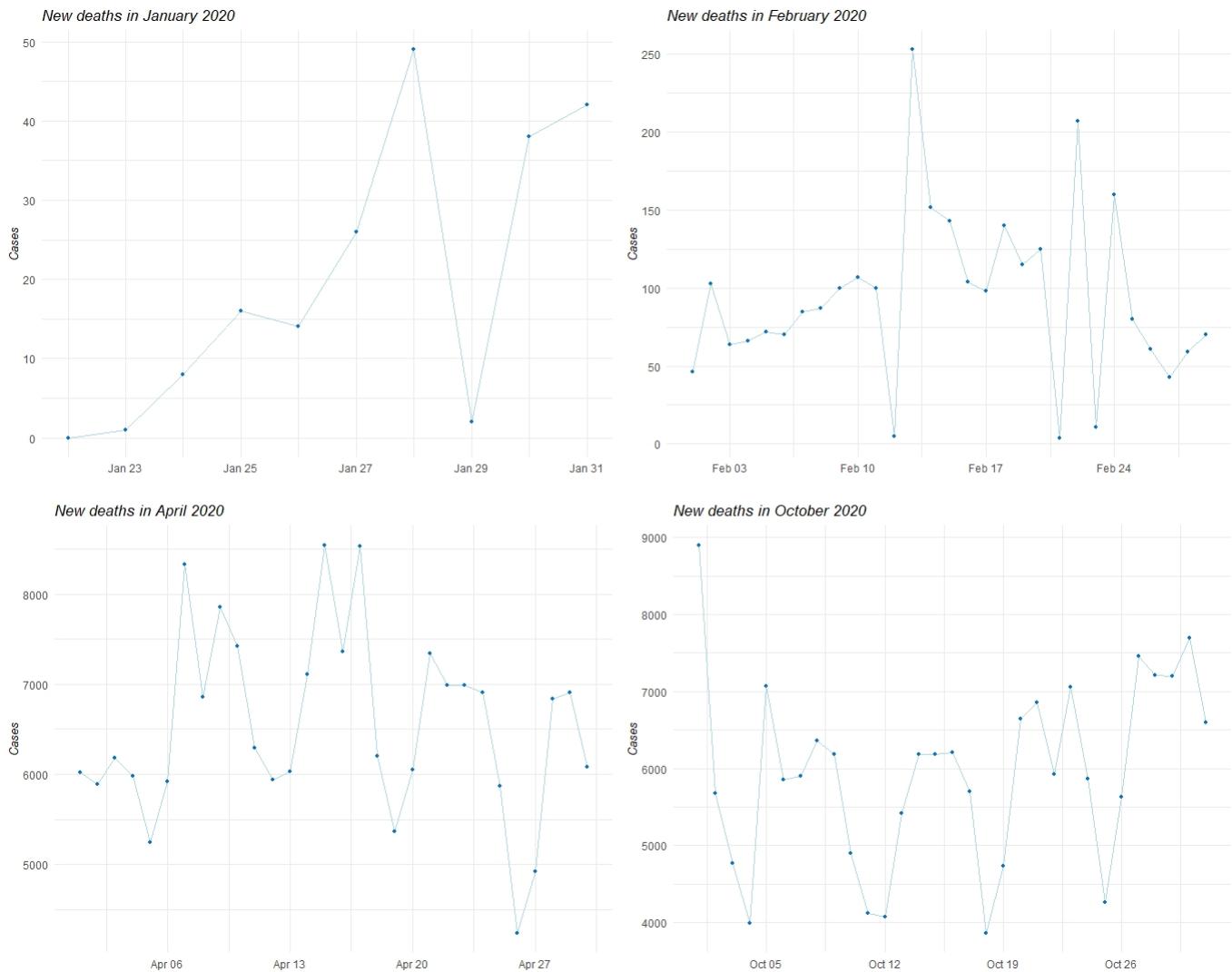
Sử dụng hàm `ggplot`, kết hợp `geom_line` và `geom_point` để vẽ biểu đồ đường và điểm biểu diễn.

```
####01.2020
dead_0120 <- ggplot(data = Jan2020, aes(x = date, y = new_deaths))+
  geom_line(color = "lightblue") +
  geom_point(size = 1, color = "#0871c2") +
  labs(x = "", y = "Cases", title = "New deaths in January 2020" ) +
  theme_minimal()+
  theme(plot.title = element_text(size = 13, face = "italic"),
        axis.title.x = element_text(size = 10, face = "italic"),
        axis.title.y = element_text(size = 10, face = "italic"))
dead_0120 #print
```

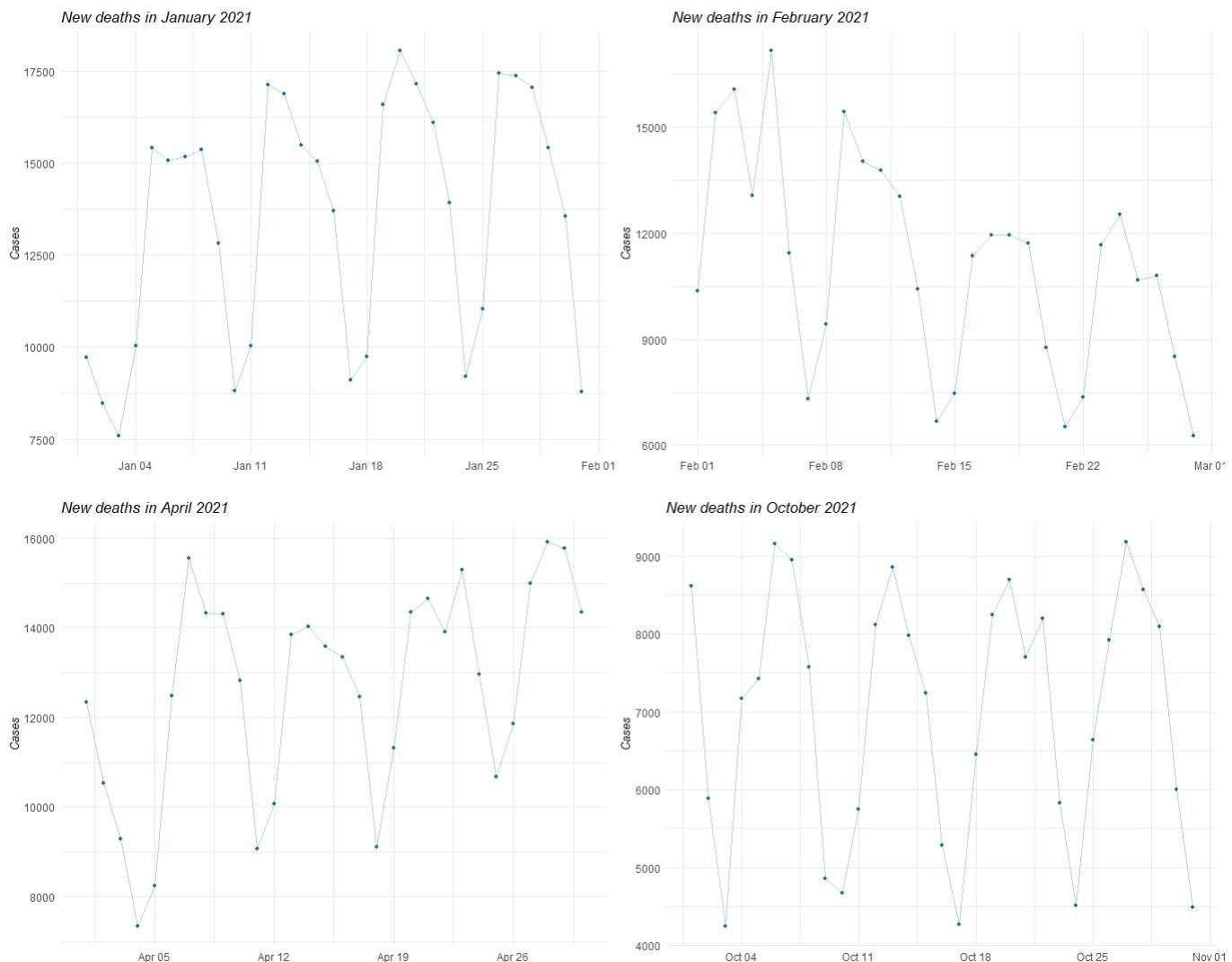
Ở đây, ta chọn biến **Jan2020** (01/2020) làm biến đại diện, làm tương tự cho các biến còn lại.

### Kết quả:

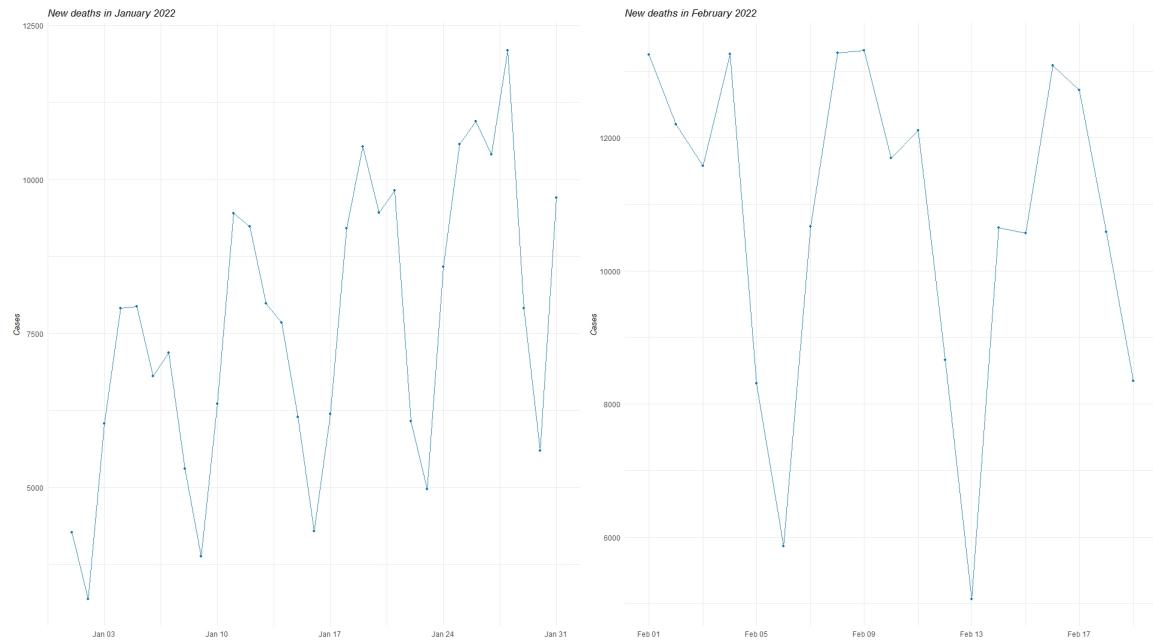
Năm 2020:



Năm 2021:



Năm 2022:



3 Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong cho từng tháng

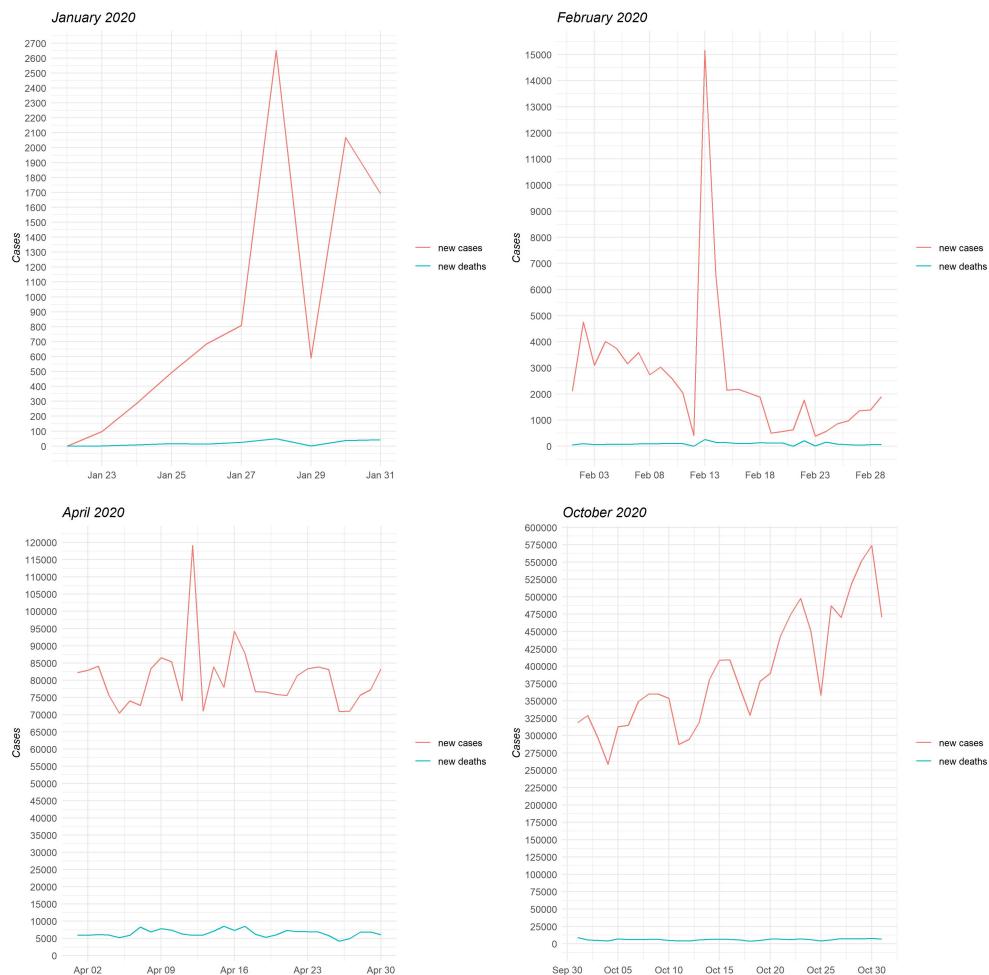
Sử dụng hàm `ggplot`, kết hợp 2 hàm `geom_line` biểu thị cho số lượng nhiễm bệnh và tử vong để vẽ biểu đồ.

```
#01/2020
CnD_0120 <- ggplot(Jan2020, aes(date)) +
  geom_line(aes(y = new_cases, colour = "new cases")) +
  geom_line(aes(y = new_deaths, colour = "new deaths")) +
  labs(x = "", y = "Cases", title = "January 2020", color = "") +
  theme_minimal() +
  theme(plot.title = element_text(size = 13, face = "italic"),
        axis.title.x = element_text(size = 10, face = "italic"),
        axis.title.y = element_text(size = 10, face = "italic")) +
  scale_y_continuous(
    breaks = seq(
      from = 0,
      to = 3000,
      by = 100
    )
  )
CnD_0120 #print
```

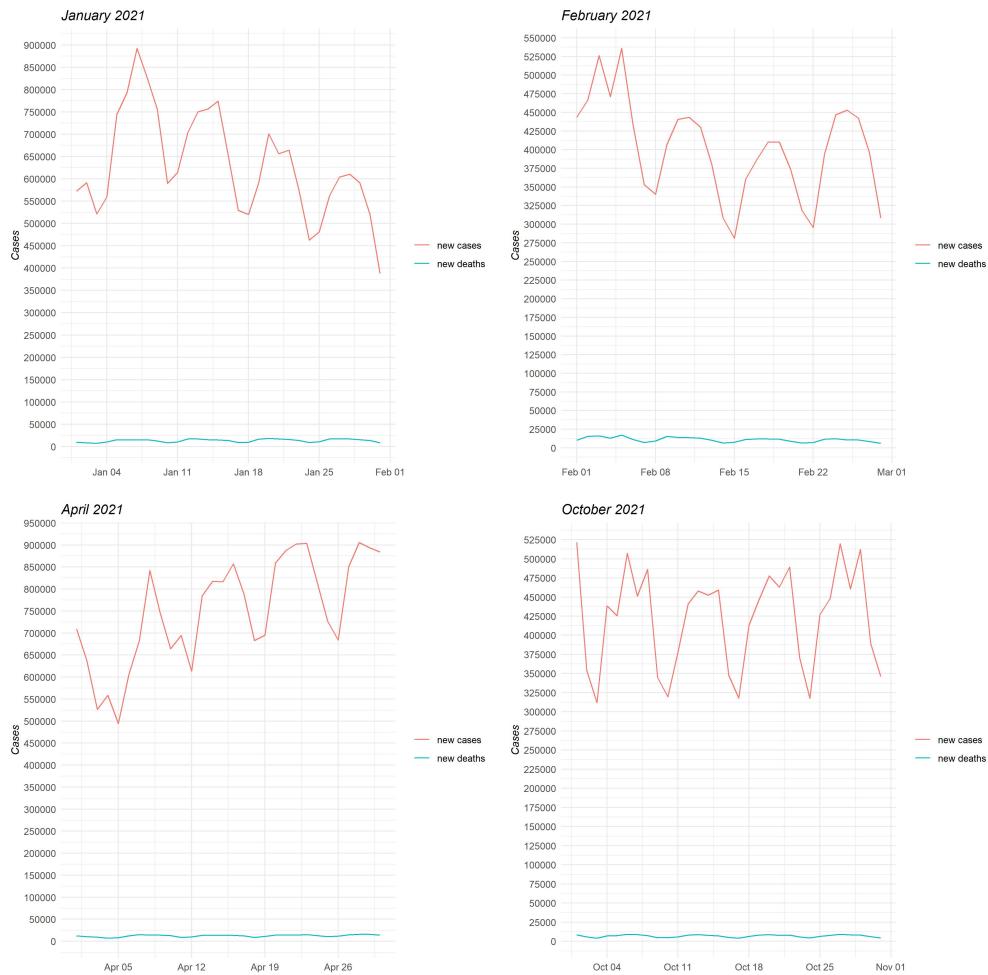
Ở đây, ta chọn biến **Jan2020** (01/2020) làm biến đại diện, làm tương tự cho các biến còn lại.

### Kết quả:

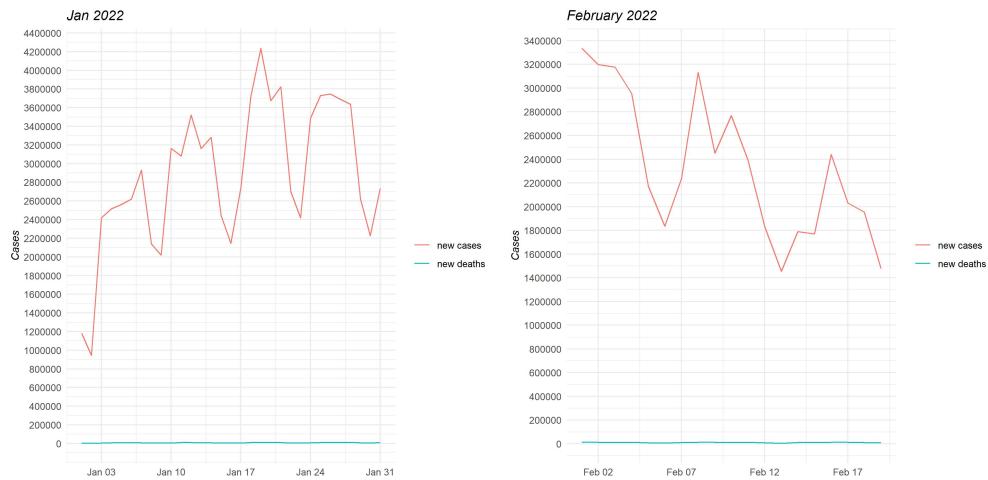
Năm 2020:



Năm 2021:



Năm 2022:



4 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh gồm 2 tháng cuối của năm

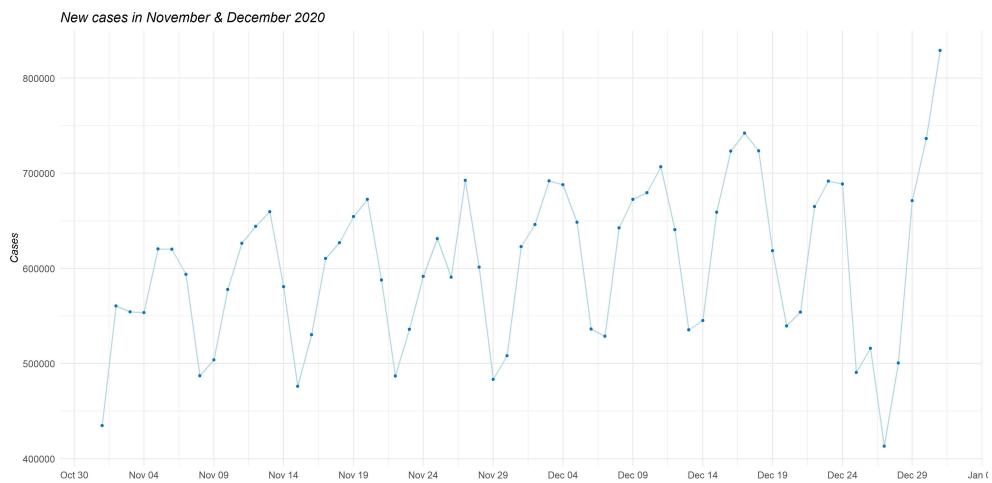
Sử dụng hàm `ggplot`, kết hợp `geom_line` và `geom_point` để vẽ biểu đồ đường và điểm biểu diễn.

```
####2020
cases_NovDec2020 <- ggplot(data = NovDec2020, aes(x = date, y = new_cases))+
  geom_line(color = "lightblue") +
  geom_point(size = 1, color = "#0871c2") +
  labs(x = "", y = "Cases", title = "New cases in November & December 2020" ) +
  theme_minimal() +
  theme(plot.title = element_text(size = 13, face = "italic"),
        axis.title.x = element_text(size = 10, face = "italic"),
        axis.title.y = element_text(size = 10, face = "italic")) +
  scale_x_date(
    date_breaks = "5 days",
    date_labels = "%b %d"
  )
cases_NovDec2020 #print
```

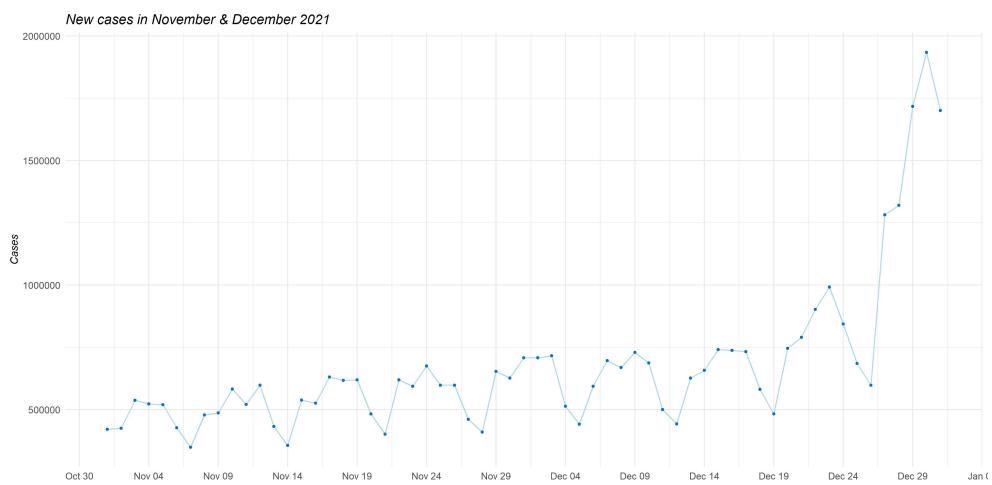
Ở đây, ta chọn biến **NovDec2020** làm biến đại diện - tương ứng với 2 tháng cuối của năm 2020, làm tương tự cho các biến còn lại.

### Kết quả:

Năm 2020:



Năm 2021:



5 Biểu đồ thể hiện thu thập dữ liệu tử vong gồm 2 tháng cuối của năm

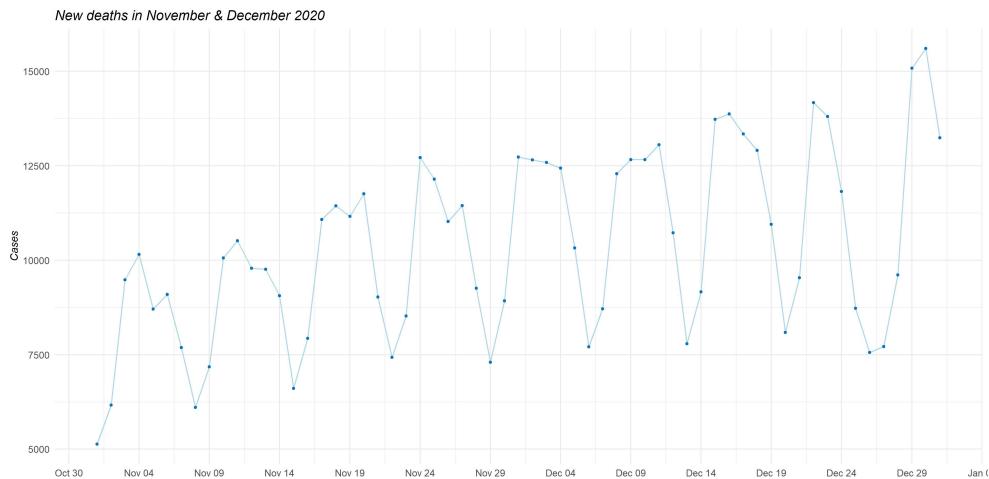
Sử dụng hàm `ggplot`, kết hợp `geom_line` và `geom_point` để vẽ biểu đồ đường và điểm biểu diễn. Lưu biểu đồ vào biến `dead_NovDec2020` để tiện lưu trữ và sử dụng.

```
dead_NovDec2020 <- ggplot(data = NovDec2020, aes(x = date, y = new_deaths))+
  geom_line(color = "lightblue") +
  geom_point(size = 1, color = "#0871c2") +
  labs(x = "", y = "Cases", title = "New deaths in November & December 2020") +
  theme_minimal() +
  theme(plot.title = element_text(size = 13, face = "italic"),
        axis.title.x = element_text(size = 10, face = "italic"),
        axis.title.y = element_text(size = 10, face = "italic")) +
  scale_x_date(
    date_breaks = "5 days",
    date_labels = "%b %d"
  )
dead_NovDec2020 #print
```

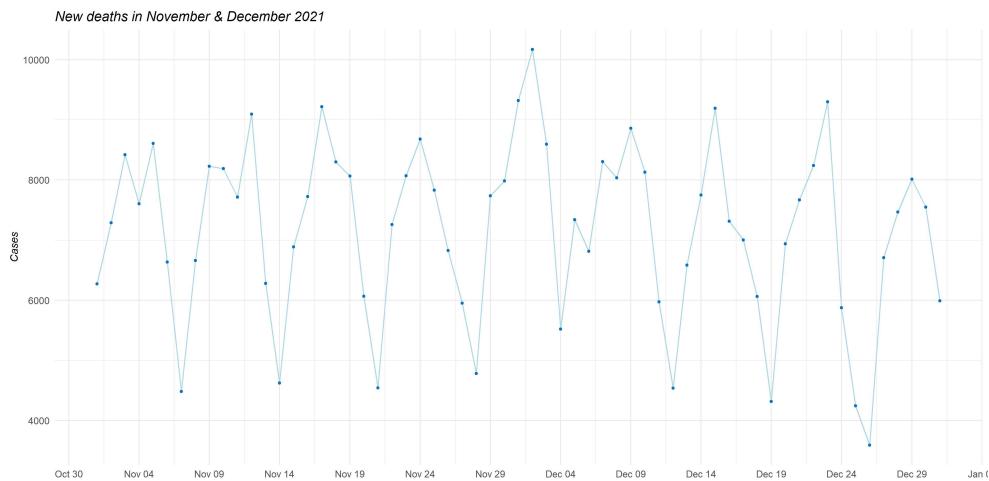
Ở đây, ta chọn biến **NovDec2020** làm biến đại diện - tương ứng với 2 tháng cuối của năm 2020, làm tương tự cho các biến còn lại.

### Kết quả:

Năm 2020:



Năm 2021:



6 Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong gồm 2 tháng cuối của năm  
Sử dụng hàm `ggplot`, kết hợp 2 hàm `geom_line` biểu thị cho số lượng nhiễm bệnh và tử vong để vẽ biểu đồ. Lưu biểu đồ vào biến `CnD_NovDec2020` (cases and deaths), để tiện lưu trữ và sử dụng.

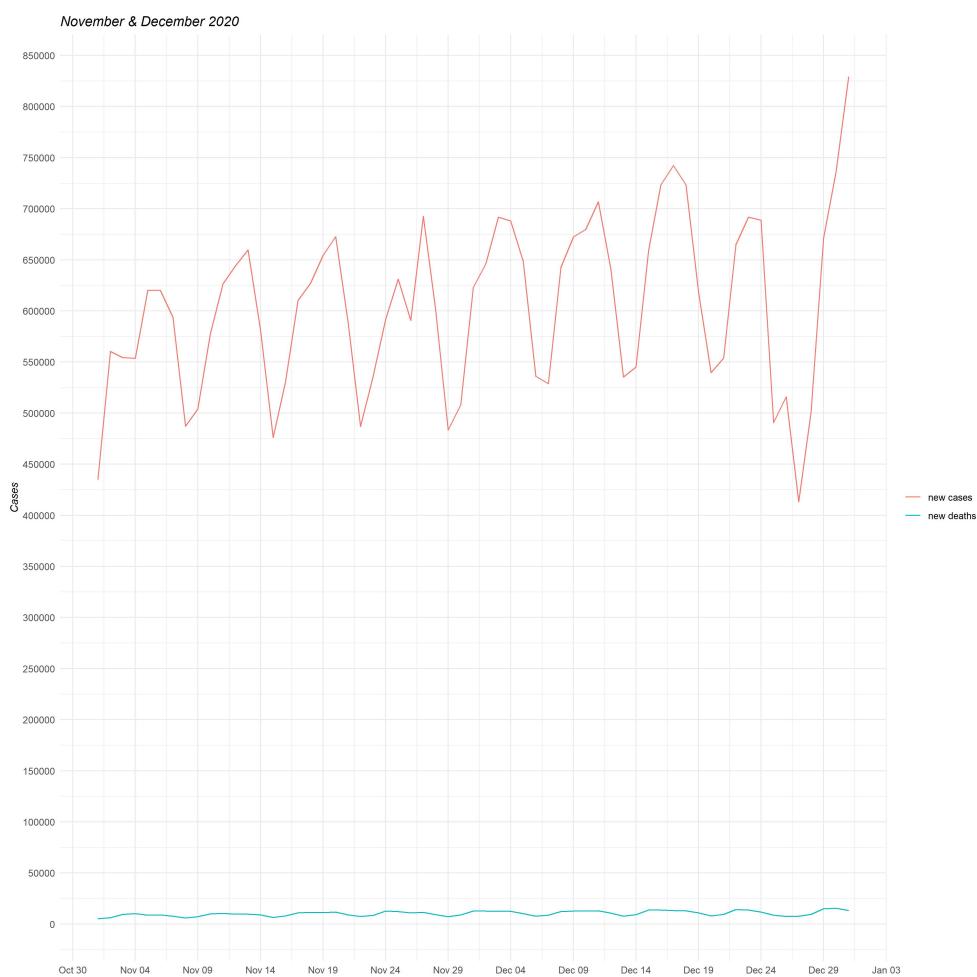
```
CnD_NovDec2020 <- ggplot(NovDec2020, aes(date)) +
  geom_line(aes(y = new_cases, colour = "new cases")) +
  geom_line(aes(y = new_deaths, colour = "new deaths")) +
  labs(x = "", y = "Cases", title = "November & December 2020", color = "") +
  theme_minimal() +
  theme(plot.title = element_text(size = 13, face = "italic"),
        axis.title.x = element_text(size = 10, face = "italic"),
        axis.title.y = element_text(size = 10, face = "italic")) +
  scale_y_continuous(
    breaks = seq(
```

```
        from = 0,
        to = 1000000,
        by = 50000
    )
) +
scale_x_date(
    date_breaks = "5 days",
    date_labels = "%b %d"
)
CnD_NovDec2020
```

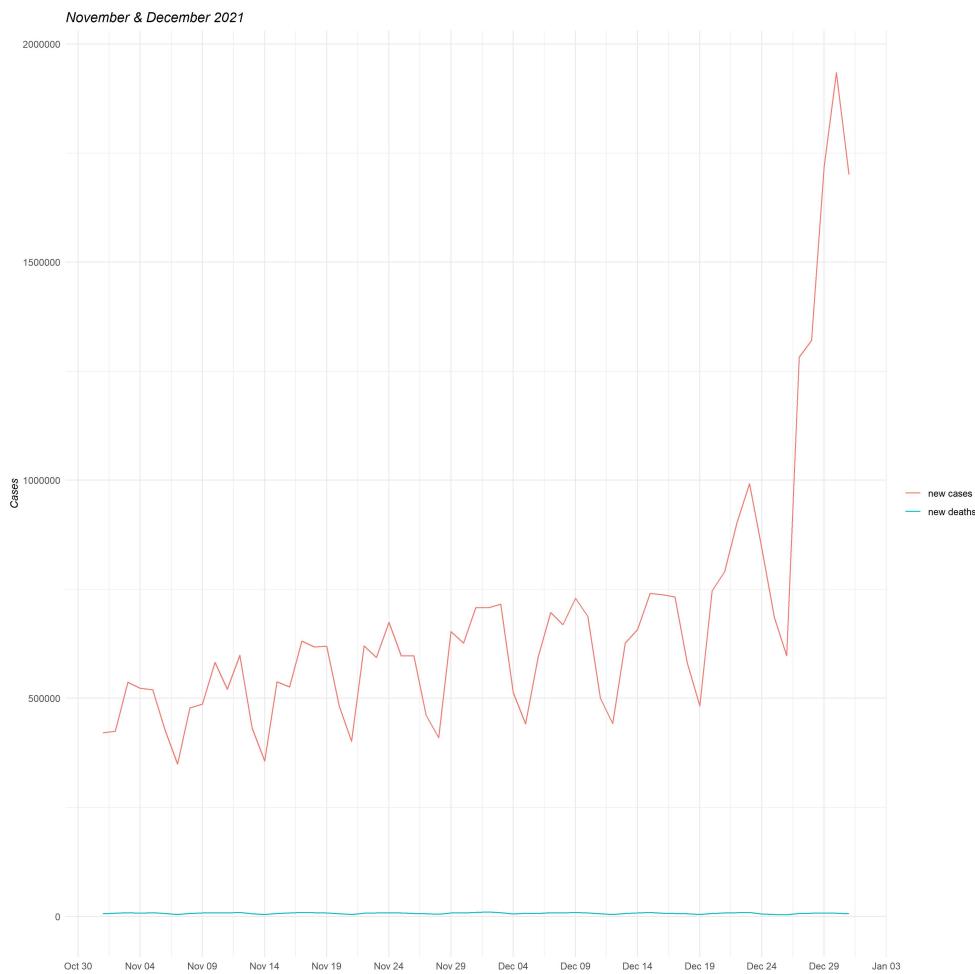
Ở đây, ta chọn biến **NovDec2020** làm biến đại diện - tương ứng với 2 tháng cuối của năm 2020, làm tương tự cho các biến còn lại.

### Kết quả:

Năm 2020:



Năm 2021:

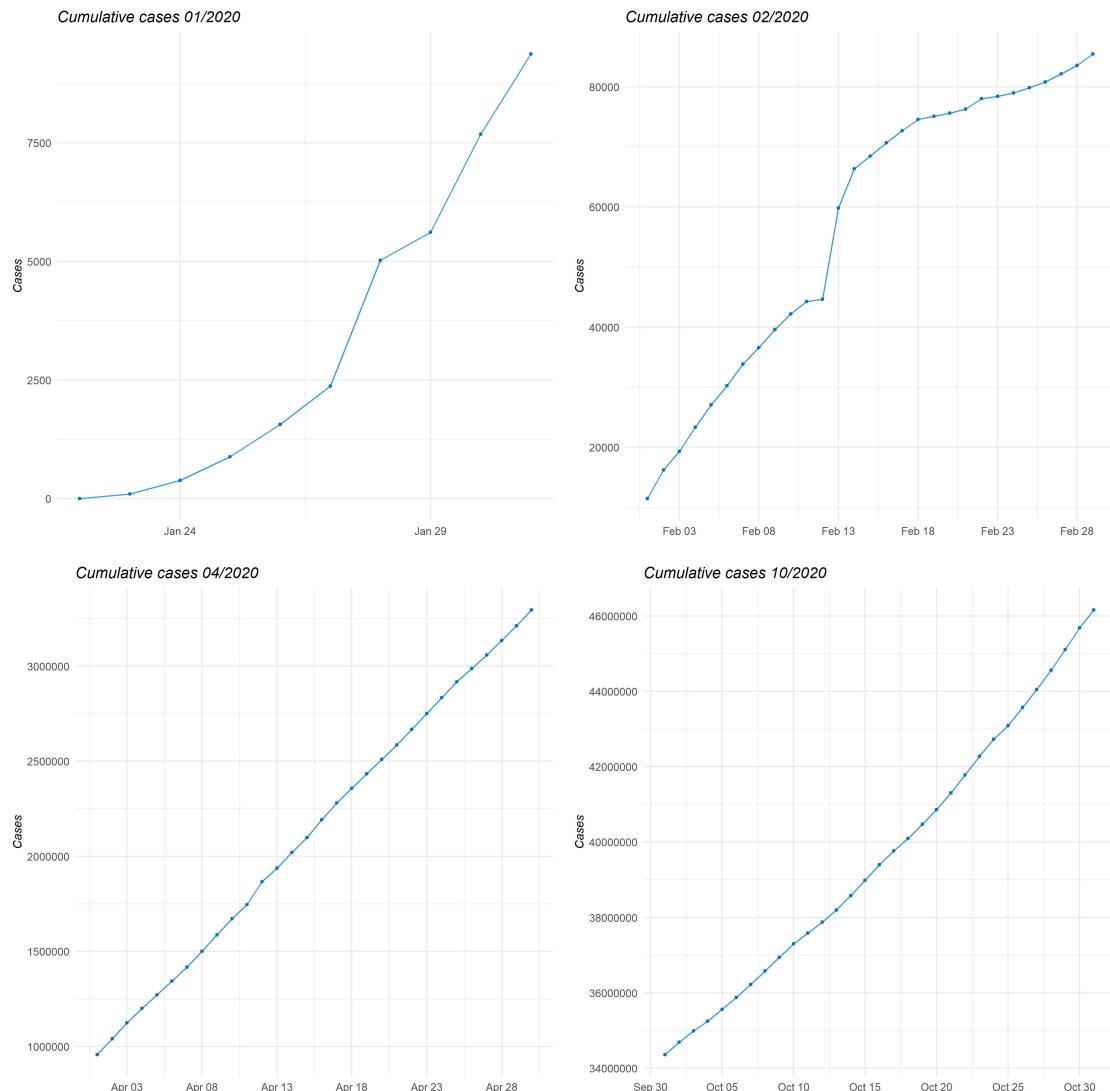


7 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy cho từng tháng

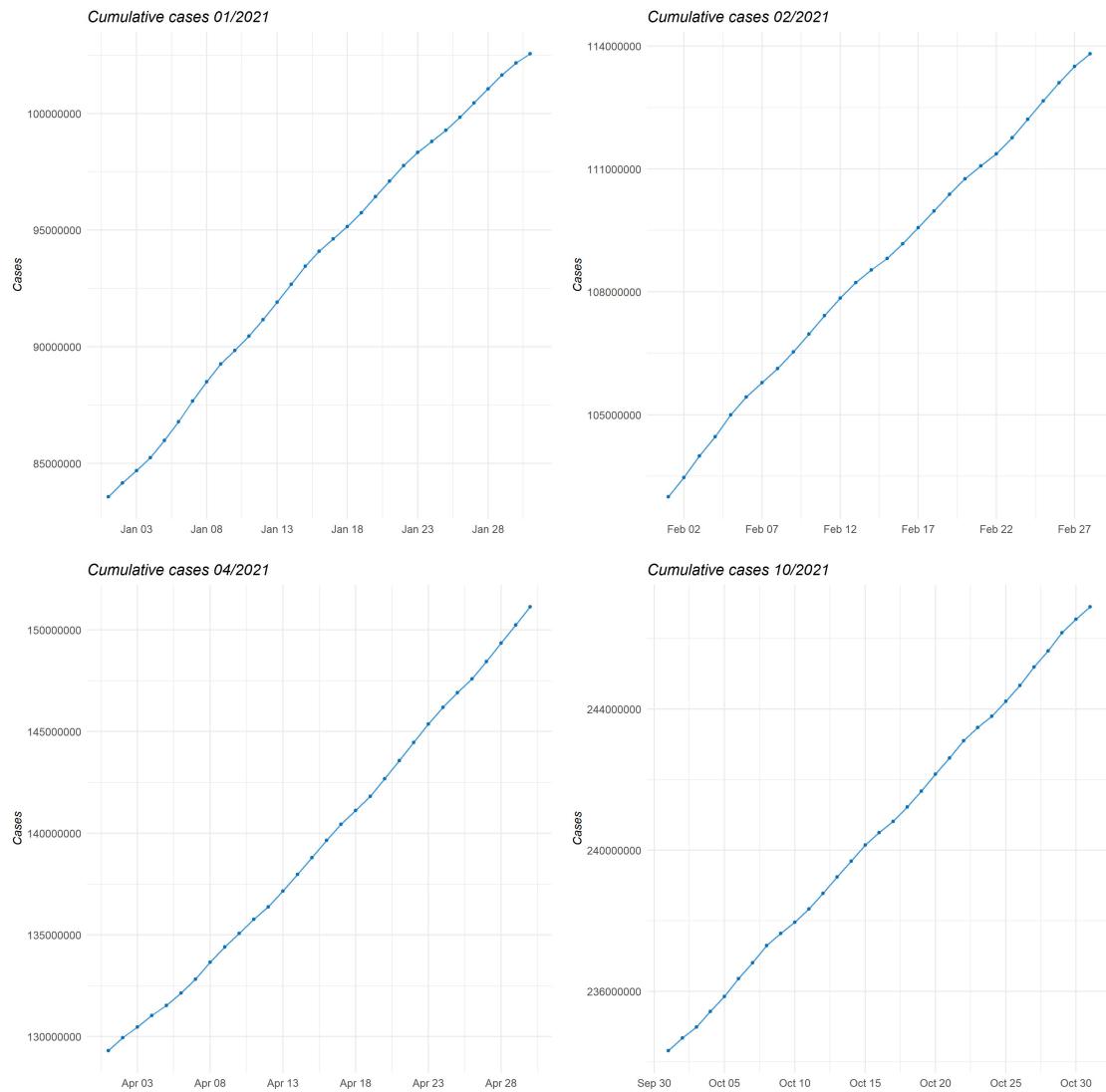
Code:

```
cumsumCases_0120 <- ggplot(data = Jan2020, aes(x = date, y = cumulative_cases))  
  +  
  geom_line(color = "#3897e0") +  
  geom_point(size = 1, color = "#0871c2") +  
  labs(x = "", y = "Cases", title = "Cumulative cases 01/2020") +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 13, face = "italic"),  
        axis.title.x = element_text(size = 10, face = "italic"),  
        axis.title.y = element_text(size = 10, face = "italic")) +  
  scale_x_date(  
    date_breaks = "5 days",  
    date_labels = "%b %d"  
)  
cumsumCases_0120 #print
```

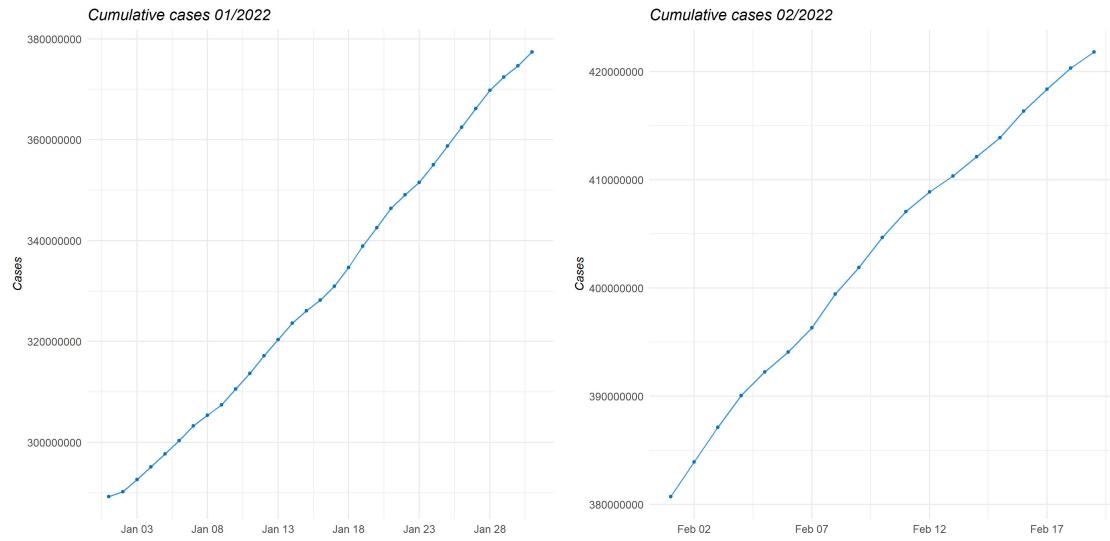
Kết quả: Năm 2020:



Năm 2021:



Năm 2022:



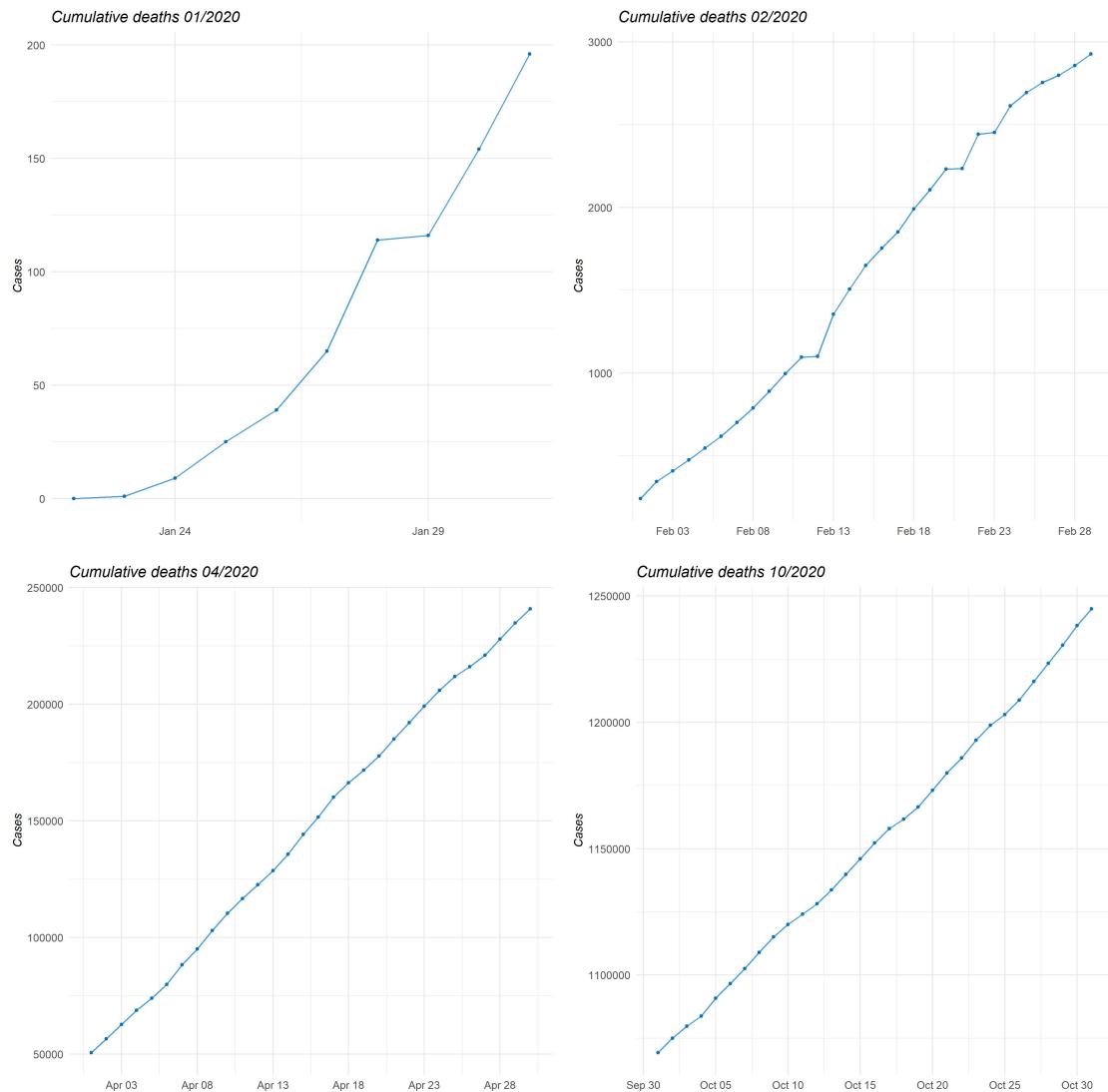
8 Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy cho từng tháng

Code:

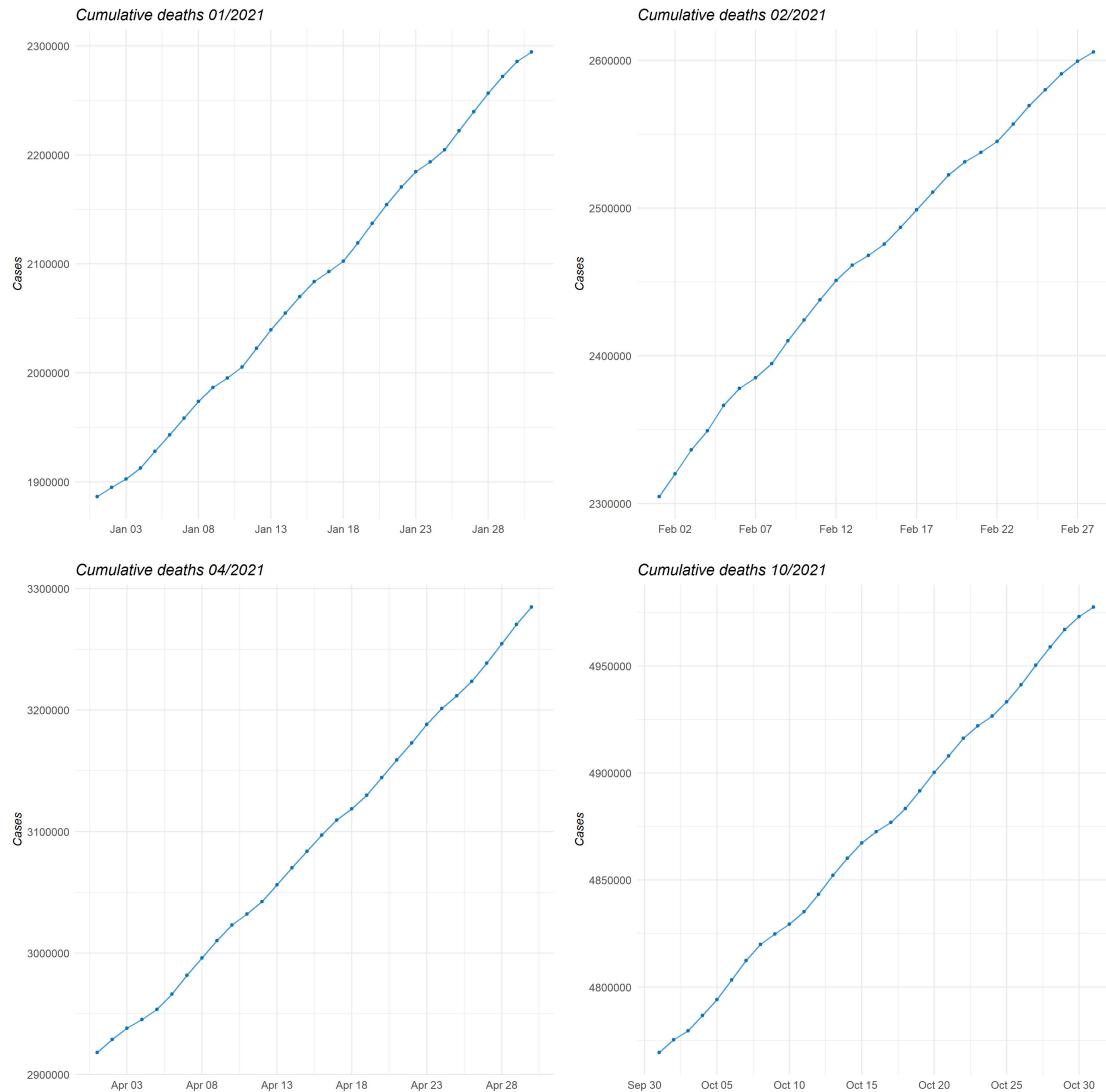
```
cumsumDeaths_0120 <- ggplot(data = Jan2020, aes(x = date, y = cumulative_deaths)) +  
  geom_line(color = "#3897e0") +  
  geom_point(size = 1, color = "#0871c2") +  
  labs(x = "", y = "Cases", title = "Cumulative deaths 01/2020") +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 13, face = "italic"),  
        axis.title.x = element_text(size = 10, face = "italic"),  
        axis.title.y = element_text(size = 10, face = "italic")) +  
  scale_x_date(  
    date_breaks = "5 days",  
    date_labels = "%b %d"  
)  
cumsumDeaths_0120
```

Kết quả:

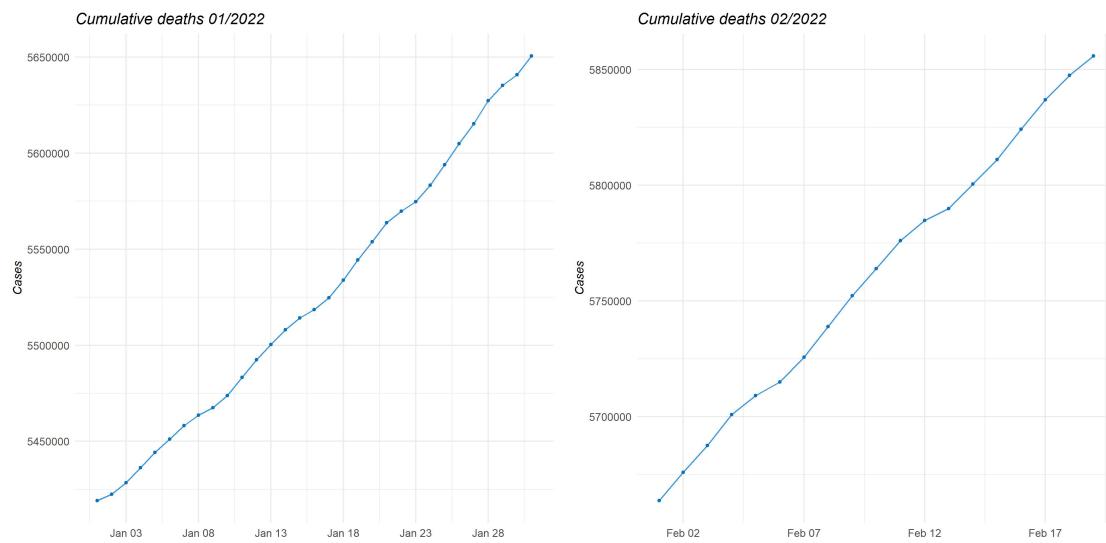
Năm 2020:



Năm 2021:



Năm 2022:





## vi Nhóm câu hỏi liên quan đến trực quan dữ liệu theo trung bình 7 ngày gần nhất

- Với mỗi quốc gia mà thuộc về nhóm, trên từng năm hãy vẽ biểu đồ thể hiện trục Ox là thời gian, trục Oy là nhiễm bệnh/tử vong. Hãy dùng 4 ký số của mã đề để vẽ 4 tháng tương ứng theo ký số đó. Nếu ký số là 0 thì lấy tháng là 10.
- Dùng trung bình của các ca nhiễm bệnh và tử vong được báo cáo trong 7 ngày gần nhất để loại trừ một số báo cáo không thường xuyên và đưa chúng ta đến gần hơn với con số hàng ngày.

\*Các bước đọc dữ liệu:

- Đọc dữ liệu vào dataframe
  - + readcsv() đọc dữ liệu file covidData.csv vào data
  - + lọc dataframe với
- mutate() biến đổi date thành dữ liệu ngày tháng bằng as.Date,
- groupby() theo ngày,
- filter theo mã iso code (IDN,JPN,VNM),
- summarise tạo cột newcases newdeaths đưa vào dataframe tương ứng

```
data <- read_csv(file='...covidData.csv')
#country
data %>%
  mutate(date = as.Date(date,"%m/%d/%Y")) %>%
  group_by(date) %>%
  filter(iso_code=="IDN") %>%
  summarise(new_cases,new_deaths)-> IDN
data %>%
  mutate(date = as.Date(date,"%m/%d/%Y")) %>%
  group_by(date) %>%
  filter(iso_code=="JPN") %>%
  summarise(new_cases,new_deaths)-> JPN
data %>%
  mutate(date = as.Date(date,"%m/%d/%Y")) %>%
  group_by(date) %>%
  filter(iso_code=="VNM") %>%
  summarise(new_cases,new_deaths)-> VNM
```

- Xử lý dữ liệu:

- + thành lập hàm để kiểm tra các dữ liệu NA, 0 và dữ liệu lỗi (âm) ở hai cột ca nhiễm và tử vong và thay thế bằng trung bình 7 ngày gần nhất

```
#7 days average replacement
fix_cases <- function(df) {
  df$new_cases =
    ifelse(
      (is.na(df$new_cases)|df$new_cases<=0),
      floor(slider::slide_dbl(df$new_cases, mean, na.rm = TRUE, .before = 7, .
        after = -1)),
      df$new_cases
    )
}
fix_deaths <- function(df) {
  df$new_deaths =
    ifelse(
      (is.na(df$new_deaths)|df$new_deaths<=0),
      floor(slider::slide_dbl(df$new_deaths, mean, na.rm = TRUE, .before = 7, .
        after = -1)),
      df$new_deaths
    )
}
IDN$new_cases <- fix_cases(IDN)
JPN$new_cases <- fix_cases(JPN)
VNM$new_cases <- fix_cases(VNM)
```



```
IDN$new_deaths <- fix_deaths(IDN)
JPN$new_deaths <- fix_deaths(JPN)
VNM$new_deaths <- fix_deaths(VNM)
```

+ kiểm tra lại các dữ liệu NA (do hàm sử dụng dữ liệu trung bình 7 ngày gần nhất, các dữ liệu NA nằm đầu dataframe không thể tính trung bình)

```
#check NA
IDN$new_cases [is.na(IDN$new_cases)] = 0
JPN$new_cases [is.na(JPN$new_cases)] = 0
VNM$new_cases [is.na(VNM$new_cases)] = 0
IDN$new_deaths [is.na(IDN$new_deaths)] = 0
JPN$new_deaths [is.na(JPN$new_deaths)] = 0
VNM$new_deaths [is.na(VNM$new_deaths)] = 0
```

+ Lọc theo tháng bằng subset (Mã 1204 -> T1,T2,T4,T10 + 2 tháng cuối năm) gán vào dataframe năm tương ứng

<do dữ liệu từ đầu 2020 đến đầu năm 2022, chỉ lấy cuối năm 2020,2021>

```
#extract data
IDN %>% subset((date>="2020/01/01"&date<="2020/01/31")
                  |(date>="2020/02/01"&date<="2020/02/29")
                  |(date>="2020/04/01"&date<="2020/04/30")
                  |(date>="2020/10/01"&date<="2020/10/31"))
) -> IDN2020
IDN %>% subset((date>="2021/01/01"&date<="2021/01/31")
                  |(date>="2021/02/01"&date<="2021/02/28")
                  |(date>="2021/04/01"&date<="2021/04/30")
                  |(date>="2021/10/01"&date<="2021/10/31"))
) -> IDN2021
IDN %>% subset((date>="2022/01/01"&date<="2022/01/31")
                  |(date>="2022/02/01"&date<="2022/02/28")
                  |(date>="2022/04/01"&date<="2022/04/30")
                  |(date>="2022/10/01"&date<="2022/10/31"))
) -> IDN2022
IDN %>% subset((date>="2020/11/01"&date<="2020/12/31"))
) -> IDN2020end
IDN %>% subset((date>="2021/11/01"&date<="2021/12/31"))
) -> IDN2021end
```

```
#scientific notation
options(scipen=999)
#getmonth
getm <- function(df,m) {
  df %>% subset(format(date,"%m")==m)
}
```

- options(scipen=999) tắt kí hiệu khoa học (100000 -> 1e5) để dễ quan sát  
- hàm getm trả về dataframe là tháng m của df

\*Quan sát dữ liệu:

- + Indonesia từ 03/02/2020 đến 19/02/2022
  - + Japan từ 22/01/2020 đến 19/02/2022
  - + Vietnam từ 23/01/2020 đến 19/02/2022
- > Có thể lược bỏ một số tháng yêu cầu (không dữ liệu)



## 1 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh cho từng tháng

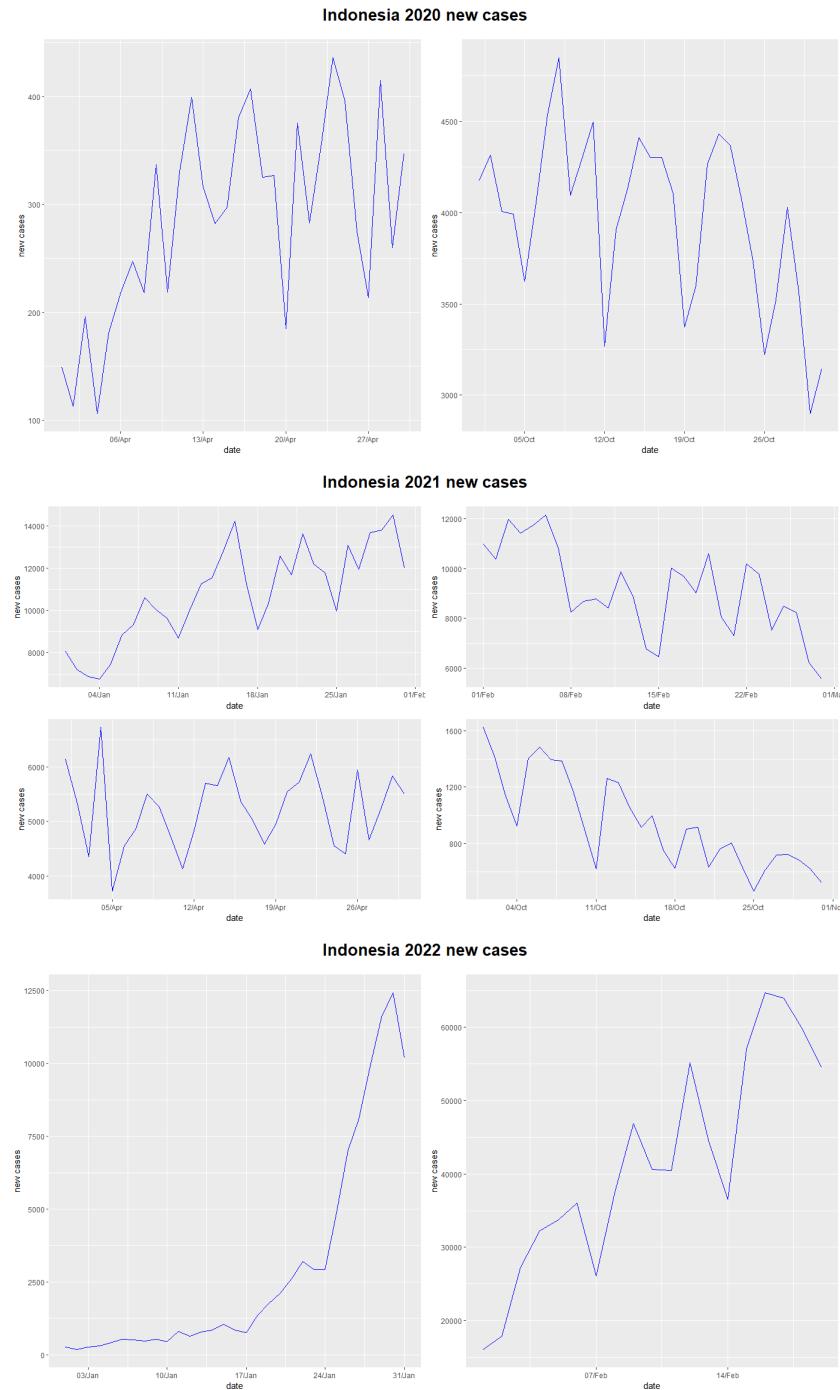
```
#1
#function plot cases
plotcases <- function(df,m){
  ggplot() +
  geom_line(getm(df,m),mapping=aes(x= date , y= new_cases),color="blue1") +
  labs(x="date",y="new cases") +
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
}
```

- Hàm plotcases nhận df và m trả về biểu đồ tháng m của dataframe df theo ca nhiễm

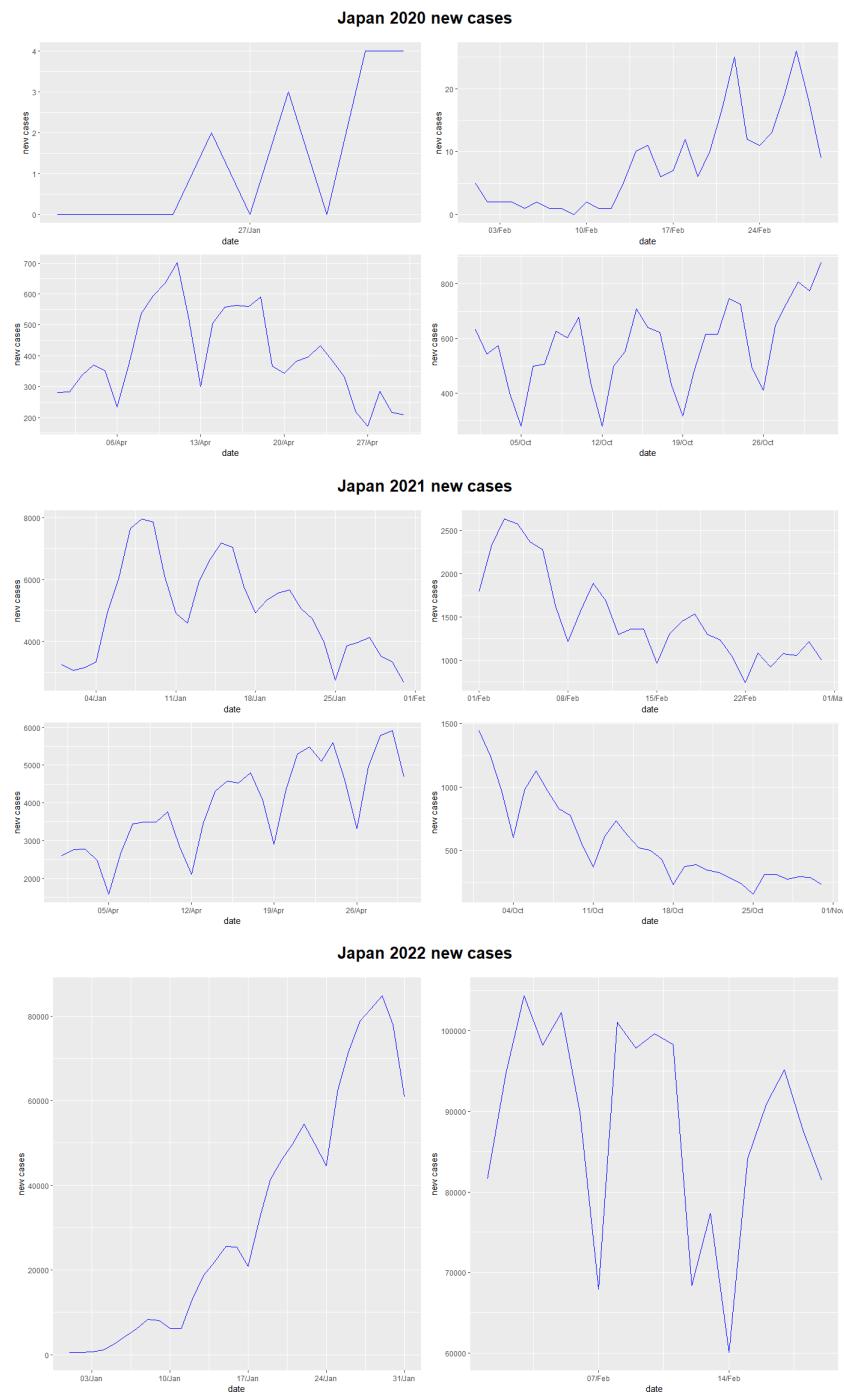
```
#IDN
plotcases(IDN2020,"04") -> t4
plotcases(IDN2020,"10") -> t10
plots <- plot_grid(
  t4 + theme(legend.position="none"),
  t10 + theme(legend.position="none"),
  axis = "tblr",
  align = "hv",
  nrow = 1,
  ncol = 2,
  rel_widths= c(1,1),
  rel_heights = c(1,1),
  scale = 1
)
title <- ggdraw() +
  draw_label("Indonesia 2020 new cases", size="20", fontface="bold")
plot_grid(title, plots, ncol=1, rel_heights=c(0.1, 1))
```

- plotgrid() cho phép gộp các biểu đồ. Tạo thêm title cho biểu đồ rồi dùng plotgrid gán vào.

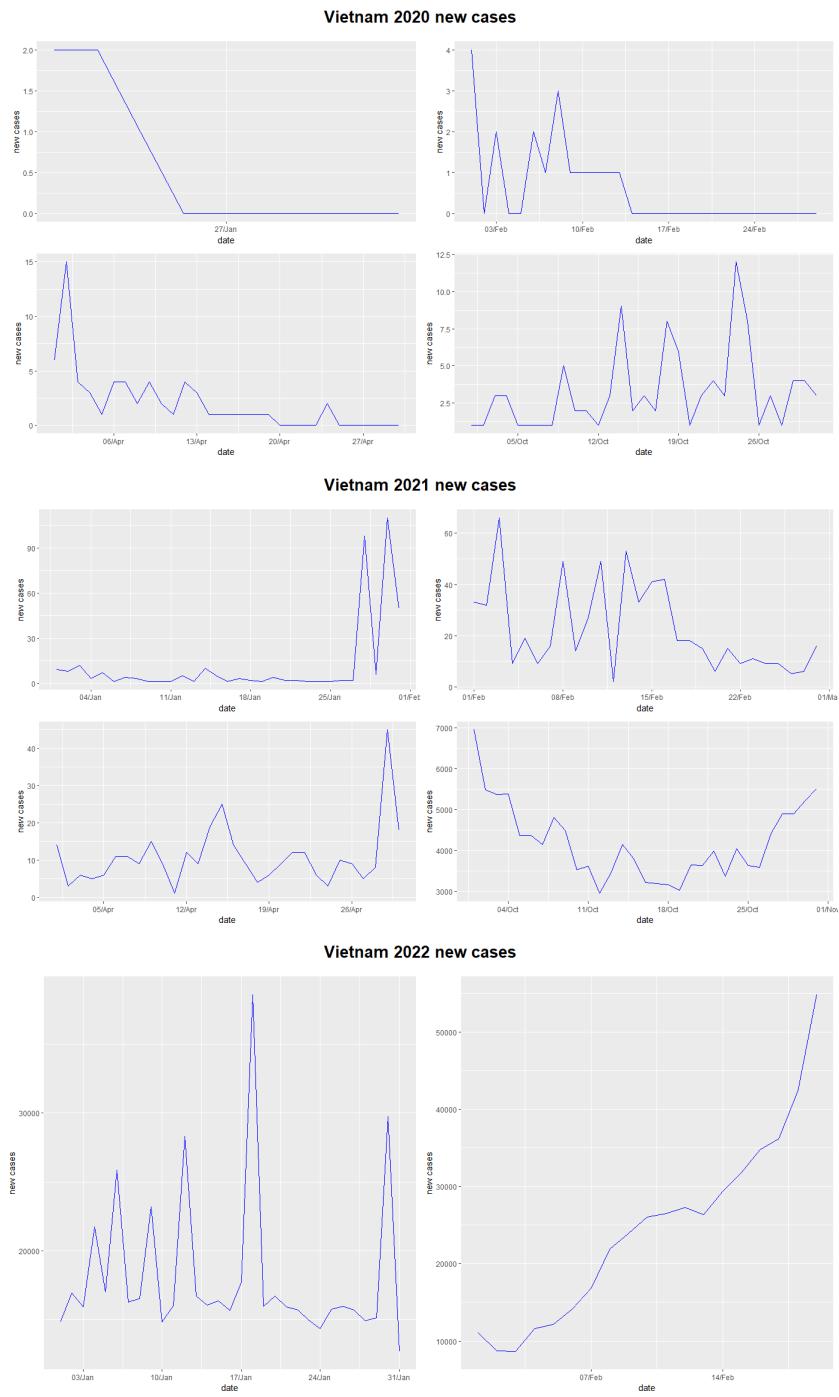
- Kết quả câu 1:
- + Indonesia



+ Japan



+ Vietnam





2 Biểu đồ thể hiện thu thập dữ liệu tử vong cho từng tháng

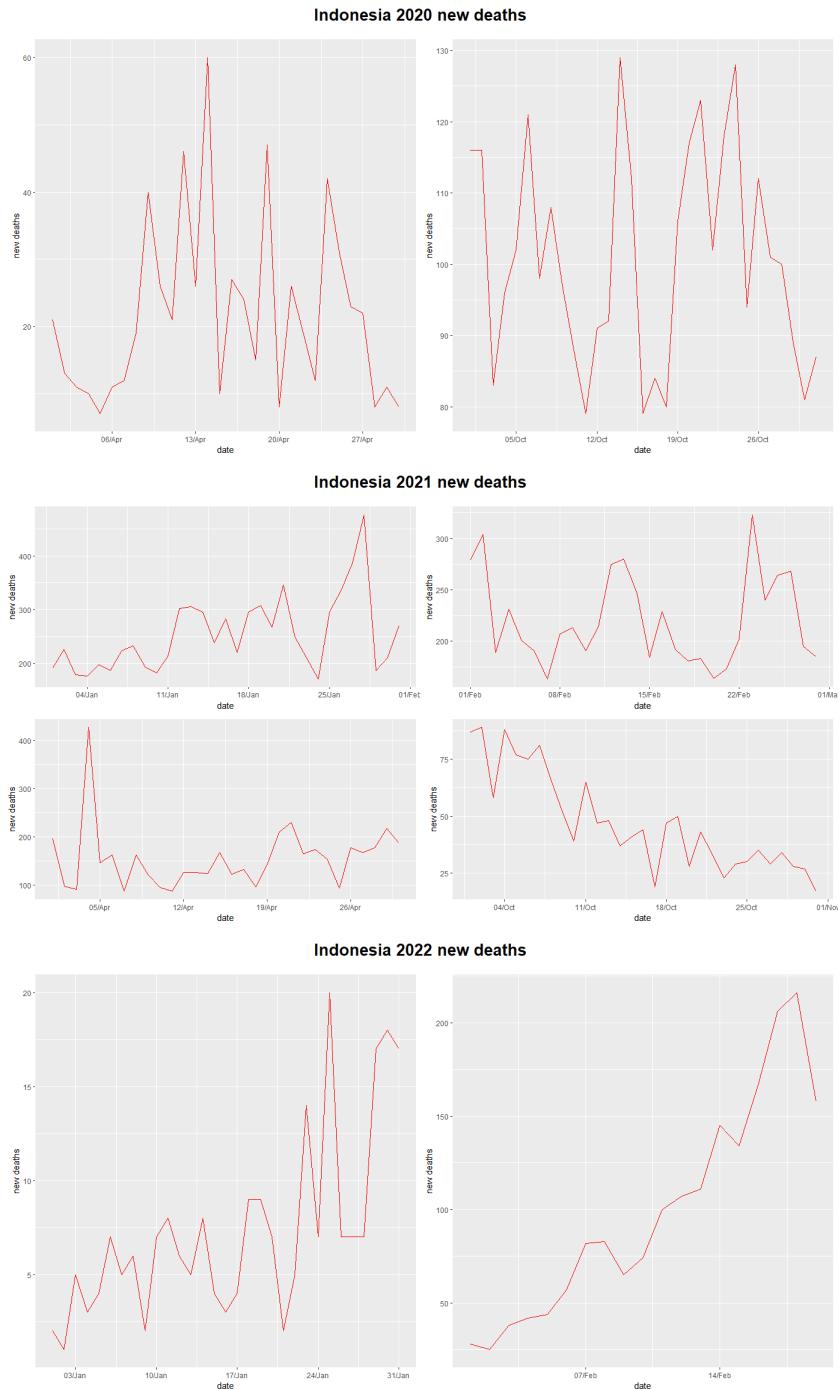
```
#2
function plot_deaths
plotdeaths <- function(df,m){
  ggplot() +
  geom_line(getm(df,m),mapping=aes(x= date , y= new_deaths),color="red2") +
  labs(x="date",y="new deaths") +
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
}
```

- Tương tự hàm plot cases, trả về plot theo tháng của dataframe theo ca tử vong

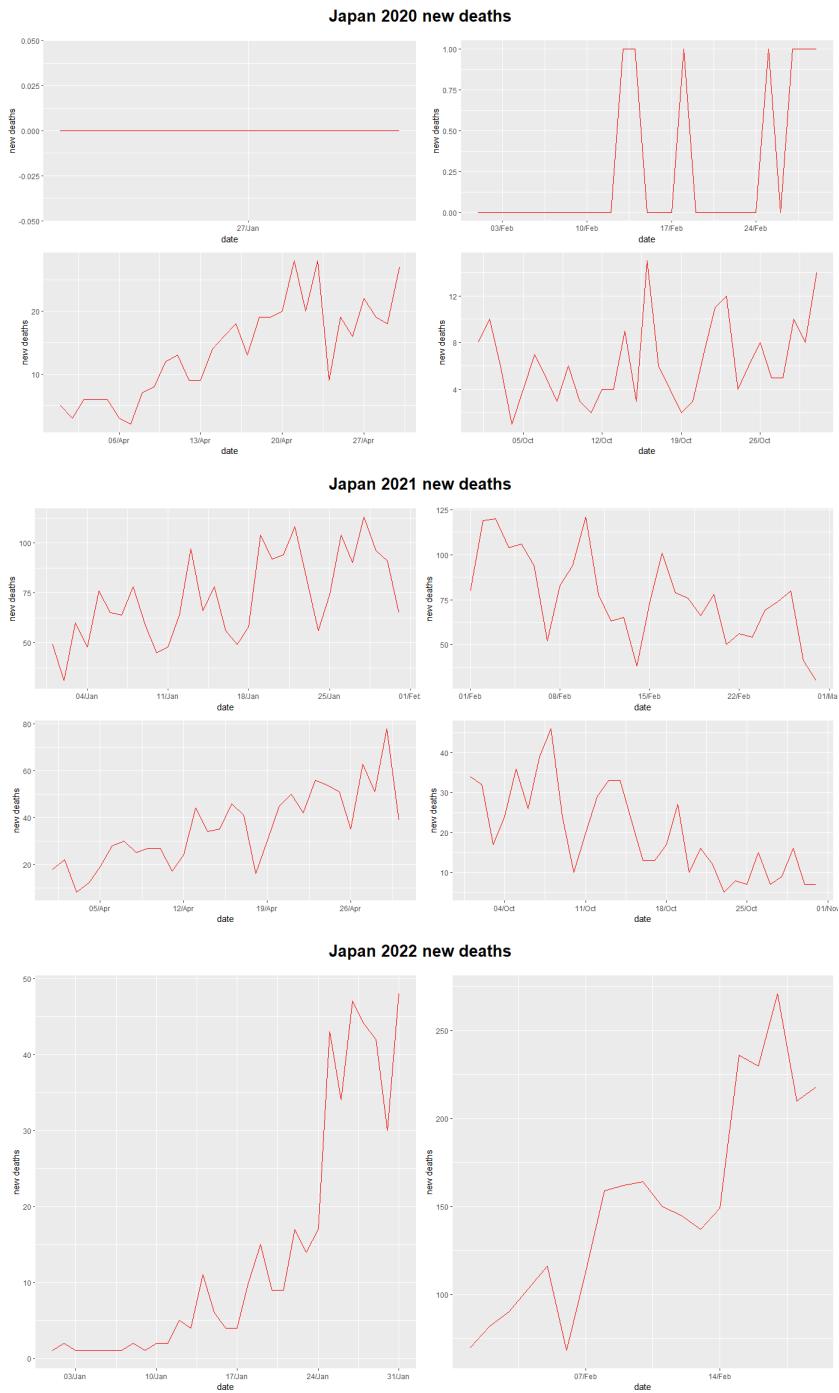
```
#IDN
plotdeaths(IDN2020,"04") -> t4
plotdeaths(IDN2020,"10") -> t10
plots <- plot_grid(
  t4 + theme(legend.position="none"),
  t10 + theme(legend.position="none"),
  axis = "tblr",
  align = "hv",
  nrow = 1,
  ncol = 2,
  rel_widths= c(1,1),
  rel_heights = c(1,1),
  scale = 1
)
title <- ggdraw() +
  draw_label("Indonesia 2020 new deaths", size="20", fontface="bold")
plot_grid(title, plots, ncol=1, rel_heights=c(0.1, 1))
```

- Tương tự như trên, plot grid() dể gộp các biểu đồ và thêm title

- Kết quả câu 2:
- + Indonesia



+ Japan



+ Vietnam





3 Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong cho từng tháng

```
#3
#function plot cases and deaths
plotcasesdeaths <- function(df,m){
  ggplot() +
    geom_line(getm(df,m),mapping=aes(x= date , y= new_cases,color="new_cases"))
    +
    geom_line(getm(df,m),mapping=aes(x= date , y= new_deaths,color="new_deaths"))
  ) +
  labs(x="date",y="new deaths") +
  theme(legend.title=element_blank())+
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
}
```

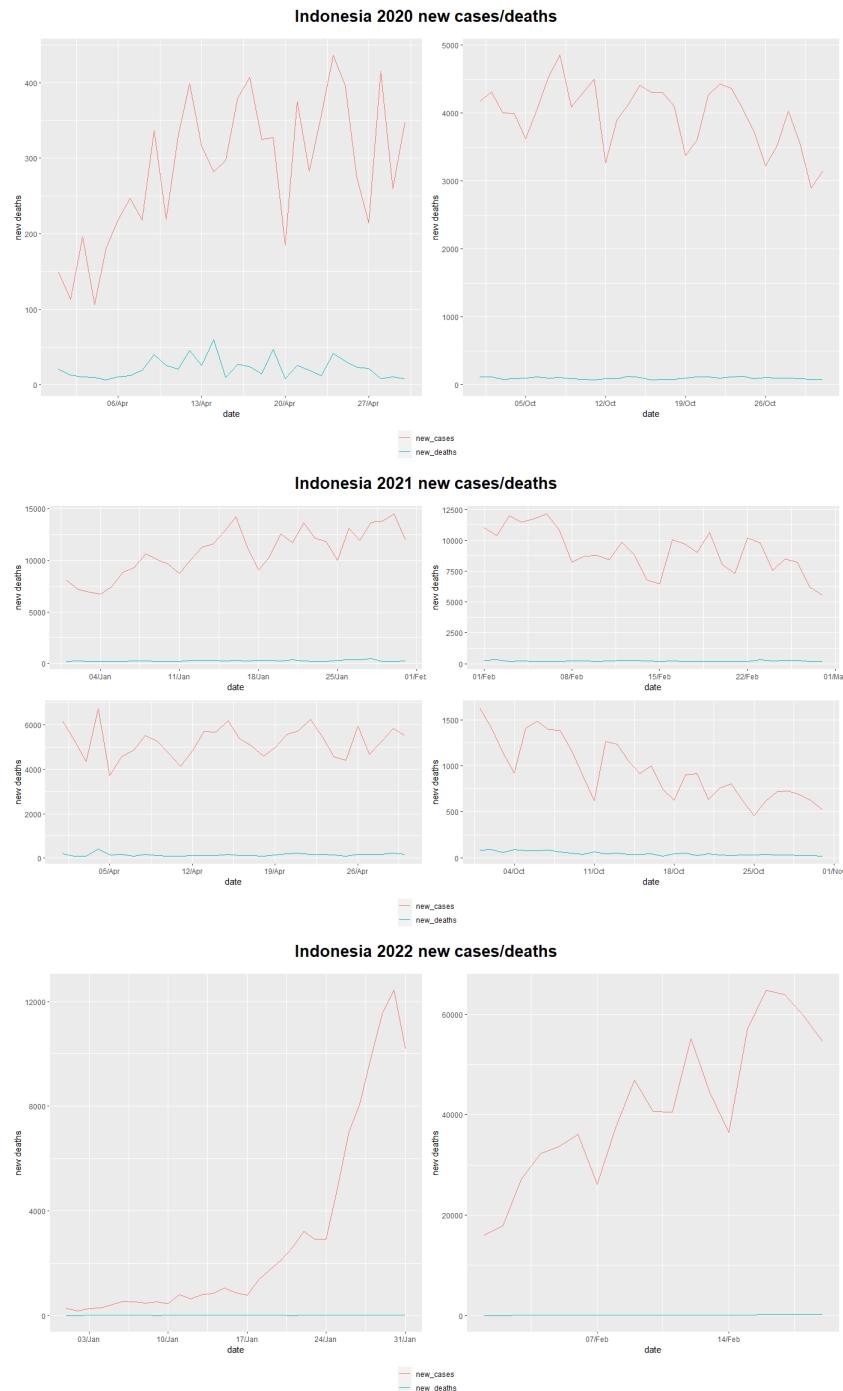
- Tương tự hàm plotcase, trả về plot theo tháng của datafram theo ca nhiễm và tử vong

```
#IDN
plotcasesdeaths(IDN2020,"04") -> t4
plotcasesdeaths(IDN2020,"10") -> t10
plots <- plot_grid(
  t4 + theme(legend.position="none"),
  t10 + theme(legend.position="none"),
  nrow = 1,
  ncol = 2,
  rel_widths= c(1,1),
  rel_heights = c(1,1),
  scale = 1
)
legend <- get_legend(t4)
title <- ggdraw() +
  draw_label("Indonesia 2020 new cases/deaths", size="20", fontface="bold")
plot_grid(
  title, plots,
  legend,
  ncol=1,
  rel_heights= c(0.1, 1,0.1)
)
```

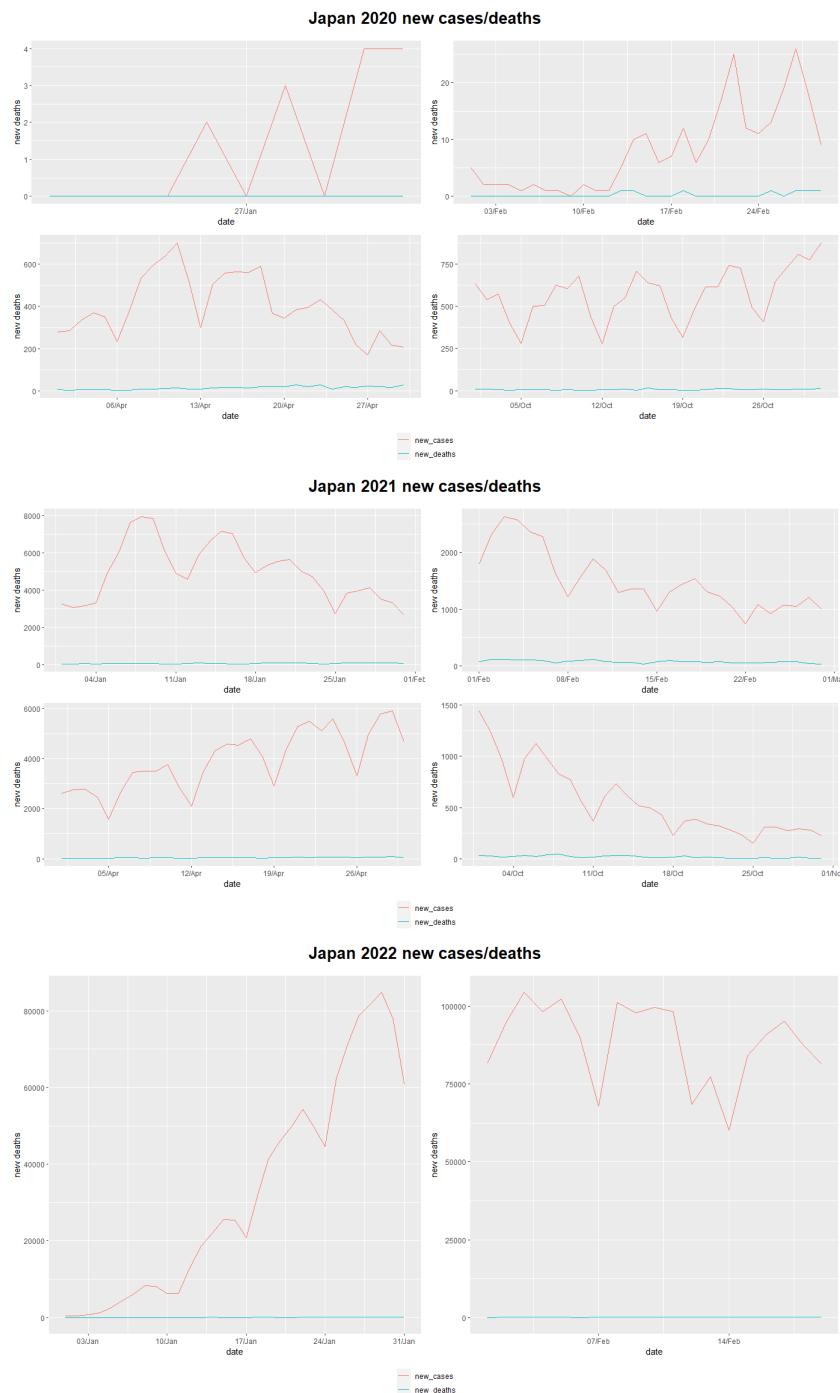
- Hàm getlegend() để lấy chú giải chung cho các biểu đồ

- Tương tự plotgrid gộp các biểu đồ và thêm title

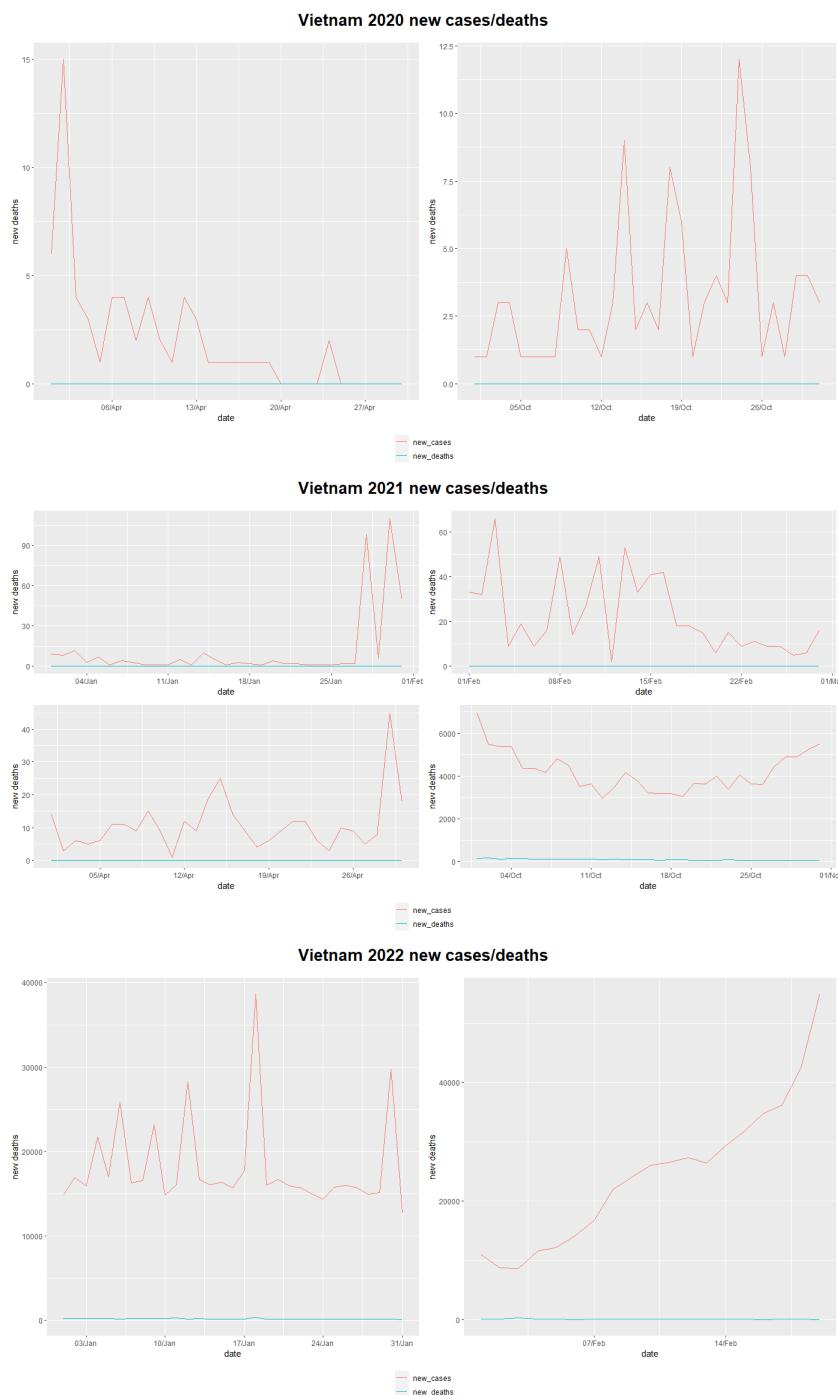
- Kết quả câu 3:
- + Indonesia



+ Japan



+ Vietnam



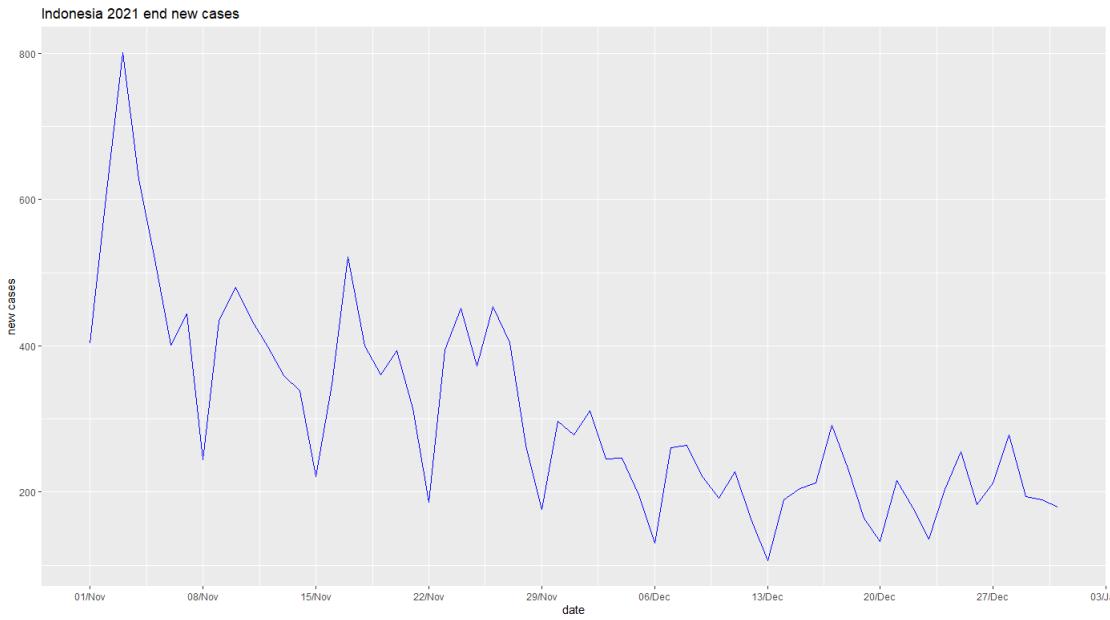
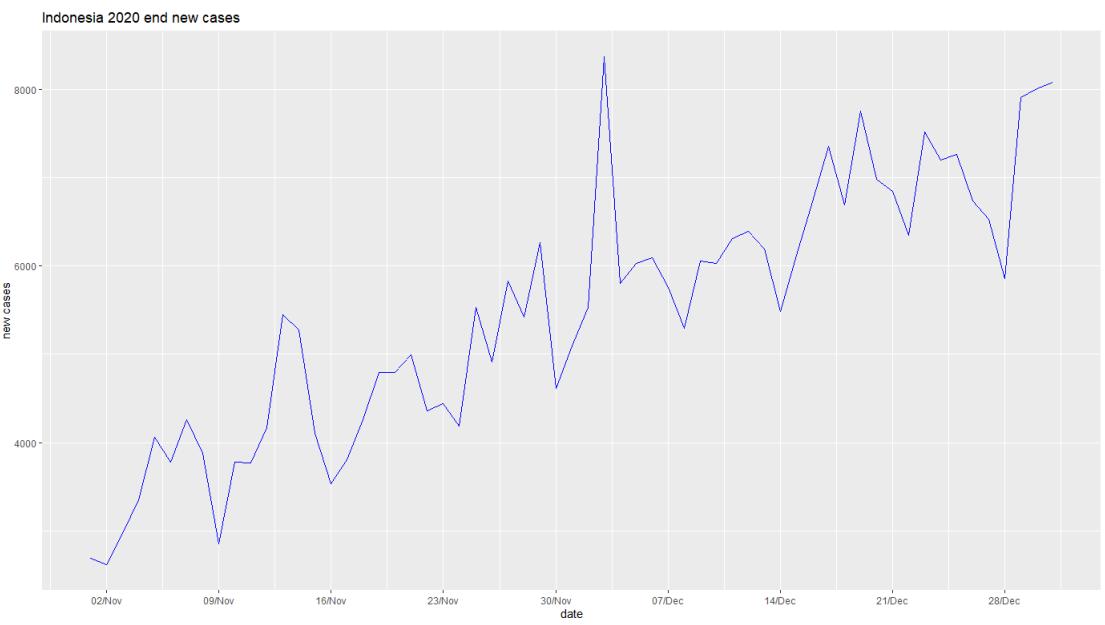


4 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh gồm 2 tháng cuối của năm

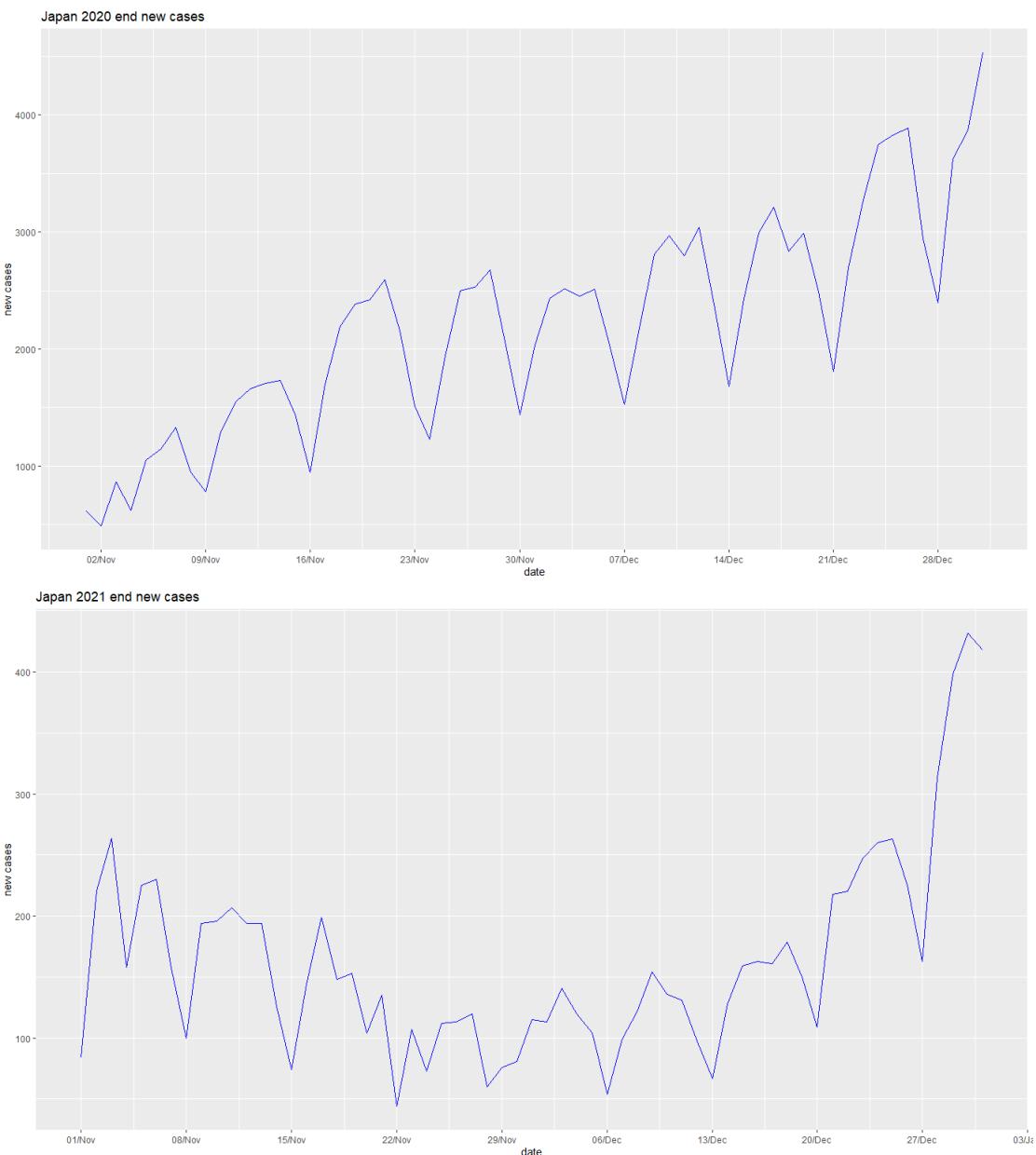
```
#4
#IDN
IDN2020end %>% ggplot() +
  geom_line(mapping=aes(x= date , y= new_cases),color="blue1") +
  labs(title = "Indonesia 2020 end new cases",x="date",y="new cases") +
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
IDN2021end %>% ggplot() +
  geom_line(mapping=aes(x= date , y= new_cases),color="blue1") +
  labs(title = "Indonesia 2021 end new cases",x="date",y="new cases") +
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
```

- dataframe -> ggplot theo ca nhiễm, loại geomline, đặt tên biểu đồ và cột

- Kết quả câu 4:  
+ Indonesia

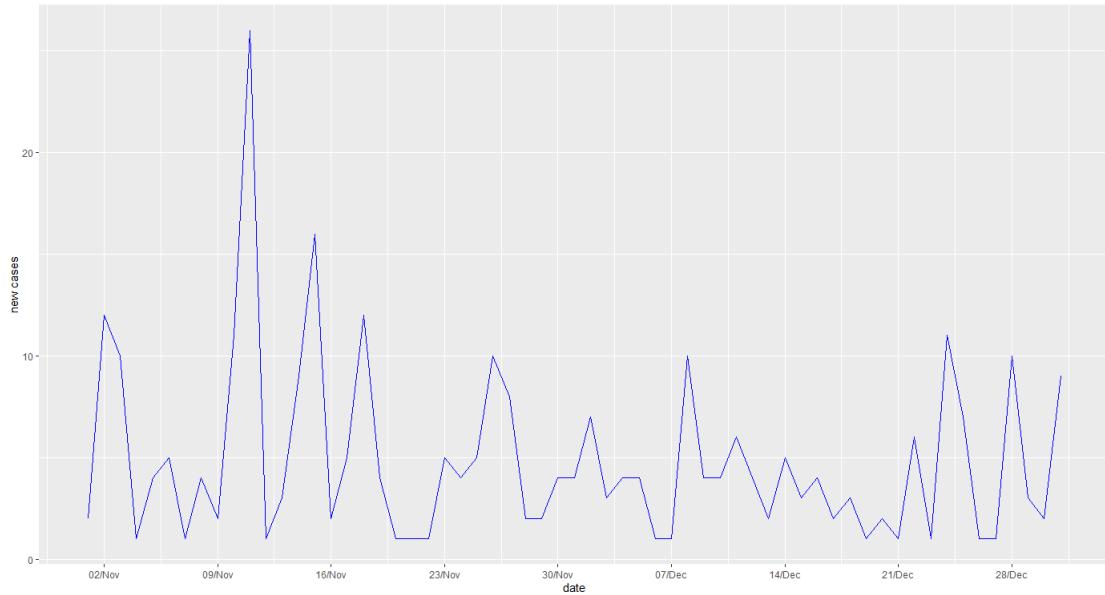


+ Japan

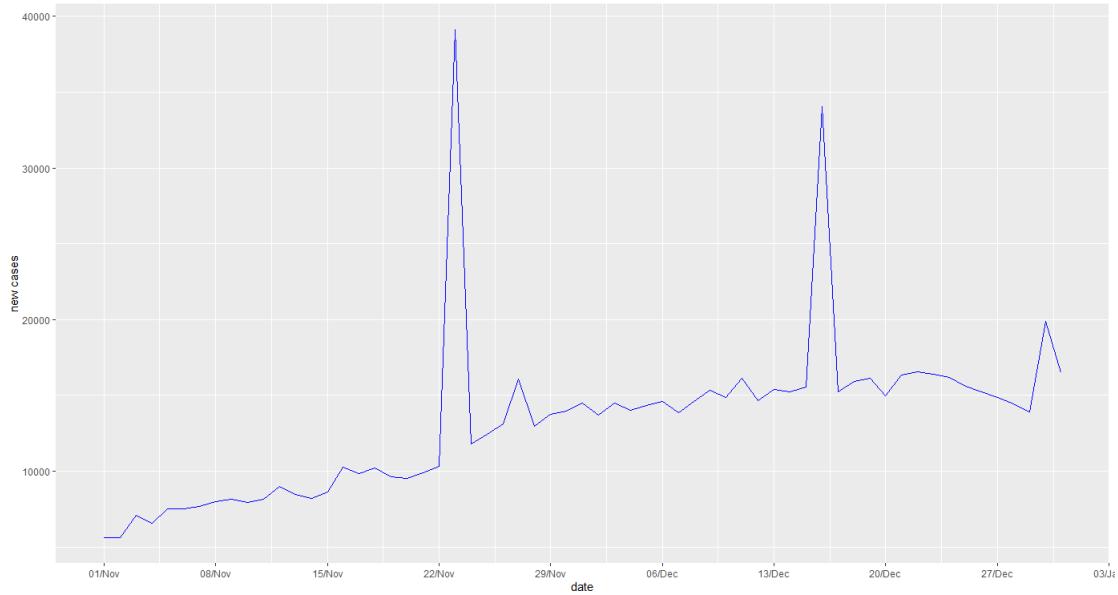


+ Vietnam

Vietnam 2020 end new cases



Vietnam 2021 end new cases





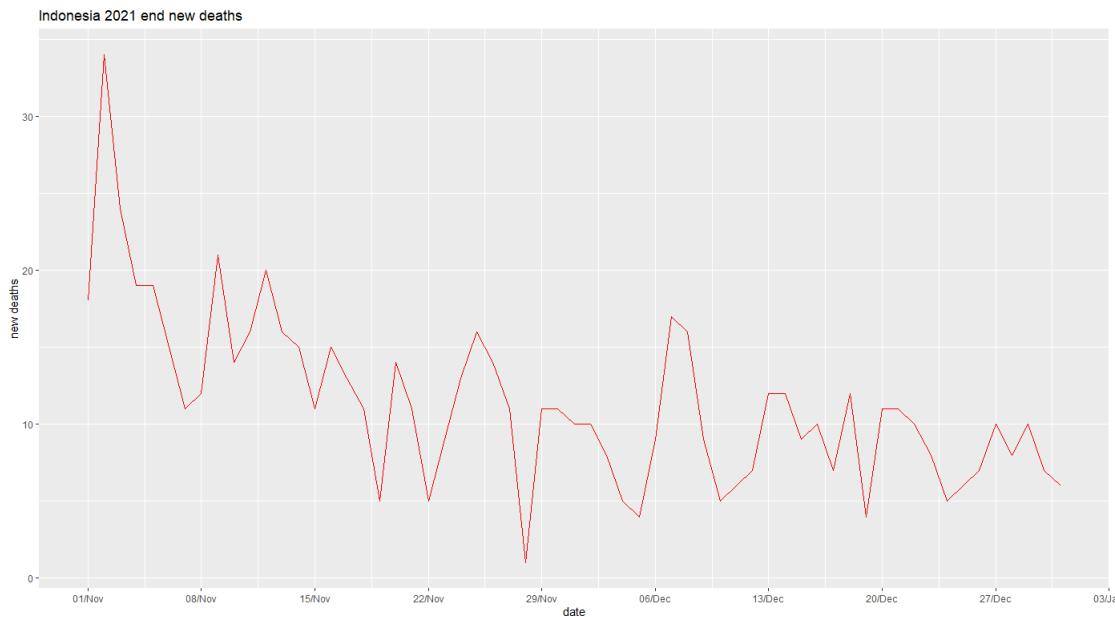
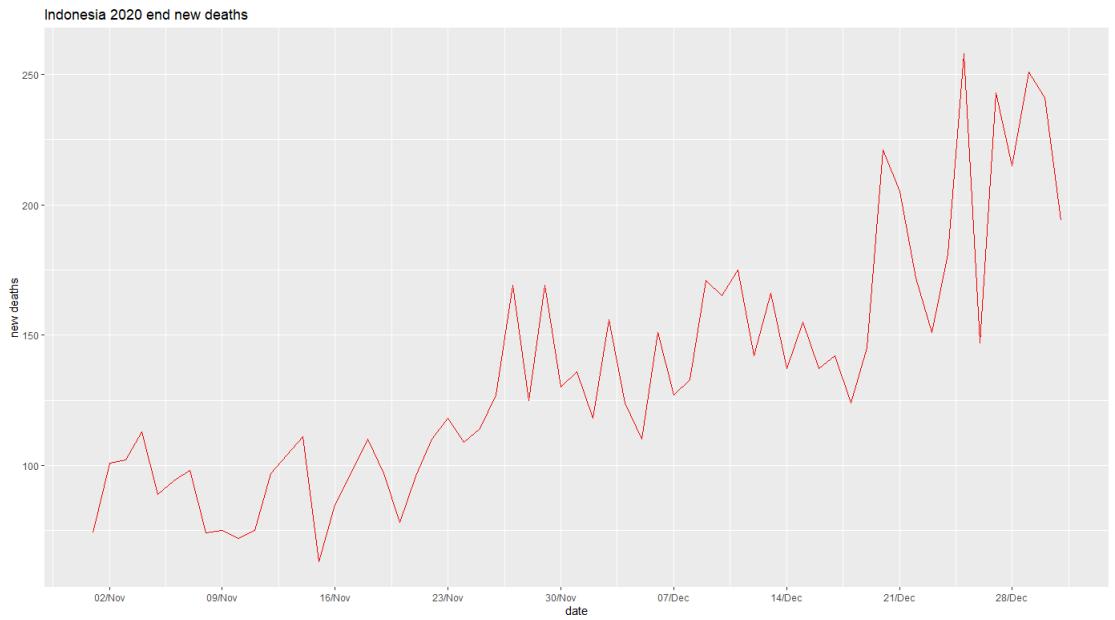
5 Biểu đồ thể hiện thu thập dữ liệu tử vong gồm 2 tháng cuối của năm

```
#5
IDN2020end %>% ggplot() +
  geom_line(mapping=aes(x= date , y= new_deaths),color="red2") +
  labs(title = "Indonesia 2020 end new deaths",x="date",y="new deaths") +
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
IDN2021end %>% ggplot() +
  geom_line(mapping=aes(x= date , y= new_deaths),color="red2") +
  labs(title = "Indonesia 2021 end new deaths",x="date",y="new deaths") +
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
```

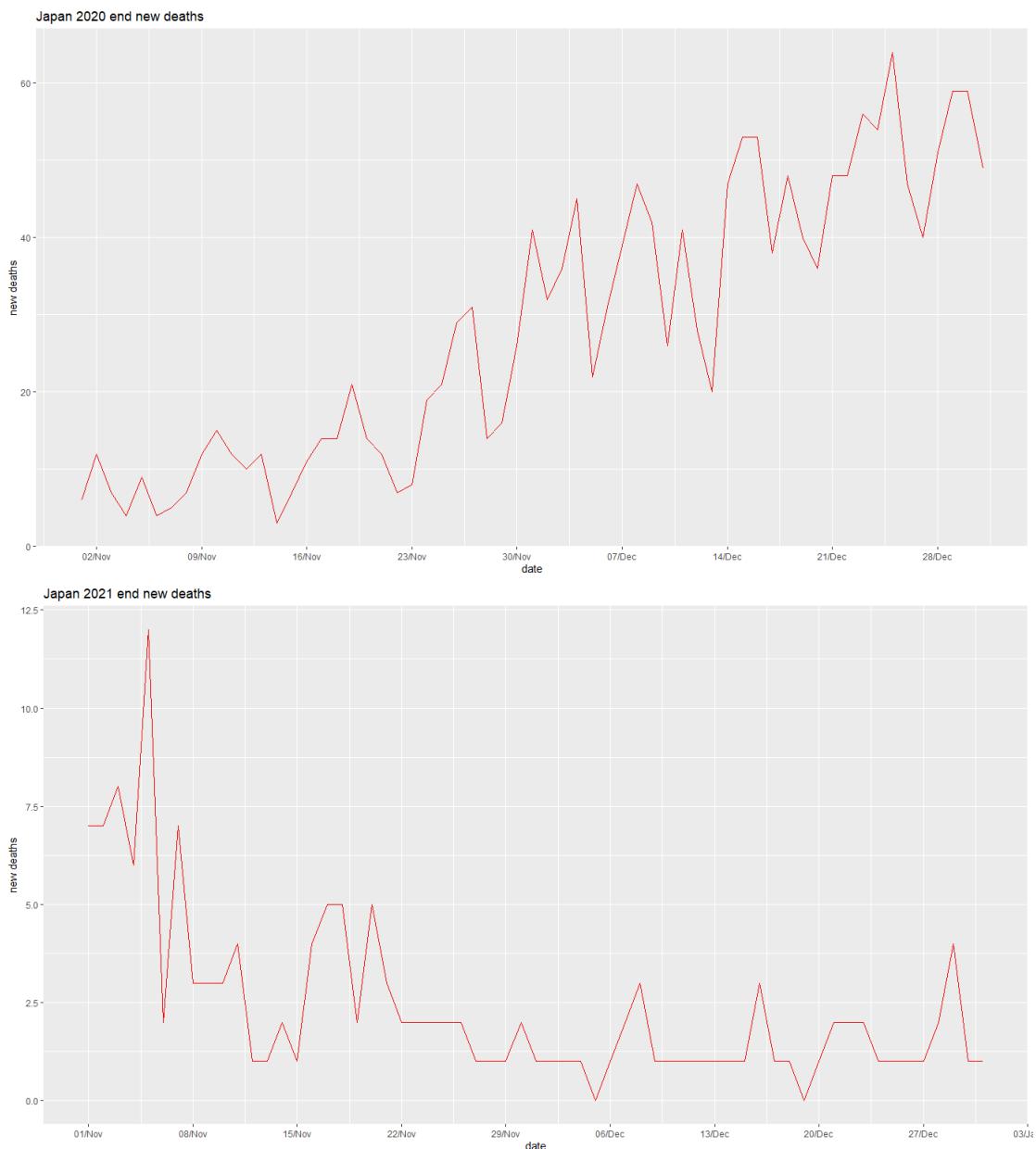
- tương tự biểu đồ ca nhiễm

- Kết quả câu 5:

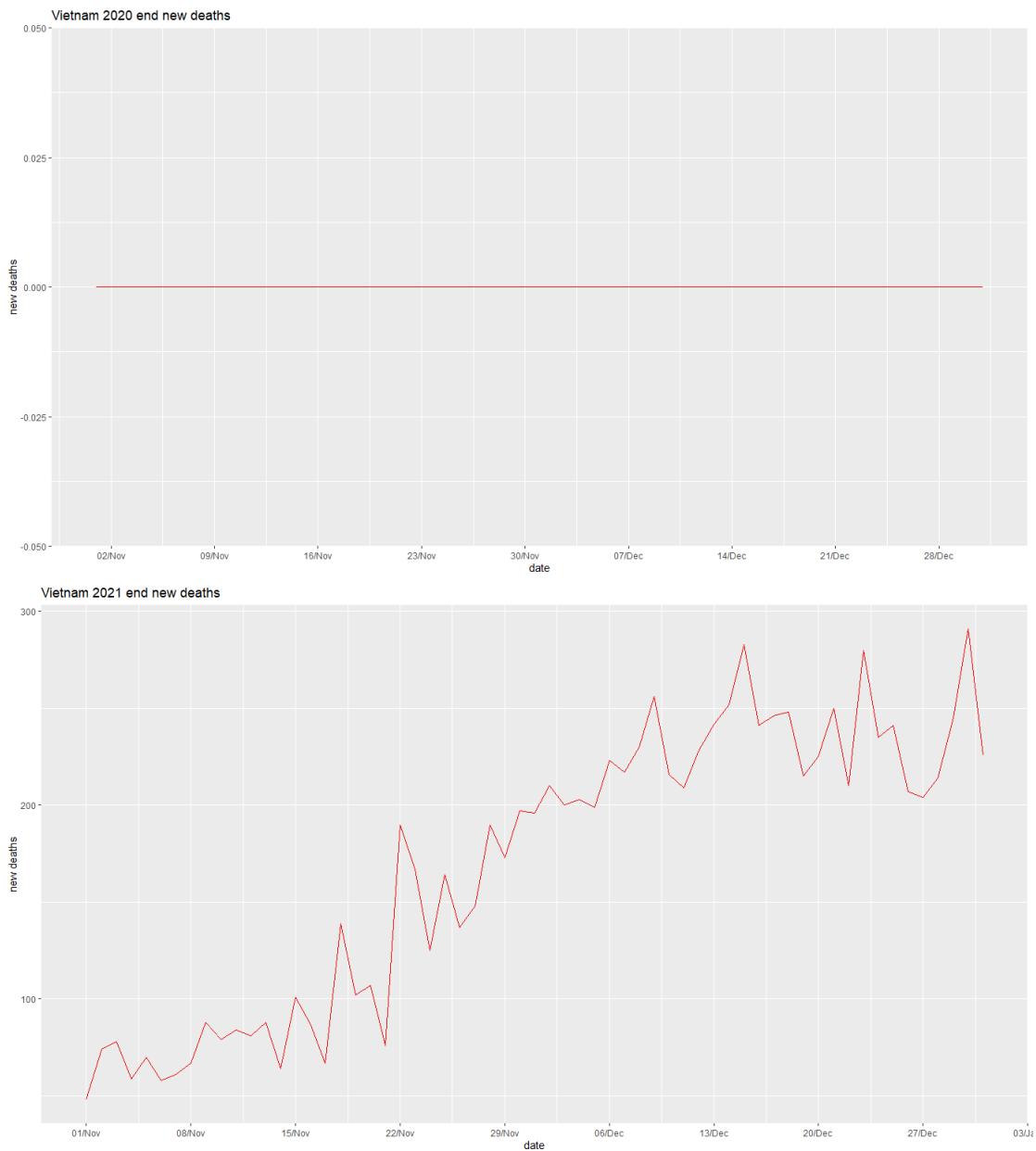
+ Indonesia



+ Japan



+ Vietnam





6 Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong gồm 2 tháng cuối của năm

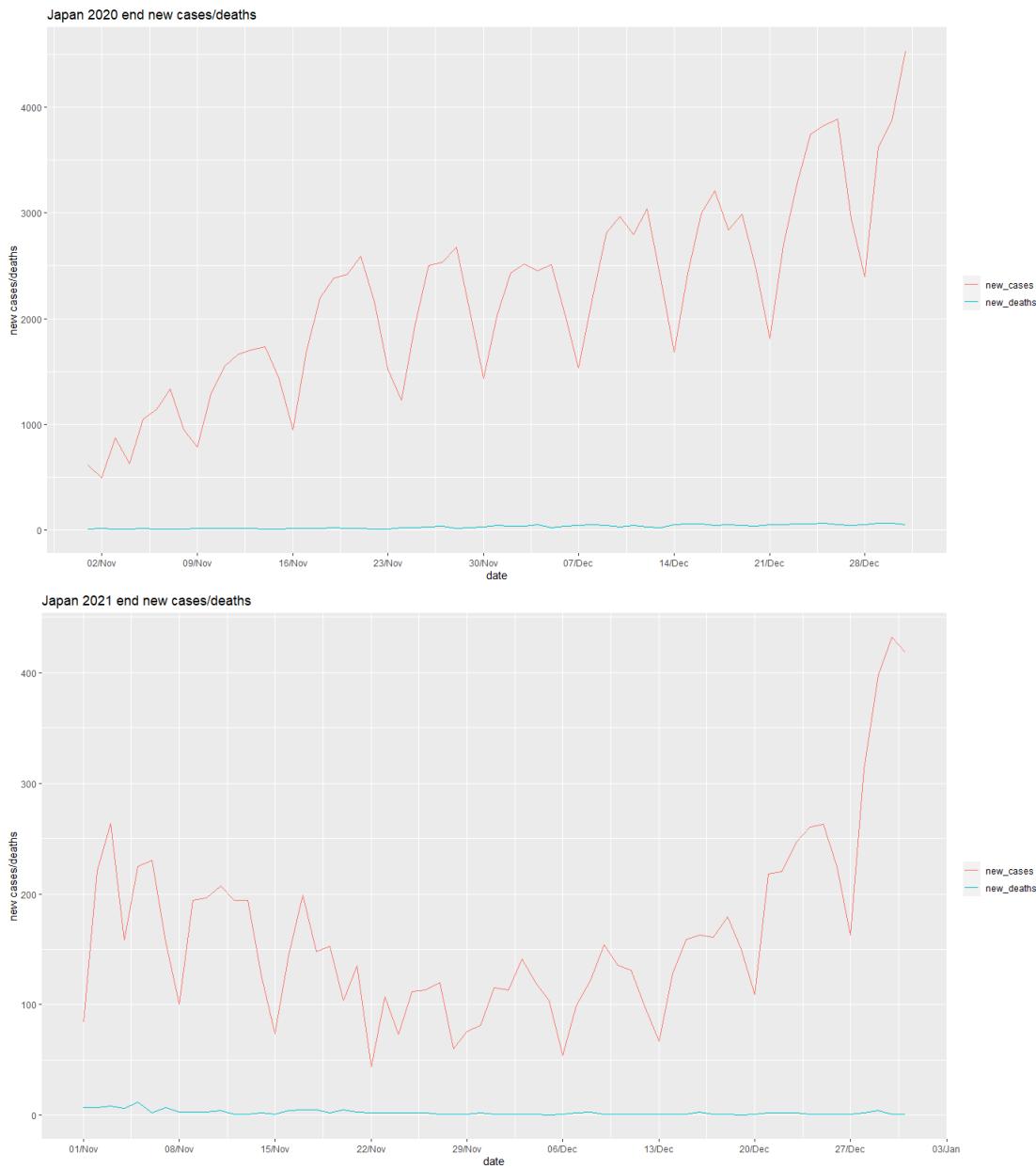
```
#6
#IDN
IDN2020end %>% ggplot() +
  geom_line(mapping=aes(x= date , y= new_cases ,color="new_cases")) +
  geom_line(mapping=aes(x= date , y= new_deaths ,color="new_deaths")) +
  labs(title = "Indonesia 2020 end new cases/deaths",x="date",y="new cases/
  deaths") +
  theme(legend.title=element_blank())+
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
IDN2021end %>% ggplot() +
  geom_line(mapping=aes(x= date , y= new_cases ,color="new_cases")) +
  geom_line(mapping=aes(x= date , y= new_deaths ,color="new_deaths")) +
  labs(title = "Indonesia 2021 end new cases/deaths",x="date",y="new cases/
  deaths") +
  theme(legend.title=element_blank())+
  scale_x_date(date_labels = "%d/%b",date_breaks= "weeks")
```

- biểu đồ gộp ca nhiễm và tử vong, chú giải hai màu newcases, new deaths

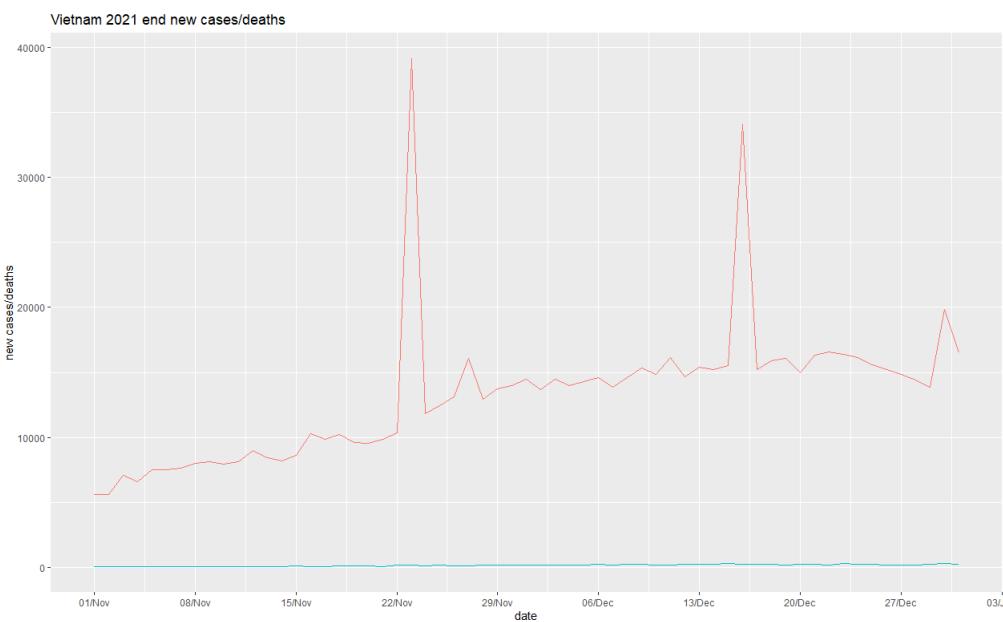
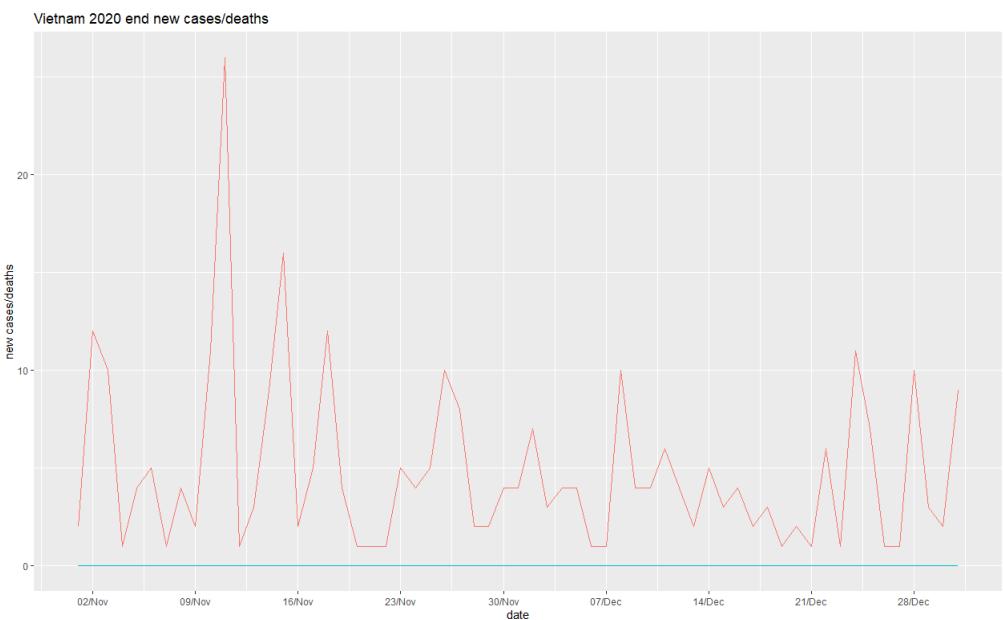
- Kết quả câu 6:  
+ Indonesia



+ Japan



+ Vietnam





7 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy cho từng tháng

```
#cumulative data
cumsumm <- function(df,m) {
  getm(df,m) %>% mutate(new_cases=cumsum(new_cases),new_deaths=cumsum(new_
  deaths))
}
```

- hàm tính tích lũy theo tháng m của dataframe

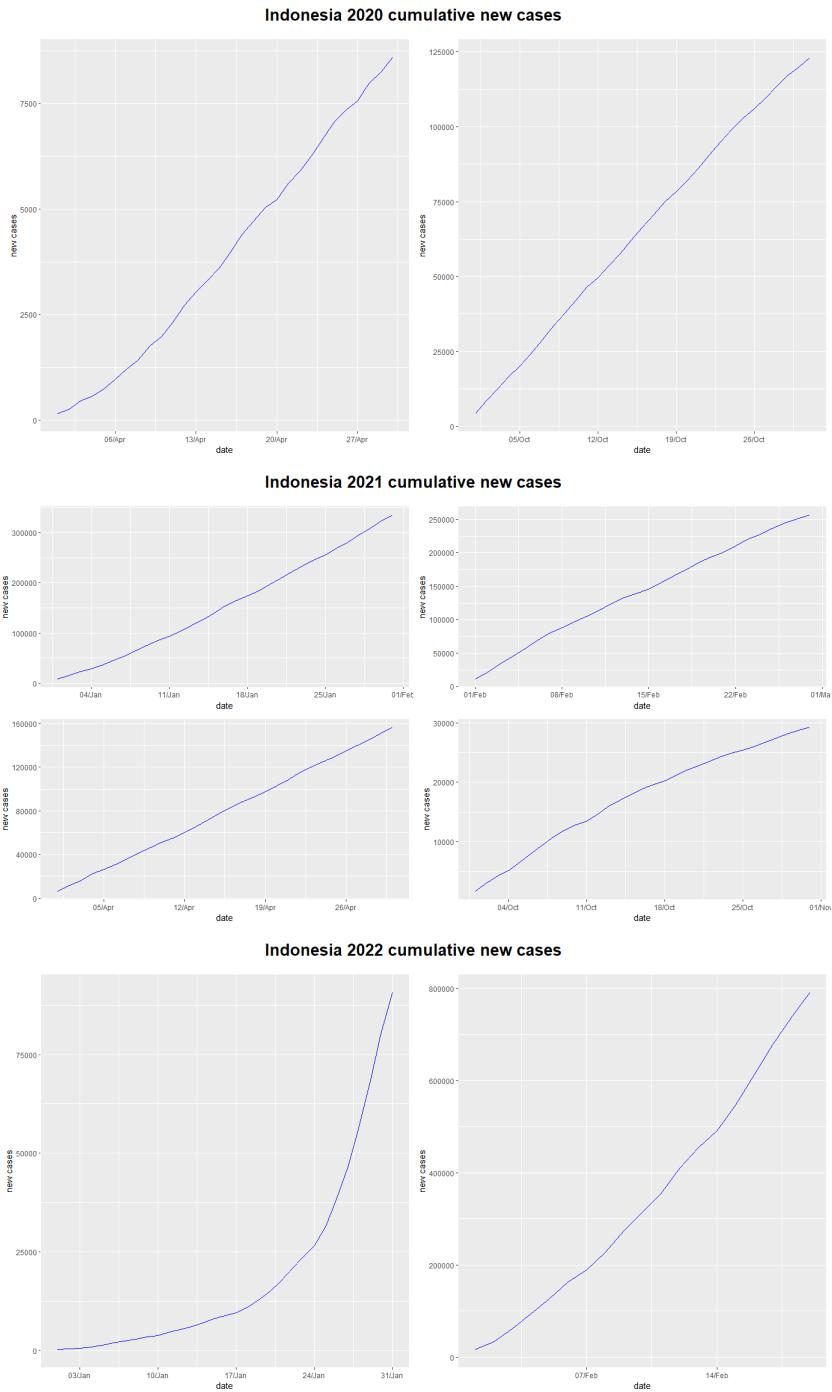
```
#IDN
cumsumm(IDN2020,"04") -> cst4
cumsumm(IDN2020,"10") -> cst10
CSIDN2020 <- rbind(cst4,cst10)
cumsumm(IDN2021,"01") -> cst1
cumsumm(IDN2021,"02") -> cst2
cumsumm(IDN2021,"04") -> cst4
cumsumm(IDN2021,"10") -> cst10
CSIDN2021 <- rbind(cst1,cst2,cst4,cst10)
cumsumm(IDN2022,"01") -> cst1
cumsumm(IDN2022,"02") -> cst2
CSIDN2022 <- rbind(cst1,cst2)
```

- các datafram CS chứa dữ liệu tích lũy tháng

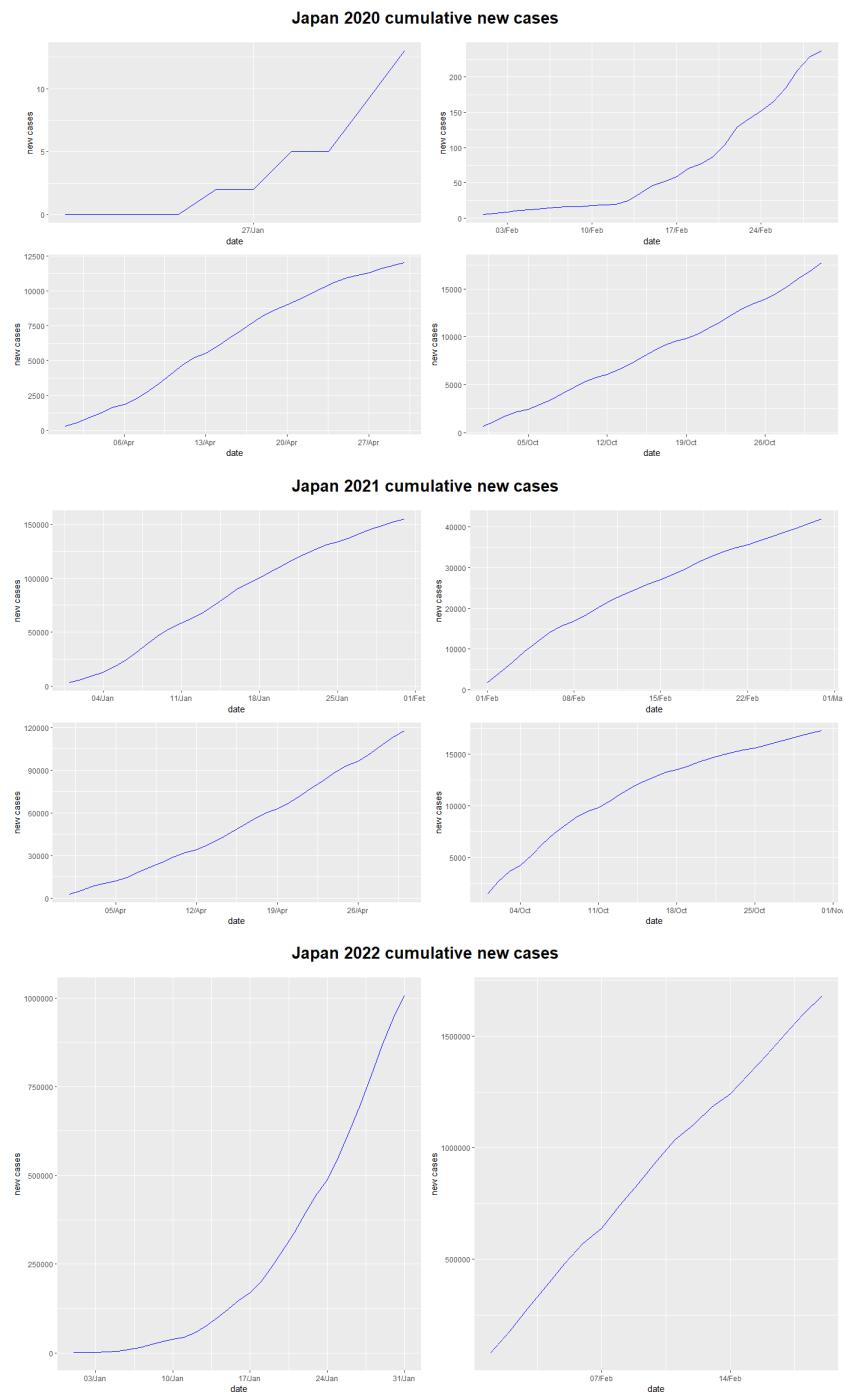
- dùng hàm plotcases/plotdeaths đã có tạo các plot tháng và dùng plotgrid để gộp

```
#7
plotcases(CSIDN2020,"04") -> t4
plotcases(CSIDN2020,"10") -> t10
plots <- plot_grid(
  t4 + theme(legend.position="none"),
  t10 + theme(legend.position="none"),
  axis = "tblr",
  align = "hv",
  nrow = 1,
  ncol = 2,
  rel_widths= c(1,1),
  rel_heights = c(1,1),
  scale = 1
)
title <- ggdraw() +
  draw_label("Indonesia 2020 cumulative new cases", size="20", fontface="bold")
plot_grid(title, plots, ncol=1, rel_heights=c(0.1, 1))
```

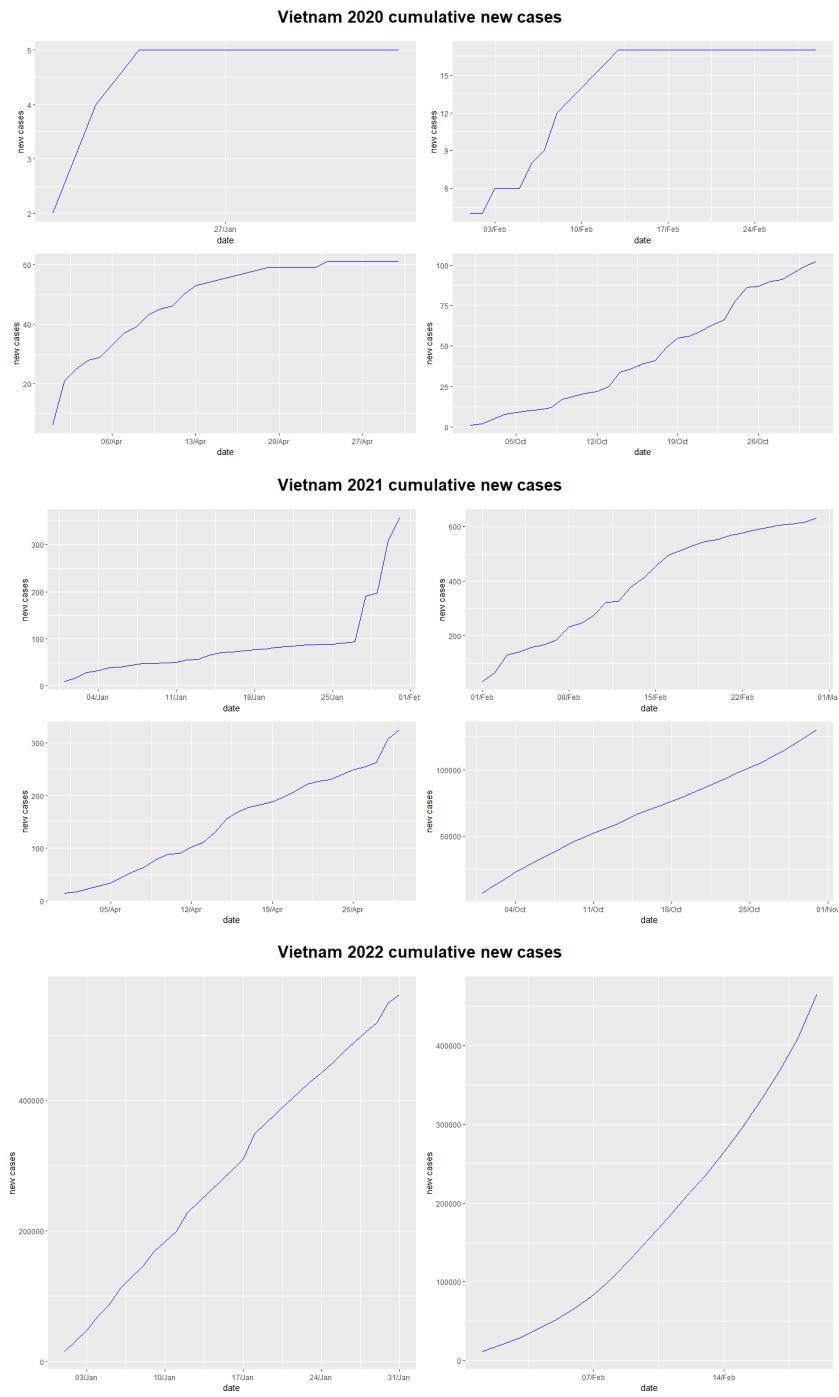
- Kết quả câu 7:
- + Indonesia



+ Japan



+ Vietnam

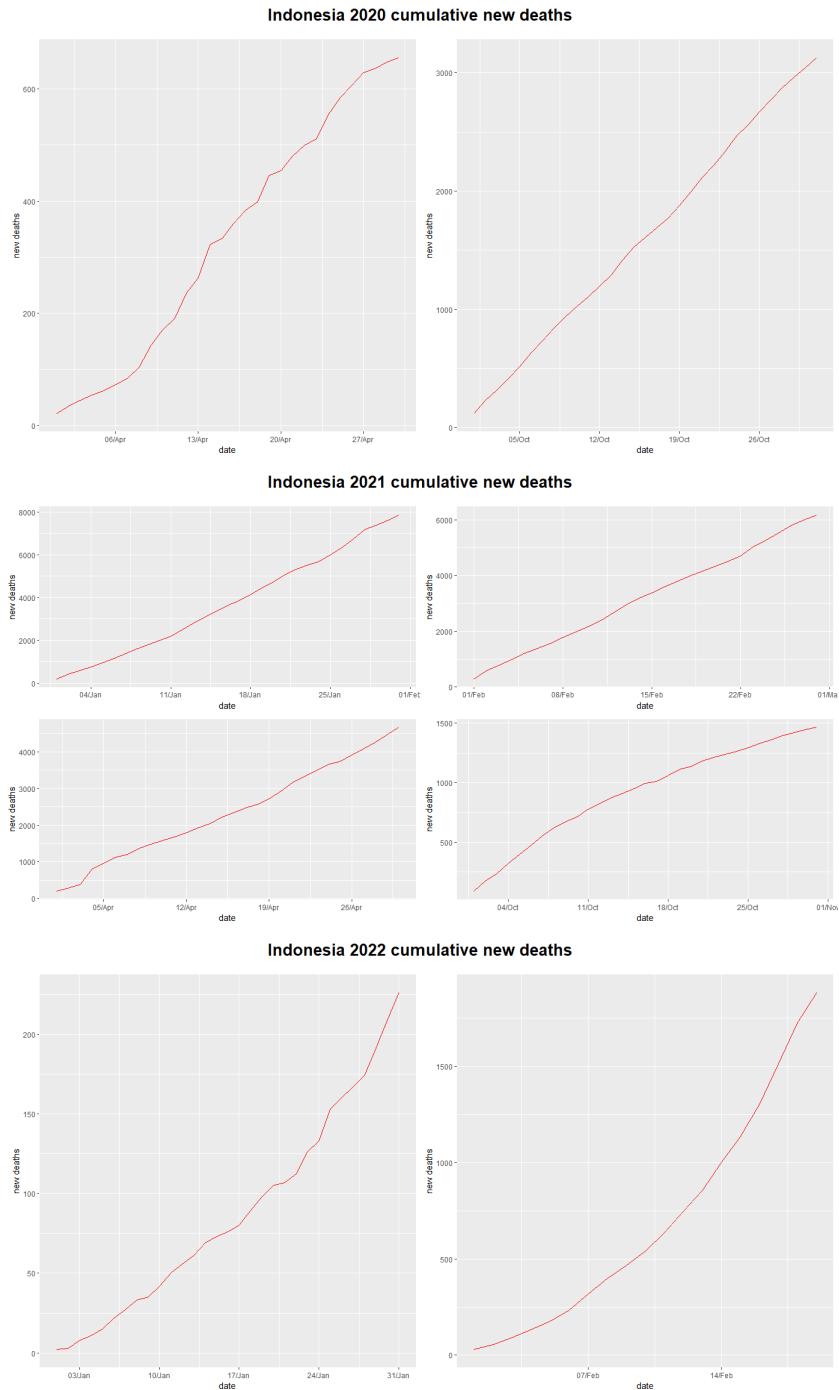




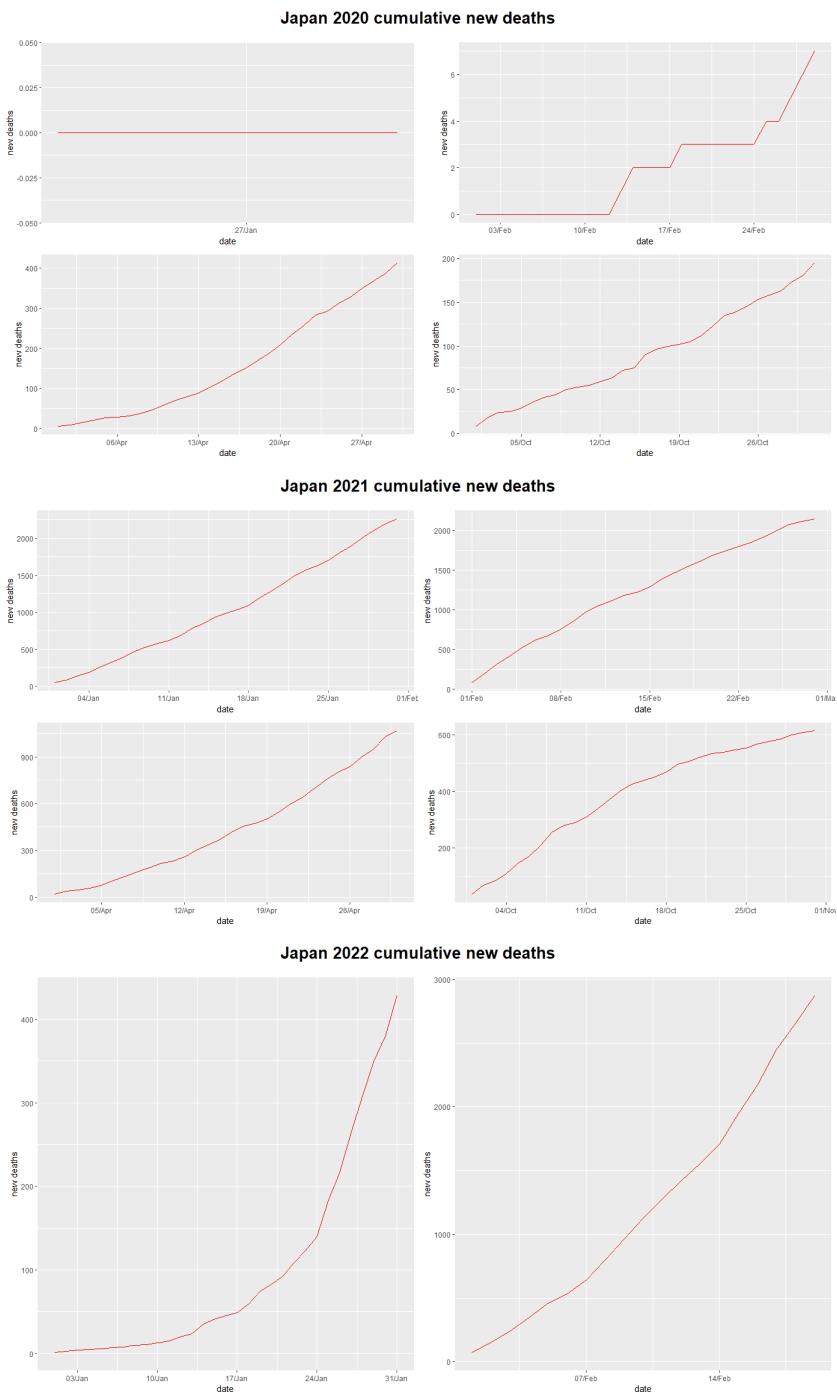
8 Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy cho từng tháng

```
#8
#IDN
plotdeaths(CSIDN2020,"04") -> t4
plotdeaths(CSIDN2020,"10") -> t10
plots <- plot_grid(
  t4 + theme(legend.position="none"),
  t10 + theme(legend.position="none"),
  axis = "tblr",
  align = "hv",
  nrow = 1,
  ncol = 2,
  rel_widths= c(1,1),
  rel_heights = c(1,1),
  scale = 1
)
title <- ggdraw() +
  draw_label("Indonesia 2020 cumulative new deaths", size="20", fontface="bold")
plot_grid(title, plots, ncol=1, rel_heights=c(0.1, 1))
```

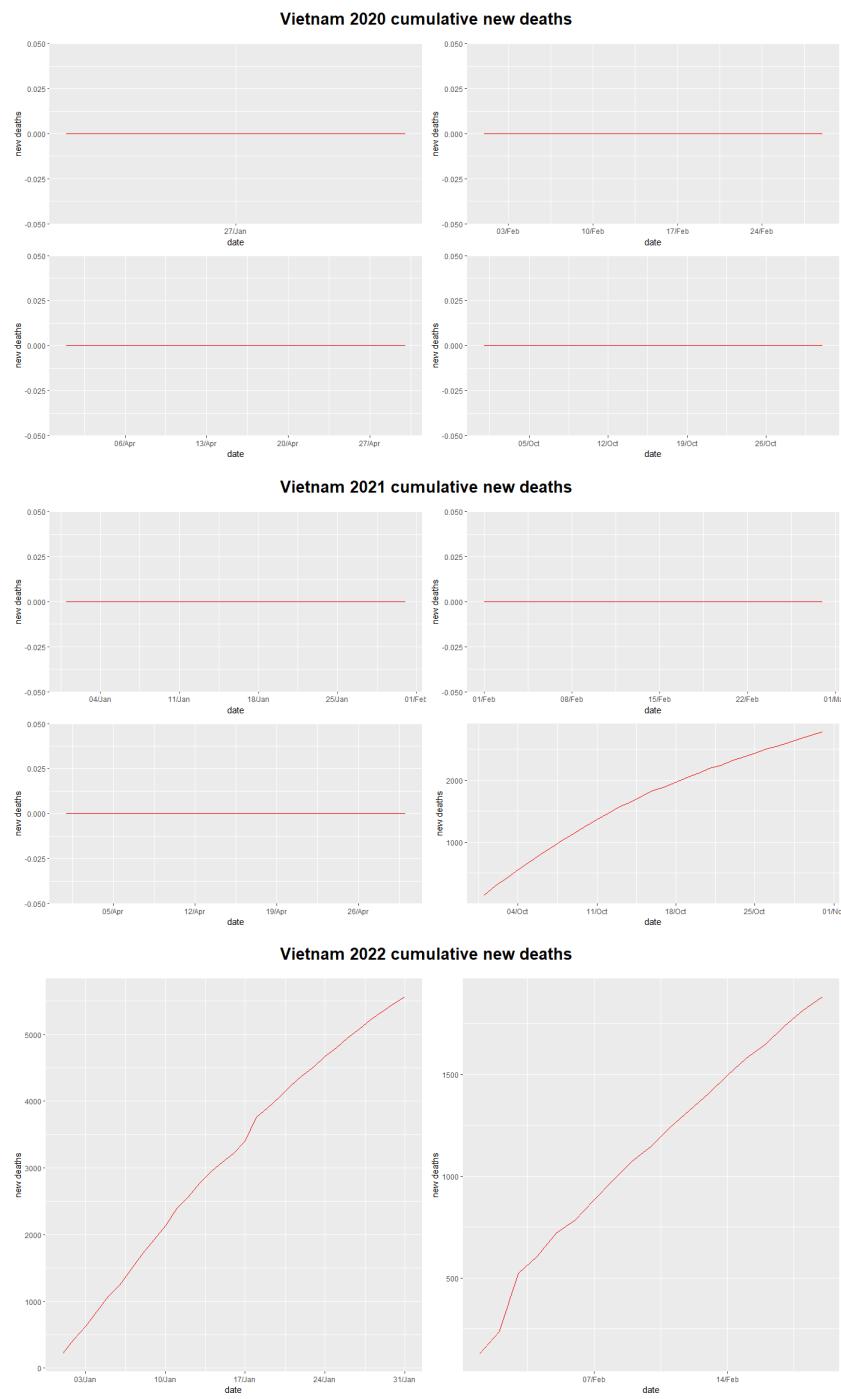
- Kết quả câu 8:
- + Indonesia



+ Japan



+ Vietnam





## vii Nhóm câu hỏi liên quan đến tất cả quốc gia theo thời gian là tháng

- Trên từng năm hãy vẽ biểu đồ thể hiện trục Ox là thời gian, trục Oy là nhiễm bệnh/tử vong. Hãy dùng 4 ký số của mã để để vẽ 4 tháng tương ứng theo ký số đó. Nếu ký số là 0 thì lấy tháng là 10.

1 Biểu đồ thu thập nhiễm bệnh theo thời gian là tháng của tất cả quốc gia

- Phương pháp làm: sẽ bao gồm những ý chính sau:

- Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file owid-covid-data.csv rồi lưu vào một dataframe, sau đó lấy ra những hàng ở cột location = "World". Tiếp theo lấy giá trị tuyệt đối cho những giá trị new\_cases và new\_deaths (để chuyển những giá trị âm thành dương).
- Bước 2: Ta lọc dữ liệu new\_cases với các tháng 1, 2, 4, 10 (theo MADE: 1204).
- Bước 3: Phần vẽ đồ thị:(4 đường riêng biệt thể hiện theo 4 tháng): Ta sử dụng hàm ggplot từ thư viện ggplot2 để vẽ đồ thị theo từng data frame nhỏ đã lọc ra.

- Code mà nhóm thực hiện: (nhóm em lấy đại diện năm 2021 vì sự thống kê sẽ diễn ra liên tục hơn)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

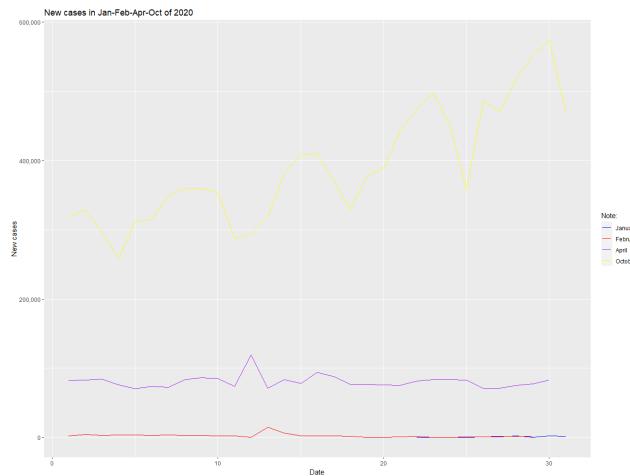
mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

thang1 <- mydata2[mydata2$date >= "2021-1-1" & mydata2$date <= "2021-1-31",
  ]
thang2 <- mydata2[mydata2$date >= "2021-2-1" & mydata2$date <= "2021-2-28",
  ]
thang4 <- mydata2[mydata2$date >= "2021-4-1" & mydata2$date <= "2021-4-30",
  ]
thang10 <- mydata2[mydata2$date >= "2021-10-1" & mydata2$date <= "2021-10-31",
  ]

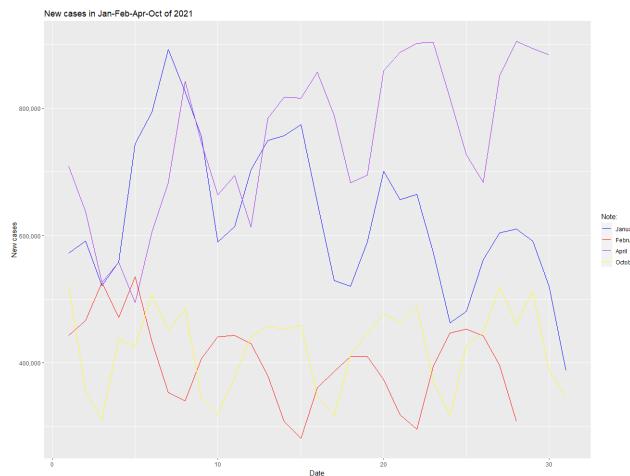
thang1$date <- day(thang1$date)
thang2$date = day(thang2$date)
thang4$date = day(thang4$date)
thang10$date = day(thang10$date)

ketqua <- ggplot()+
  geom_line(data = thang1, mapping = aes(x = date, y = new_cases, color = "January"))+
  geom_line(data = thang2, mapping = aes(x = date, y = new_cases, color = "February"))+
  geom_line(data = thang4, mapping = aes(x = date, y = new_cases, color = "April"))+
  geom_line(data = thang10, mapping = aes(x = date, y = new_cases, color = "October"))+
  scale_color_manual(name = "Note:", values = c("January" = "blue", "February" = "red", "April" = "purple", "October" = "yellow"))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma)
print(res + ggtitle("New cases in Jan-Feb-Apr-Oct of 2021") + labs(x = "Date", y = "New cases"))
```

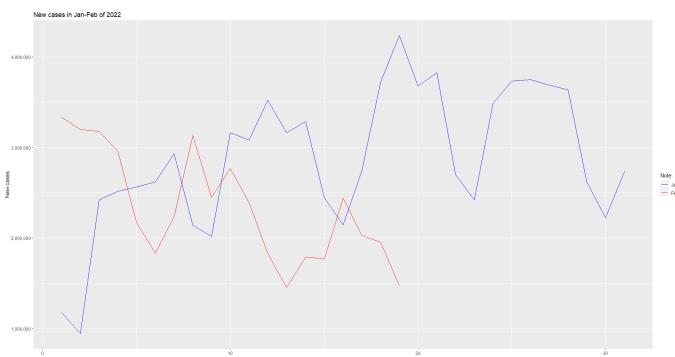
- Output câu 1 năm 2020:



- Output câu 1 năm 2021:



- Output câu 1 năm 2022:





2 Biểu đồ thu thập tử vong theo thời gian là tháng của tất cả quốc gia

- Phương pháp làm: tương tự như câu 1) phía trên nhưng là thay vẽ đồ thị cho dữ liệu của cột new\_cases thì ta sẽ vẽ đồ thị cho dữ liệu ở cột new\_deaths.

- Code mà nhóm thực hiện: (nhóm em lấy đại diện năm 2021 vì sự thống kê sẽ diễn ra liên tục hơn)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

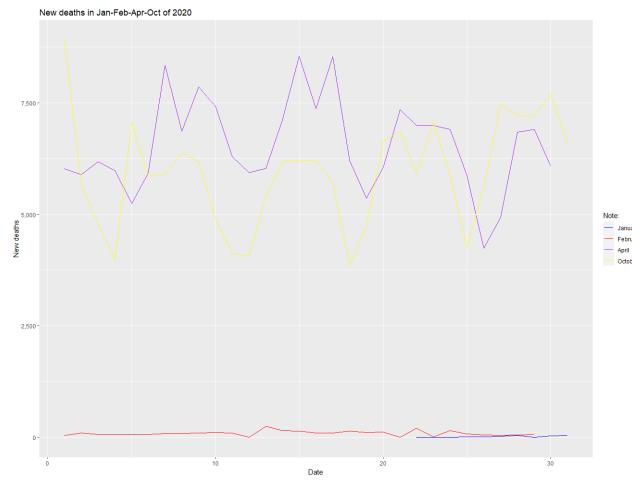
mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

thang1 <- mydata2[mydata2$date >= "2021-1-1" & mydata2$date <= "2021-1-31",
  ]
thang2 <- mydata2[mydata2$date >= "2021-2-1" & mydata2$date <= "2021-2-28",
  ]
thang4 <- mydata2[mydata2$date >= "2021-4-1" & mydata2$date <= "2021-4-30",
  ]
thang10 <- mydata2[mydata2$date >= "2021-10-1" & mydata2$date <= "2021-10-
  31", ]

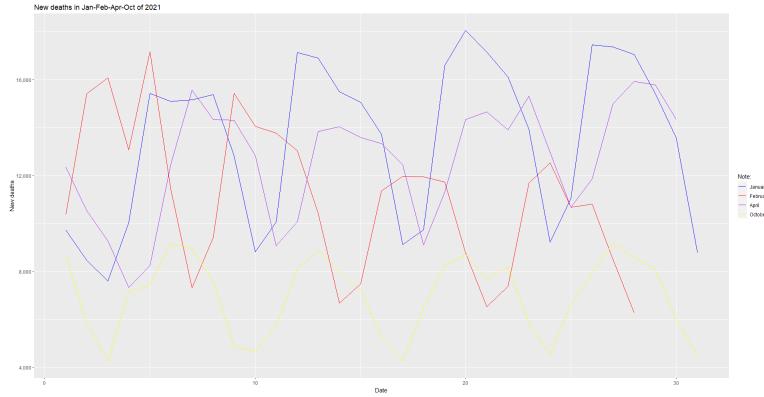
thang1$date <- day(thang1$date)
thang2$date = day(thang2$date)
thang4$date = day(thang4$date)
thang10$date = day(thang10$date)

ketqua <- ggplot()+
  geom_line(data = thang1, mapping = aes(x = date, y = new_deaths, color
    = "January"))+
  geom_line(data = thang2, mapping = aes(x = date, y = new_deaths, color
    = "February"))+
  geom_line(data = thang4, mapping = aes(x = date, y = new_deaths, color
    = "April"))+
  geom_line(data = thang10, mapping = aes(x = date, y = new_deaths,
    color = "October"))+
  scale_color_manual(name = "Note:", values = c("January" = "blue", "February" = "red", "April" = "purple", "October" = "yellow"))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma)
print(res + ggtitle("New deaths in Jan-Feb-Apr-Oct of 2021") + labs(x =
  "Date", y = "New deaths"))
```

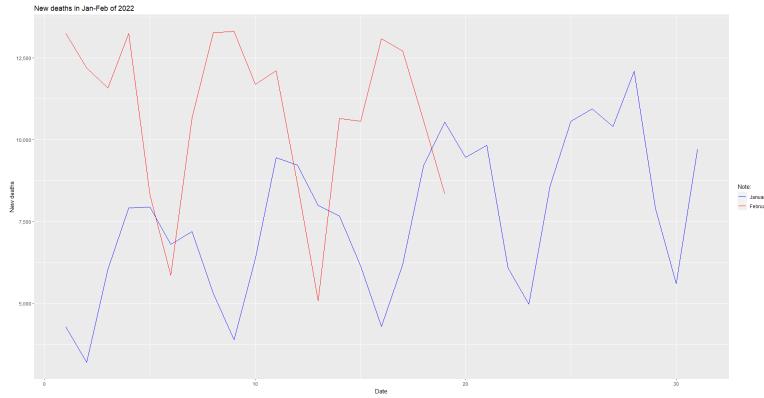
- Output câu 2 năm 2020:



- Output câu 2 năm 2021:



- Output câu 2 năm 2022:





3 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo thời gian là 2 tháng cuối của năm của tất cả quốc gia

- Phương pháp làm: sẽ bao gồm những ý chính sau:

- Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file owid-covid-data.csv rồi lưu vào một data frame, sau đó lấy ra những hàng ở cột location có tên là "World". Tiếp theo lấy giá trị tuyệt đối cho những giá trị new\_cases và new\_deaths (để chuyển những giá trị âm thành dương).
- Bước 2: Ta lọc dữ liệu new\_cases với 2 tháng 11 và 12 vào 2 data frame.
- Bước 3: Phần vẽ đồ thị (2 đường riêng biệt thể hiện theo 2 tháng): Ta sử dụng hàm ggplot từ thư viện ggplot2 để vẽ đồ thị theo 2 data frame đã lọc ra.

- Code mà nhóm thực hiện: (nhóm em lấy đại diện năm 2021 vì sự thống kê sẽ diễn ra liên tục hơn)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

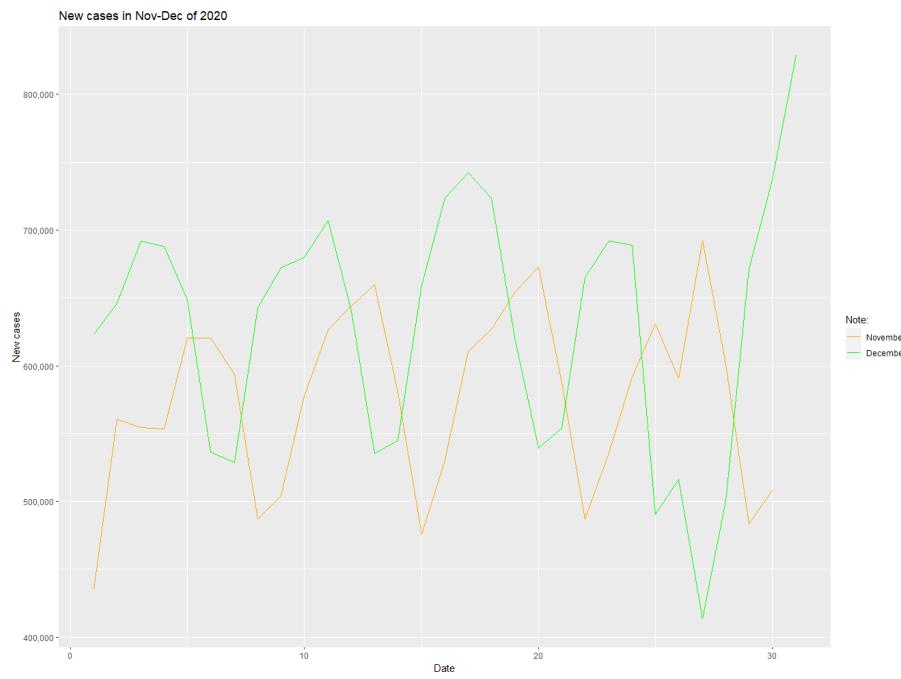
mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

thang11 <- mydata2[mydata2$date >= "2021-11-1" & mydata2$date <= "2021-11-30", ]
thang12 <- mydata2[mydata2$date >= "2021-12-1" & mydata2$date <= "2021-12-31", ]

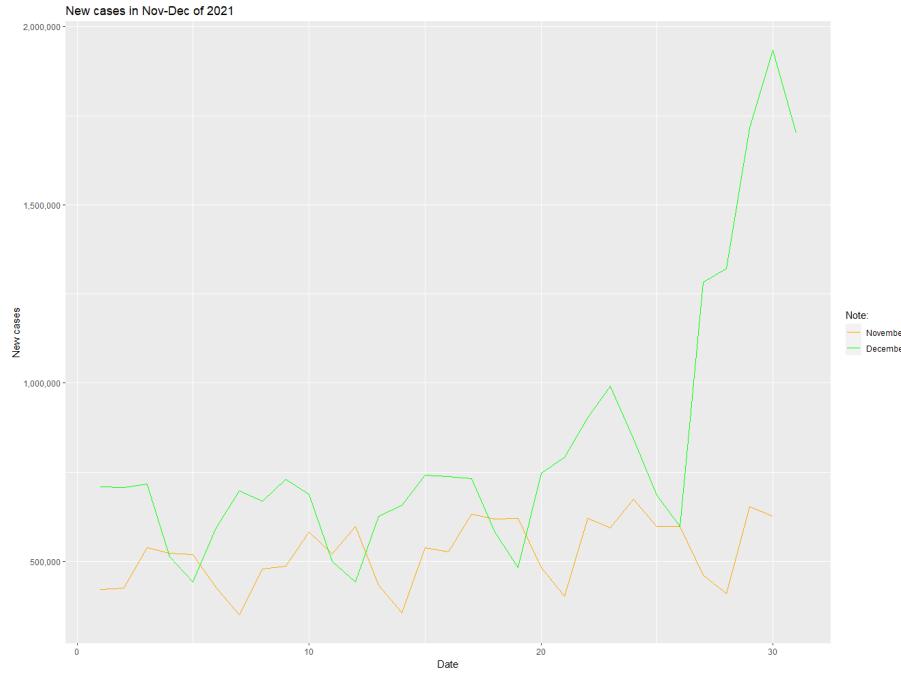
thang11$date <- day(thang11$date)
thang12$date <- day(thang12$date)

ketqua <- ggplot() +
  geom_line(data = thang11, mapping = aes(x = date, y = new_cases, color =
  "November"))+
  geom_line(data = thang12, mapping = aes(x = date, y = new_cases, color =
  "December"))+
  scale_color_manual(name = "Note:", values = c("November" = "orange", "December" = "green"))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma)
print(res + ggtitle("New cases in Nov-Dec of 2021") + labs(x = "Date", y =
  "New cases"))
```

- Output câu 3 năm 2020:



- Output câu 3 năm 2021:





- 4 Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là 2 tháng cuối của năm của tất cả quốc gia  
- Phương pháp làm: tương tự như câu 3) phía trên nhưng là thay vẽ đồ thị cho dữ liệu của cột new\_cases thì ta sẽ vẽ đồ thị cho dữ liệu ở cột new\_deaths.

- Code mà nhóm thực hiện: (nhóm em lấy đại diện năm 2021 vì sự thống kê sẽ diễn ra liên tục hơn)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

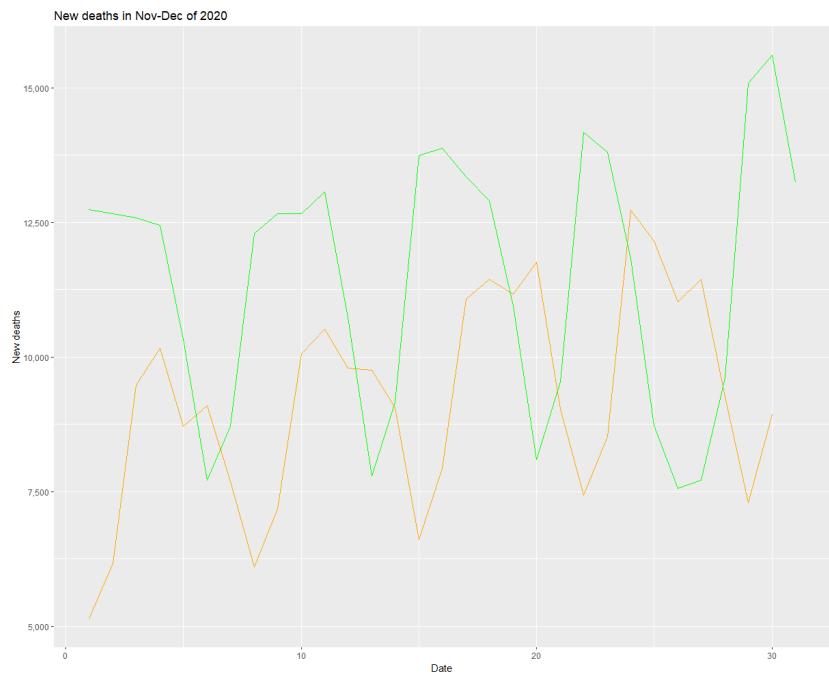
mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

thang11 <- mydata2[mydata2$date >= "2021-11-1" & mydata2$date <= "2021-11-30", ]
thang12 <- mydata2[mydata2$date >= "2021-12-1" & mydata2$date <= "2021-12-31", ]

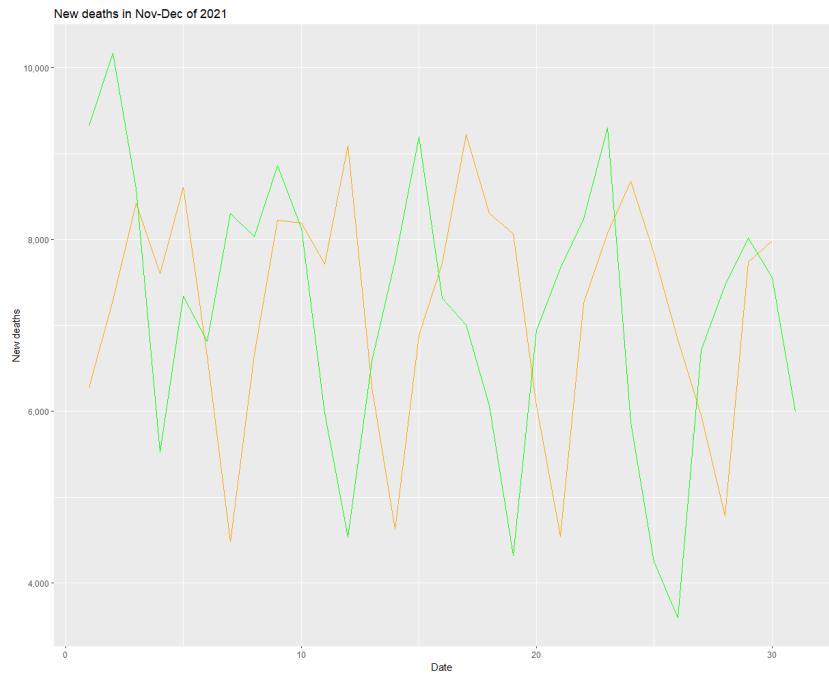
thang11$date <- day(thang11$date)
thang12$date <- day(thang12$date)

ketqua <- ggplot()+
  geom_line(data = thang11, mapping = aes(x = date, y = new_deaths,
  color = "November"))+
  geom_line(data = thang12, mapping = aes(x = date, y = new_deaths,
  color = "December"))+
  scale_color_manual(name = "Note:", values = c("November" = "orange",
  "December" = "green"))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma)
print(res + ggtitle("New deaths in Nov-Dec of 2021") + labs(x = "Date",
y = "New deaths"))
```

- Output câu 4 năm 2020:



- Output câu 4 năm 2021:





5 Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tương đối tích lũy theo thời gian là 2 tháng cuối của năm của tất cả quốc gia

- Phương pháp làm: sẽ bao gồm những ý chính sau:

- a) Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file owid-covid-data.csv rồi lưu vào một data frame, sau đó lấy ra những hàng ở cột location có tên là "World". Tiếp theo lấy giá trị tuyệt đối cho những giá trị new\_cases và new\_deaths (để chuyển những giá trị âm thành dương).
- b) Bước 2: Ta sử dụng hàm cumsum để tính tổng nhiễm bệnh tích lũy.
- c) Bước 3: Ta lọc dữ liệu new\_cases với các tháng 1, 2, 4, 10 (theo MADE: 1204).
- d) Bước 4: Phần vẽ đồ thị(4 đường riêng biệt thể hiện theo 4 tháng): Ta sử dụng hàm ggplot từ thư viện ggplot2 để vẽ đồ thị theo từng data frame nhỏ đã lọc ra.

- Code mà nhóm thực hiện: (nhóm em lấy đại diện năm 2021 vì sự thống kê sẽ diễn ra liên tục hơn)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

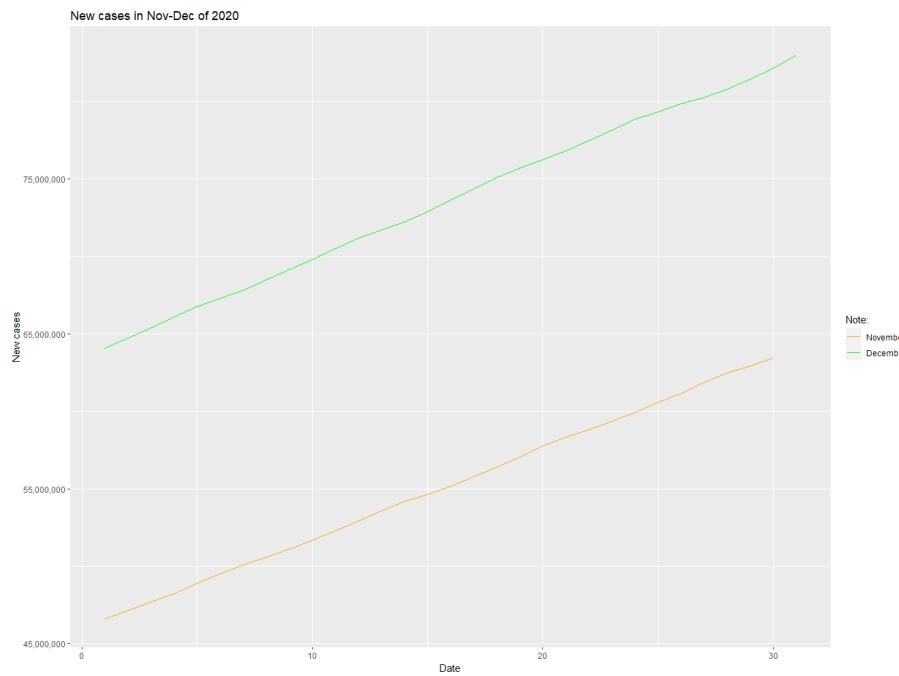
mydata2$new_cases <- cumsum(mydata2$new_cases)

thang11 <- mydata2[mydata2$date >= "2021-11-1" & mydata2$date <= "2021-11-30", ]
thang12 <- mydata2[mydata2$date >= "2021-12-1" & mydata2$date <= "2021-12-31", ]

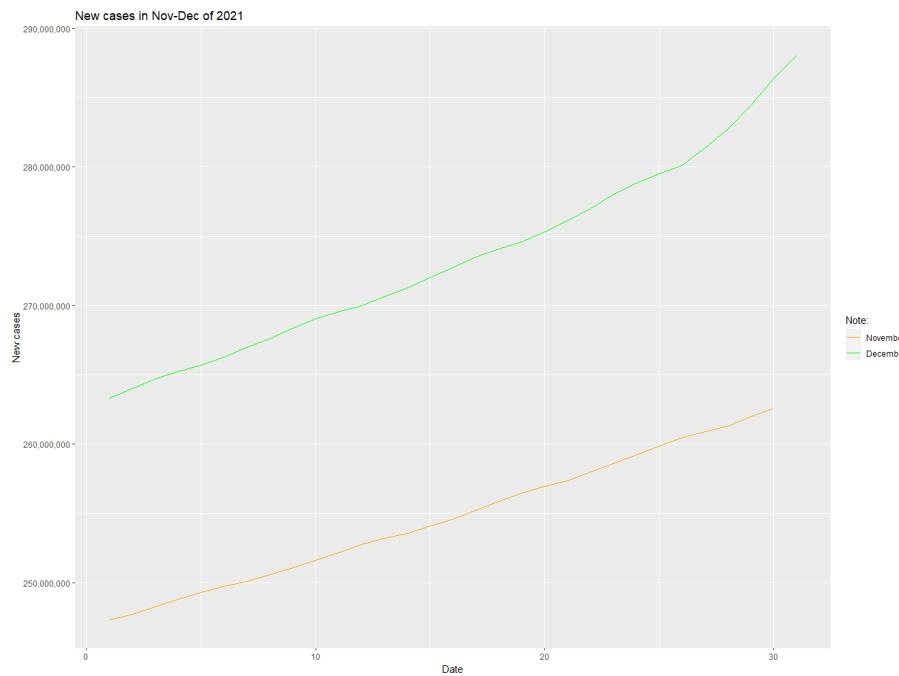
thang11$date <- day(thang11$date)
thang12$date <- day(thang12$date)

ketqua <- ggplot()+
  geom_line(data = thang11, mapping = aes(x = date, y = new_cases, color = "November"))+
  geom_line(data = thang12, mapping = aes(x = date, y = new_cases, color = "December"))+
  scale_color_manual(name = "Note:", values = c("November" = "orange", "December" = "green"))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma)
print(res + ggtitle("New cases in Nov-Dec of 2021") + labs(x = "Date", y = "New cases"))
```

- Output câu 5 năm 2020:



- Output câu 5 năm 2021:





6 Biểu đồ thể hiện thu thập dữ liệu tử vong tương đối tích lũy theo thời gian là 2 tháng cuối của năm của tất cả quốc gia

- Phương pháp làm: tương tự như câu 5) phía trên nhưng là thay vẽ đồ thị cho dữ liệu của cột new\_cases thì ta sẽ vẽ đồ thị cho dữ liệu ở cột new\_deaths.

- Code mà nhóm thực hiện: (nhóm em lấy đại diện năm 2021 vì sự thống kê sẽ diễn ra liên tục hơn)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

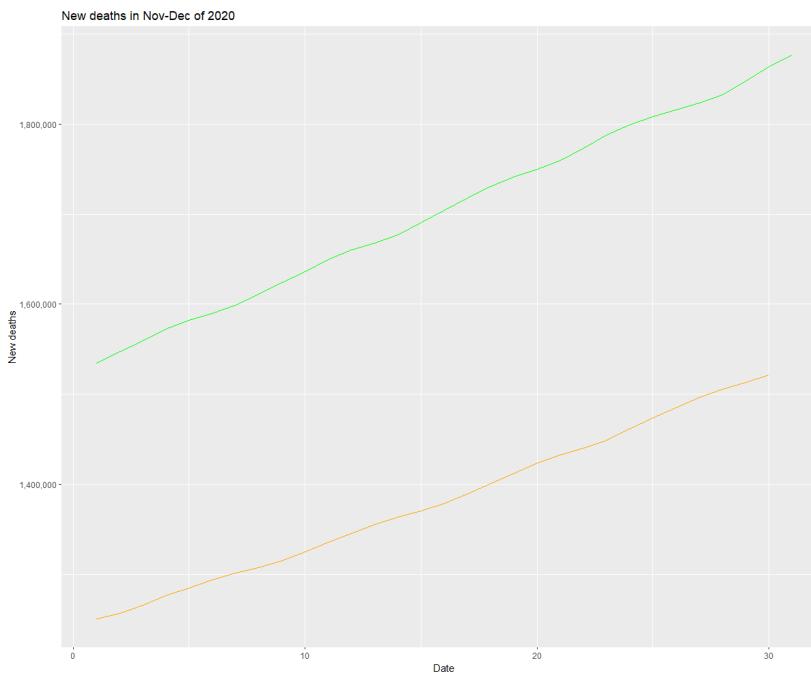
mydata2$new_deaths <- cumsum(mydata2$new_deaths)

thang11 <- mydata2[mydata2$date >= "2021-11-1" & mydata2$date <= "2021-11-30", ]
thang12 <- mydata2[mydata2$date >= "2021-12-1" & mydata2$date <= "2021-12-31", ]

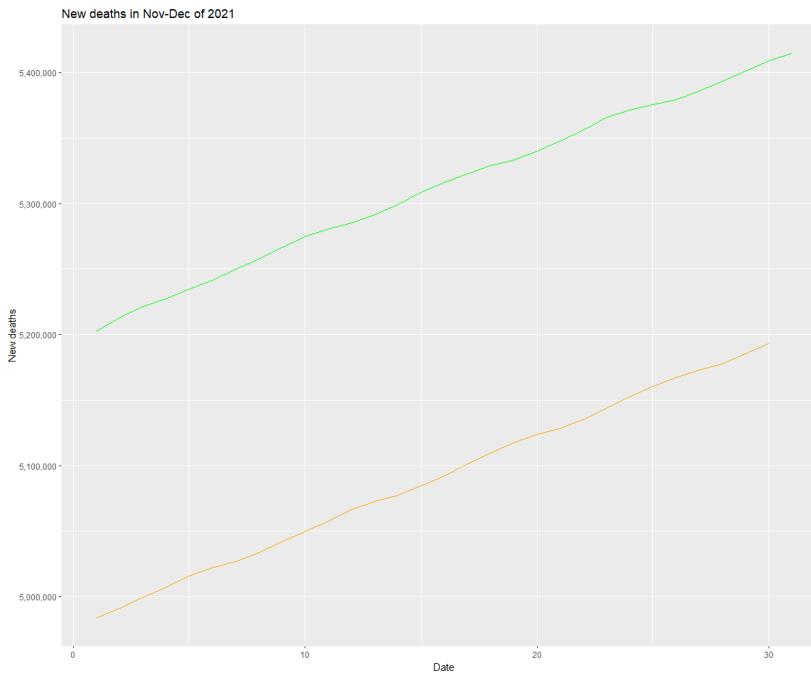
thang11$date <- day(thang11$date)
thang12$date <- day(thang12$date)

ketqua <- ggplot()+
  geom_line(data = thang11, mapping = aes(x = date, y = new_deaths,
                                             color = "November"))+
  geom_line(data = thang12, mapping = aes(x = date, y = new_deaths,
                                             color = "December"))+
  scale_color_manual(name = "Note:", values = c("November" = "orange",
                                                "December" = "green"))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma)
print(res + ggtitle("New deaths in Nov-Dec of 2021") + labs(x = "Date", y = "New deaths"))
```

- Output câu 6 năm 2020:



- Output câu 6 năm 2021:





## viii Nhóm câu hỏi liên quan tất cả quốc gia theo trung bình 7 ngày gần nhất

Trên từng năm vẽ biểu đồ thể hiện trực Ox là thời gian, trực Oy là nhiễm bệnh/tử vong. Dùng 4 ký số của mã đề để vẽ 4 tháng tương ứng theo ký số đó. Nếu ký số là 0 thì lấy tháng là 10.

- 1) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo thời gian là tháng của tất cả quốc gia theo trung bình 7 ngày gần nhất

- Phương pháp làm: sẽ bao gồm những ý chính sau:

- Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file owid-covid-data.csv rồi lưu vào một data frame, sau đó lấy ra những hàng ở cột location có tên là "World". Tiếp theo lấy giá trị tuyệt đối cho những giá trị new\_cases và new\_deaths (để chuyển những giá trị âm thành dương).
- Bước 2: Ta tính toán dữ liệu new\_cases theo trung bình 7 ngày gần nhất bằng vòng lặp for. Khi sử dụng vòng lặp ta tính theo 2 trường hợp:
  - Trường hợp 1: Tại thời điểm xét số ngày trước đó ít hơn 7 ngày thì sẽ được tính như sau:
$$\text{day1} = \text{day1}/1$$
$$\text{day2} = (\text{day1} + \text{day2})/2$$
$$\dots$$
$$\text{day7} = (\text{day1} + \text{day2} + \dots + \text{day7})/7$$
  - Trường hợp 2: từ ngày thứ 8 trong data frame trở đi ta tính như sau:
$$\text{day}(n) = (\text{day}(n-6) + \text{day}(n-5) + \dots + \text{day}(n))/7$$
Dữ liệu trên được lưu vào vector result, rồi sau đó gắn lại vào data frame ở mục new\_cases.
- Bước 3: Sau đó ta lọc dữ liệu theo các tháng 1, 2, 4, 10 của các năm 2020, 2021, 2022 rồi lưu vào các data frame nhỏ hơn.
- Bước 4: Phần vẽ đồ thị:(4 đường riêng biệt thể hiện theo 4 tháng): Ta sử dụng hàm ggplot từ thư viện ggplot2 để vẽ đồ thị theo từng data frame nhỏ đã lọc ra.

- Code mà nhóm thực hiện: (tương tự lọc dữ liệu theo thời gian cho năm 2021 và 2022)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

result <- vector(length = 760)
for(i in 1: 7){
  result[i] <- mean(mydata2$new_cases[1:i])
}
for (i in 8: 760){
  result[i] <- mean(mydata2$new_cases[(i-6):i])
}
mydata2$new_cases <- result

thang1 <- mydata2[mydata2$date >= "2020-1-1" & mydata2$date <= "2020-1-31",
  ]
thang2 <- mydata2[mydata2$date >= "2020-2-1" & mydata2$date <= "2020-2-29",
  ]
thang4 <- mydata2[mydata2$date >= "2020-4-1" & mydata2$date <= "2020-4-30",
  ]
thang10 <- mydata2[mydata2$date >= "2020-10-1" & mydata2$date <= "2020-10-31",
  ]

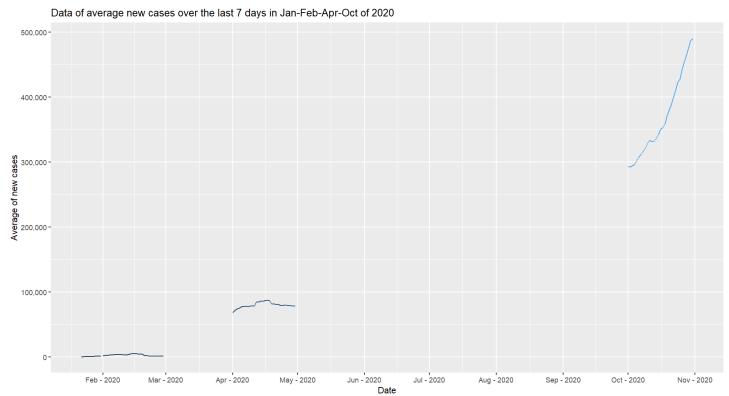
ketqua <- ggplot()+
  geom_line(data = thang1, mapping = aes(x = date, y = new_cases, color =
  date))+
```

```

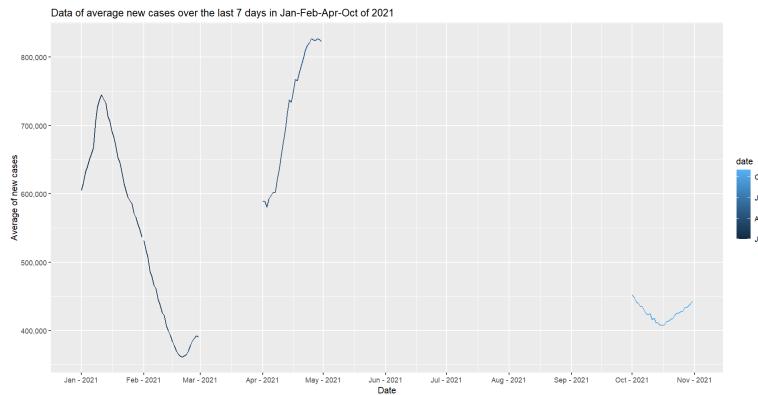
geom_line(data = thang2, mapping = aes(x = date, y = new_cases, color
= date))+
geom_line(data = thang4, mapping = aes(x = date, y = new_cases, color
= date))+
geom_line(data = thang10, mapping = aes(x = date, y = new_cases,
color = date))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma) + scale_x_date(date_
breaks = "1 month", date_labels = "%b - %Y")
print(res + ggtitle("Data of average new cases over the last 7 days in
Jan-Feb-Apr-Oct of 2020") + labs(x = "Date", y = "Average of new cases
"))

```

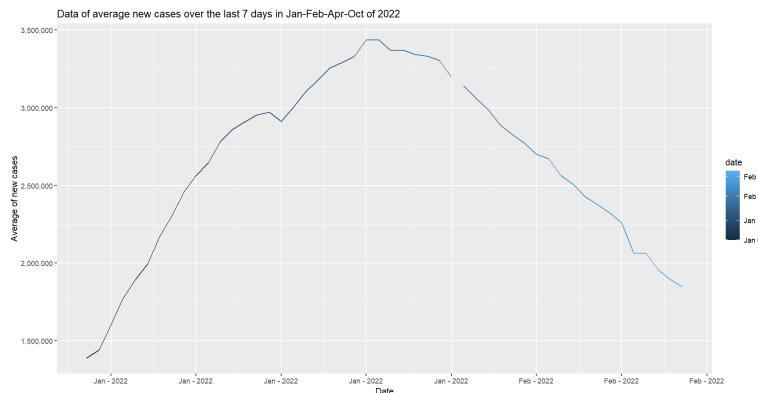
- Output câu 1 năm 2020:



- Output câu 1 năm 2021:



- Output câu 1 năm 2022:





2) Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là tháng của tất cả quốc gia theo trung bình 7 ngày gần nhất

- Phương pháp làm: tương tự như câu 1) phía trên nhưng là thay vì tính dữ liệu của cột new\_cases thì ta sẽ tính dữ liệu ở cột new\_deaths và sau đó cũng vẽ đồ thị.

- Code mà nhóm thực hiện: (tương tự lọc dữ liệu theo thời gian cho năm 2021 và 2022)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

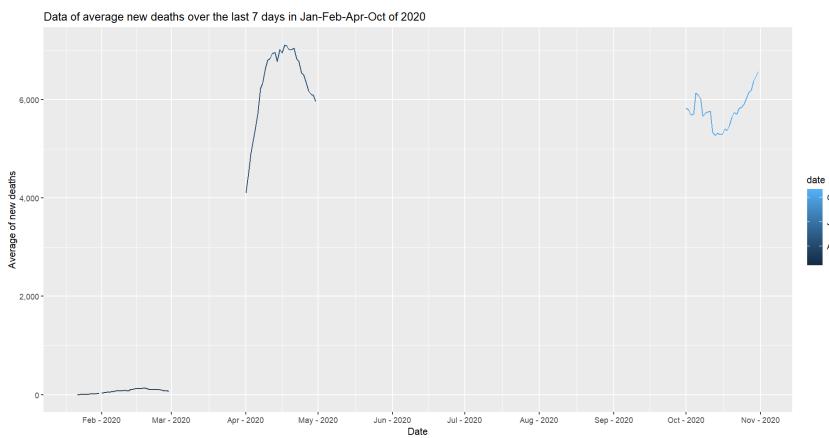
mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

result <- vector(length = 760)
for(i in 1: 7){
  result[i] <- mean(mydata2$new_deaths[1:i])
}
for (i in 8: 760){
  result[i] <- mean(mydata2$new_deaths[(i-6):i])
}
mydata2$new_deaths <- result

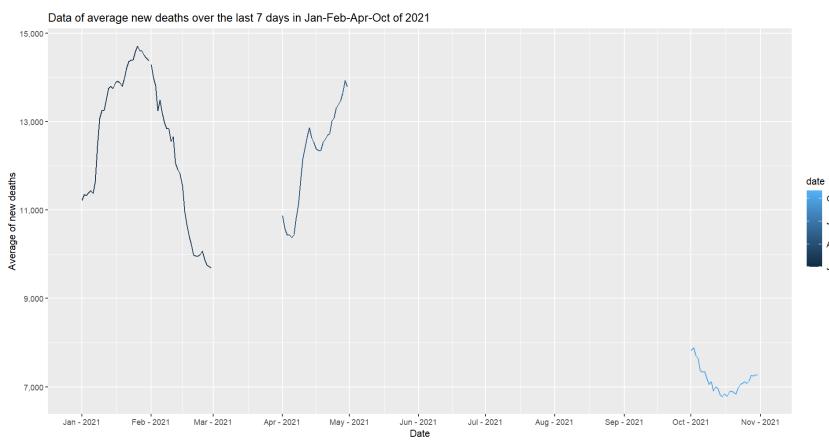
thang1 <- mydata2[mydata2$date >= "2020-1-1" & mydata2$date <= "2020-1-31",
  ]
thang2 <- mydata2[mydata2$date >= "2020-2-1" & mydata2$date <= "2020-2-29",
  ]
thang4 <- mydata2[mydata2$date >= "2020-4-1" & mydata2$date <= "2020-4-30",
  ]
thang10 <- mydata2[mydata2$date >= "2020-10-1" & mydata2$date <= "2020-10-31",
  ]

ketqua <- ggplot()+
  geom_line(data = thang1, mapping = aes(x = date, y = new_deaths,
    color = date))+
  geom_line(data = thang2, mapping = aes(x = date, y = new_deaths,
    color = date))+
  geom_line(data = thang4, mapping = aes(x = date, y = new_deaths,
    color = date))+
  geom_line(data = thang10, mapping = aes(x = date, y = new_deaths,
    color = date))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma) + scale_x_date(date_
  breaks = "1 month", date_labels = "%b - %Y")
print(res + ggtitle("Data of average new deaths over the last 7 days in
  Jan-Feb-Apr-Oct of 2020"))
+ labs(x = "Date", y = "Average of new deaths"))
```

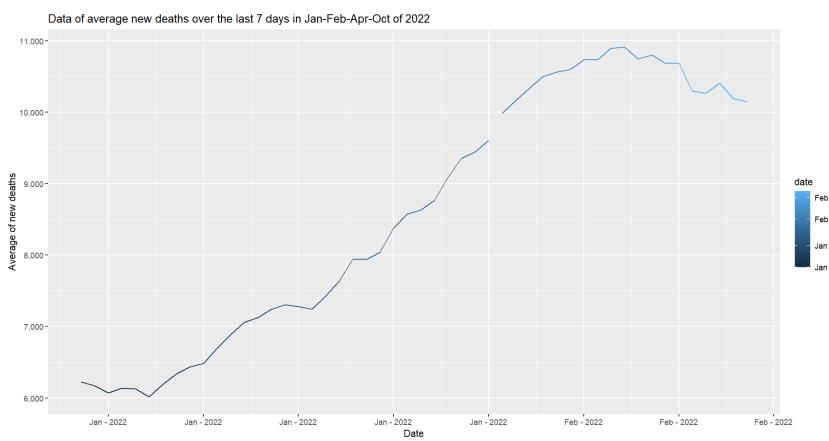
- Output câu 2 năm 2020:



- Output câu 2 năm 2021:



- Output câu 2 năm 2022:





- 3) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo thời gian là 2 tháng của năm của tất cả quốc gia/ theo trung bình 7 ngày gần nhất

- Phương pháp làm: sẽ bao gồm những ý chính sau:

- a) Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file gốc lấy ra những cột location có tên là "World". Sau đó lấy giá trị tuyệt đối cho những giá trị `new_cases` và `new_deaths` (để chuyển những giá trị âm thành dương).
- b) Bước 2: Ta tính toán dữ liệu `new_cases` theo trung bình 7 ngày gần nhất bằng vòng lặp for. Khi sử dụng vòng lặp ta tính theo 2 trường hợp:

- Trường hợp 1: Tại thời điểm xét số ngày trước đó ít hơn 7 ngày thì sẽ được tính như sau:

$$\begin{aligned} \text{day1} &= \text{day1}/1 \\ \text{day2} &= (\text{day1} + \text{day2})/2 \end{aligned}$$

...

$$\text{day7} = (\text{day1} + \text{day2} + \dots + \text{day7})/7$$

- Trường hợp 2: từ ngày thứ 8 trong data frame trở đi ta tính như sau:

$$\text{day}(n) = (\text{day}(n-6) + \text{day}(n-5) + \dots + \text{day}(n))/7$$

Dữ liệu trên được lưu vào vector result, rồi sau đó gắn lại vào data frame ở mục `new_cases`.

- c) Bước 3: Sau đó ta lọc dữ liệu 2 tháng 11 và 12 của các năm 2020, 2021, 2022 rồi lưu vào các data frame nhỏ hơn.
- d) Bước 4: Phần vẽ đồ thị: Ta sử dụng hàm `ggplot` từ thư viện `ggplot2` để vẽ đồ thị theo từng data frame nhỏ đã lọc ra.

- Code mà nhóm thực hiện: (tương tự lọc dữ liệu theo thời gian cho năm 2021, vì dữ liệu năm 2022 không có 2 tháng 11 và 12)

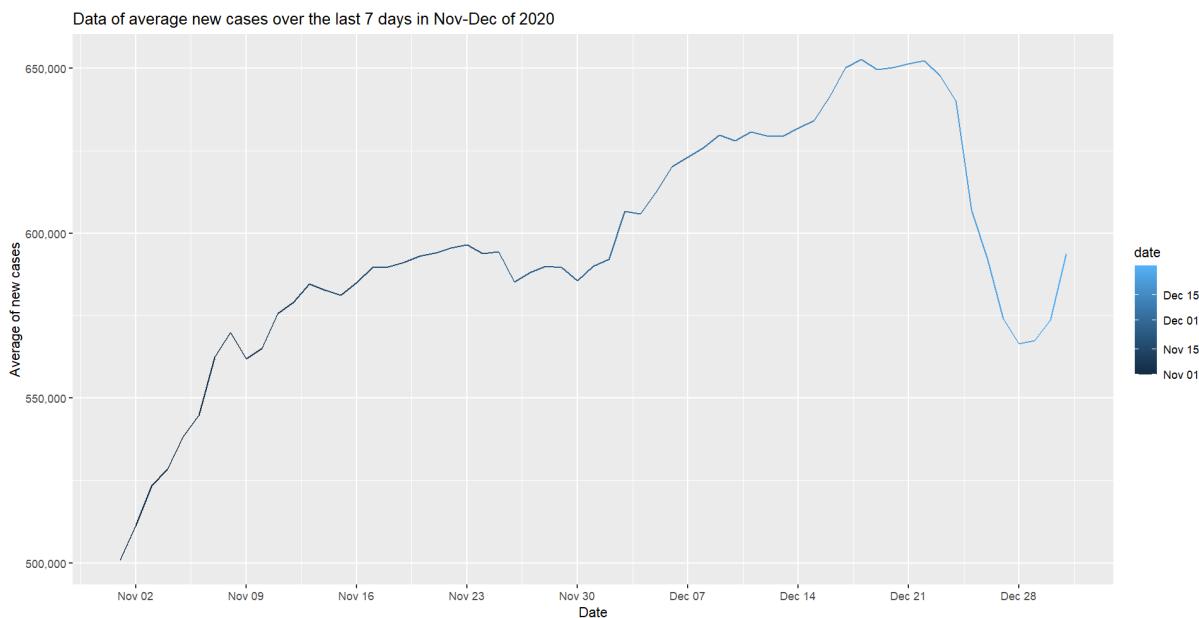
```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

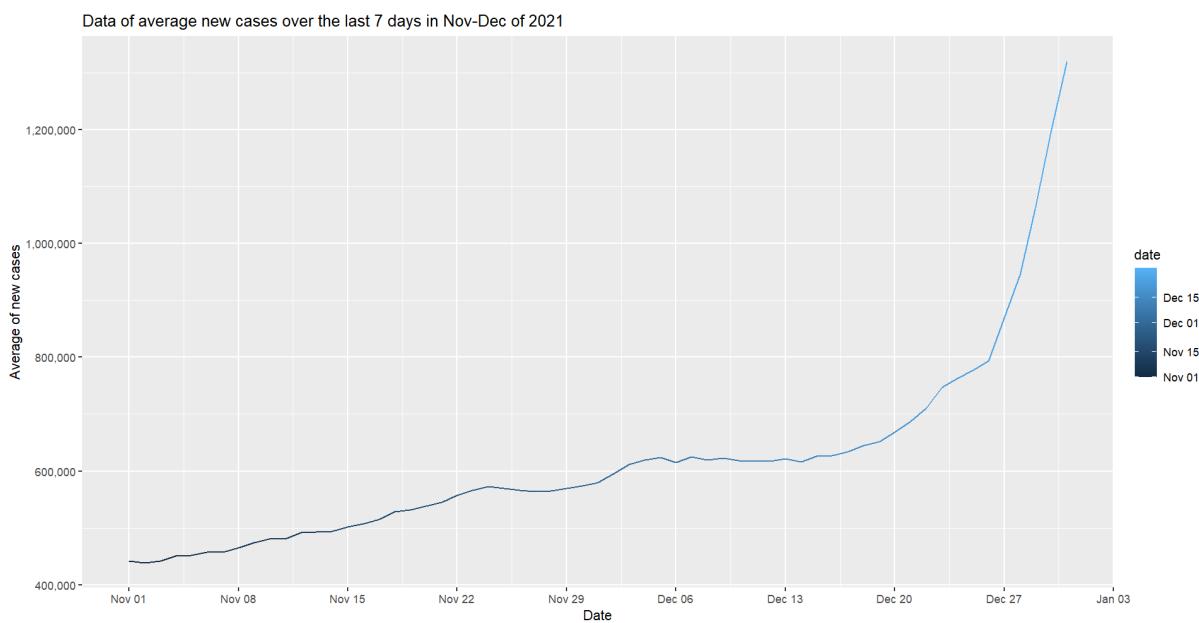
result <- vector(length = 760)
for(i in 1: 7){
  result[i] <- mean(mydata2$new_cases[1:i])
}
for (i in 8: 760){
  result[i] <- mean(mydata2$new_cases[(i-6):i])
}
mydata2$new_cases <- result
mydata2 <- mydata2[mydata2$date >= "2020-11-1" & mydata2$date <= "2020-12-31", ]

ketqua <- ggplot() + geom_line(data = mydata2, mapping = aes(x = date,
  = new_cases, color = date))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma) + scale_x_date(date_
  breaks = "1 week", date_labels = "%b %d")
print(res + ggtitle("Data of average new cases over the last 7 days in
  Nov-Dec of 2020") + labs(x = "Date", y = "Average of new cases"))
```

- Output câu 3 năm 2020:



- Output câu 3 năm 2021:





- 4) Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là 2 tháng của năm của tất cả quốc gia theo trung bình 7 ngày gần nhất

- Phương pháp làm: tương tự như câu 3) phía trên nhưng là thay vì tính dữ liệu của cột new\_cases thì ta sẽ tính dữ liệu ở cột new\_deaths và sau đó cũng vẽ đồ thị.

- Code mà nhóm thực hiện: (tương tự lọc dữ liệu theo thời gian cho năm 2021, vì dữ liệu năm 2022 không có 2 tháng 11 và 12)

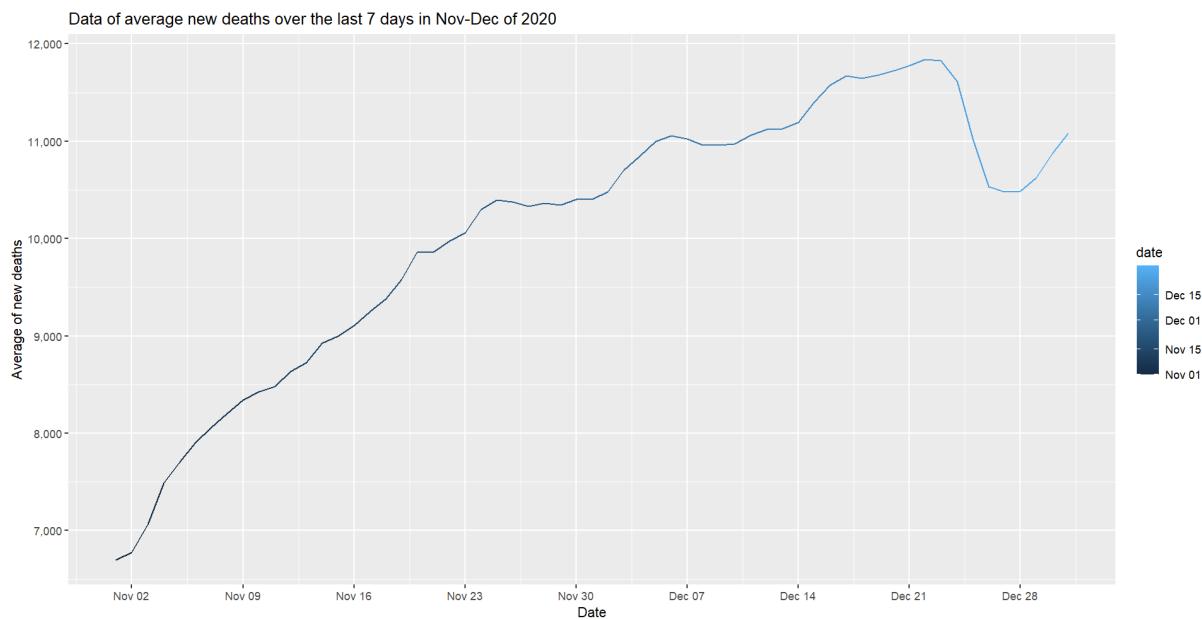
```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

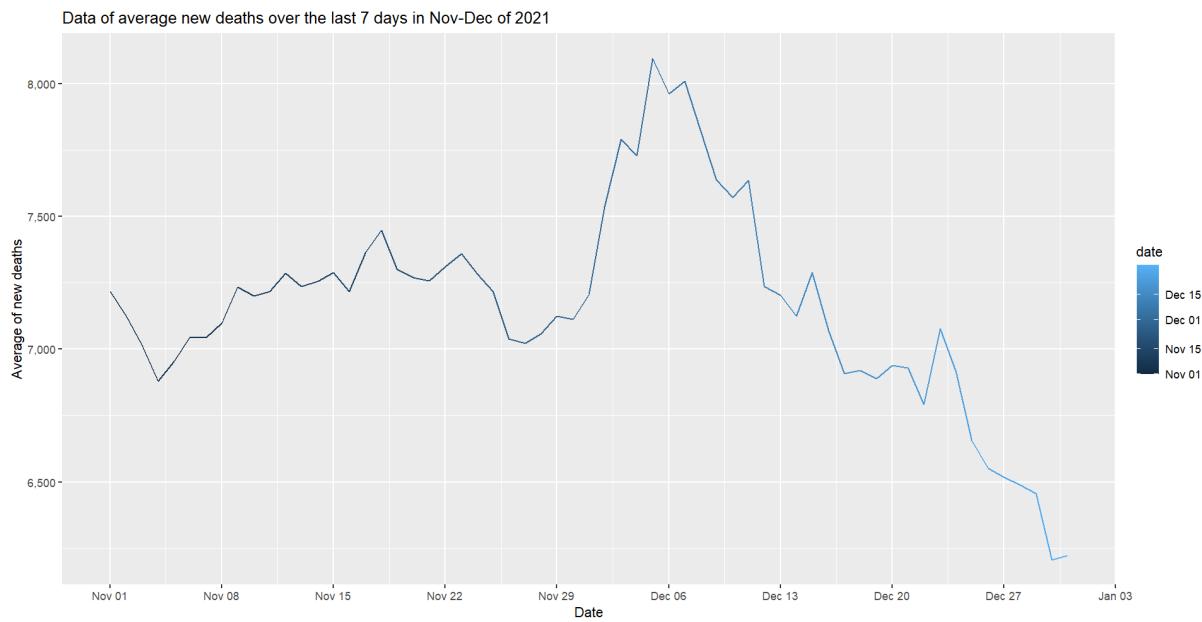
result <- vector(length = 760)
for(i in 1: 7){
  result[i] <- mean(mydata2$new_deaths[1:i])
}
for (i in 8: 760){
  result[i] <- mean(mydata2$new_deaths[(i-6):i])
}
mydata2$new_deaths <- result
mydata2 <- mydata2[mydata2$date >= "2020-11-1" & mydata2$date <= "2020-12-31", ]

ketqua <- ggplot() + geom_line(data = mydata2, mapping = aes(x = date,
  = new_deaths, color = date))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma) + scale_x_date(date_
  breaks = "1 week", date_labels = "%b %d")
print(res + ggtitle("Data of average new deaths over the last 7 days in
  Nov-Dec of 2020") + labs(x = "Date", y = "Average of new deaths"))
```

- Output câu 4 năm 2020:



- Output câu 4 năm 2021:





- 5) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy theo thời gian là 2 tháng của năm của tất cả quốc gia/ theo trung bình 7 ngày gần nhất

- Phương pháp làm: sẽ bao gồm những ý chính sau:

- a) Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file gốc lấy ra những cột location có tên là "World". Sau đó lấy giá trị tuyệt đối cho những giá trị `new_cases` và `new_deaths` (để chuyển những giá trị âm thành dương).
- b) Bước 2: Ta tính toán dữ liệu `new_cases` theo trung bình 7 ngày gần nhất bằng vòng lặp for. Khi sử dụng vòng lặp ta tính theo 2 trường hợp:
- Trường hợp 1: Tại thời điểm xét số ngày trước đó ít hơn 7 ngày thì sẽ được tính như sau:  
$$\text{day1} = \text{day1}/1$$
  
$$\text{day2} = (\text{day1} + \text{day2})/2$$
  
$$\dots$$
  
$$\text{day7} = (\text{day1} + \text{day2} + \dots + \text{day7})/7$$
  - Trường hợp 2: từ ngày thứ 8 trong data frame trở đi ta tính như sau:  
$$\text{day}(n) = (\text{day}(n-6) + \text{day}(n-5) + \dots + \text{day}(n))/7$$
- Dữ liệu trên được lưu vào vector `result1`, rồi sau đó gắn lại vào data frame ở mục `new_cases`.
- c) Bước 3: Sau đó ta sử dụng hàm `cumsum` để tính tổng tích lũy của cột `new_cases` lưu vào vector `result2`, rồi sau đó gắn lại vào data frame ở mục `new_cases`.
- d) Bước 4: Sau đó ta lọc dữ liệu 2 tháng 11 và 12 của các năm 2020, 2021, 2022 rồi lưu vào các data frame nhỏ hơn.
- e) Bước 5: Phần vẽ đồ thị: Ta sử dụng hàm `ggplot` từ thư viện `ggplot2` để vẽ đồ thị theo từng data frame nhỏ đã lọc ra.

- Code mà nhóm thực hiện: (tương tự lọc dữ liệu theo thời gian cho năm 2021, vì dữ liệu năm 2022 không có 2 tháng 11 và 12)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

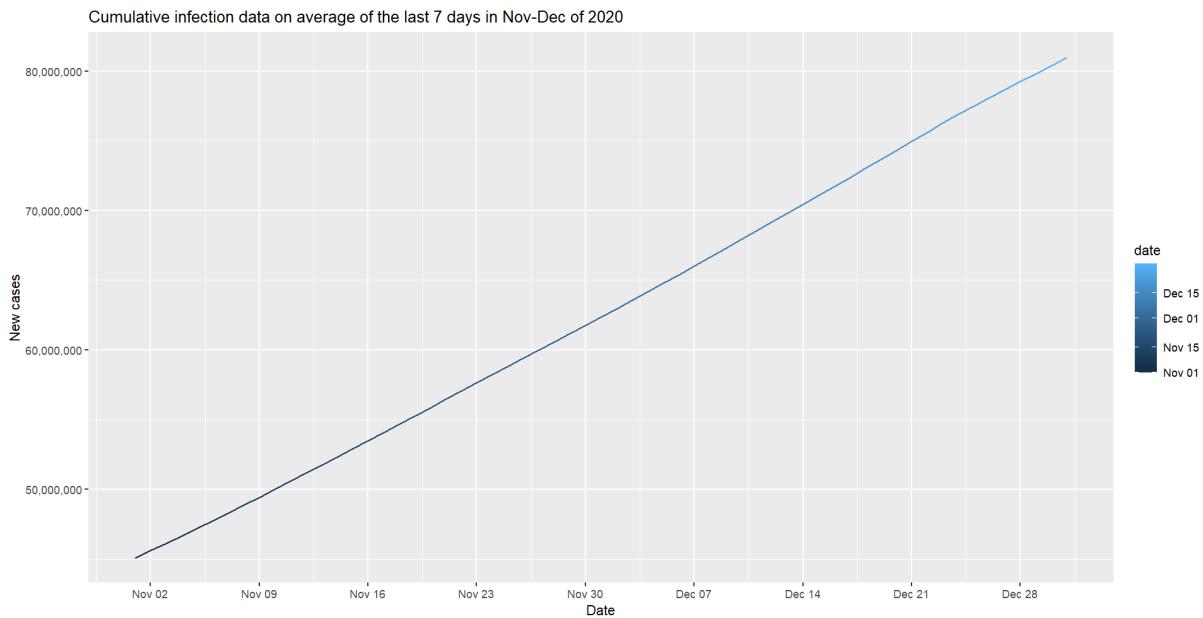
result1 <- vector(length = 760)
for(i in 1: 7){
  result1[i] <- mean(mydata2$new_cases[1:i])
}
for (i in 8: 760){
  result1[i] <- mean(mydata2$new_cases[(i-6):i])
}
mydata2$new_cases <- result1

result2 <- cumsum(mydata2$new_cases)
mydata2$new_cases <- result2

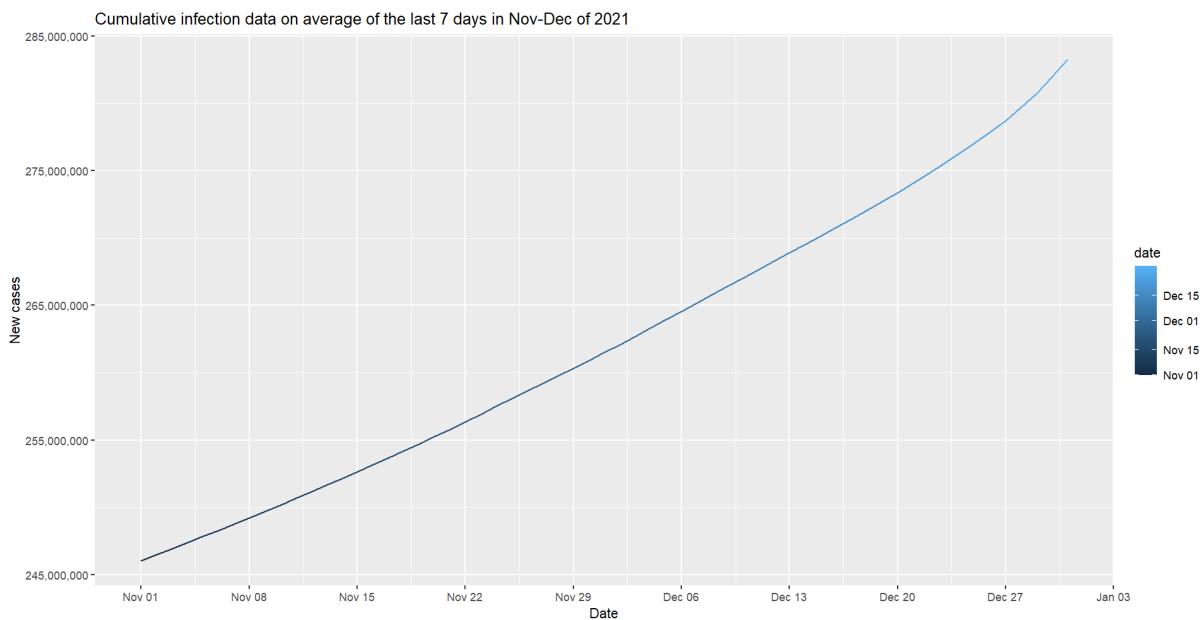
mydata2 <- mydata2[mydata2$date >= "2020-11-1" & mydata2$date <= "2020-12-31", ]

ketqua <- ggplot() + geom_line(data = mydata2, mapping = aes(x = date,
  = new_cases, color = date))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma) + scale_x_date(date =
  breaks = "1 week", date_labels = "%b %d")
print(res + ggtitle("Cumulative infection data on average of the last 7
  days in Nov-Dec of 2020") + labs(x = "Date", y = "New cases"))
```

- Output câu 5 năm 2020:



- Output câu 5 năm 2021:





- 6) Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy theo thời gian là 2 tháng của năm của tất cả quốc gia theo trung bình 7 ngày gần nhất

- Phương pháp làm: tương tự như câu 5) phía trên nhưng là thay vì tính dữ liệu của cột new\_cases thì ta sẽ tính dữ liệu ở cột new\_deaths và sau đó cũng vẽ đồ thị.

- Code mà nhóm thực hiện: (tương tự lọc dữ liệu theo thời gian cho năm 2021, vì dữ liệu năm 2022 không có 2 tháng 11 và 12)

```
mydata1 <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata1, file = 'mydata1.rda')
mydata2 <- mydata1[mydata1$location == "World",]
attach(mydata2)

mydata2 %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
mydata2$date <- as.Date(mydata2$date, "%m/%d/%Y")

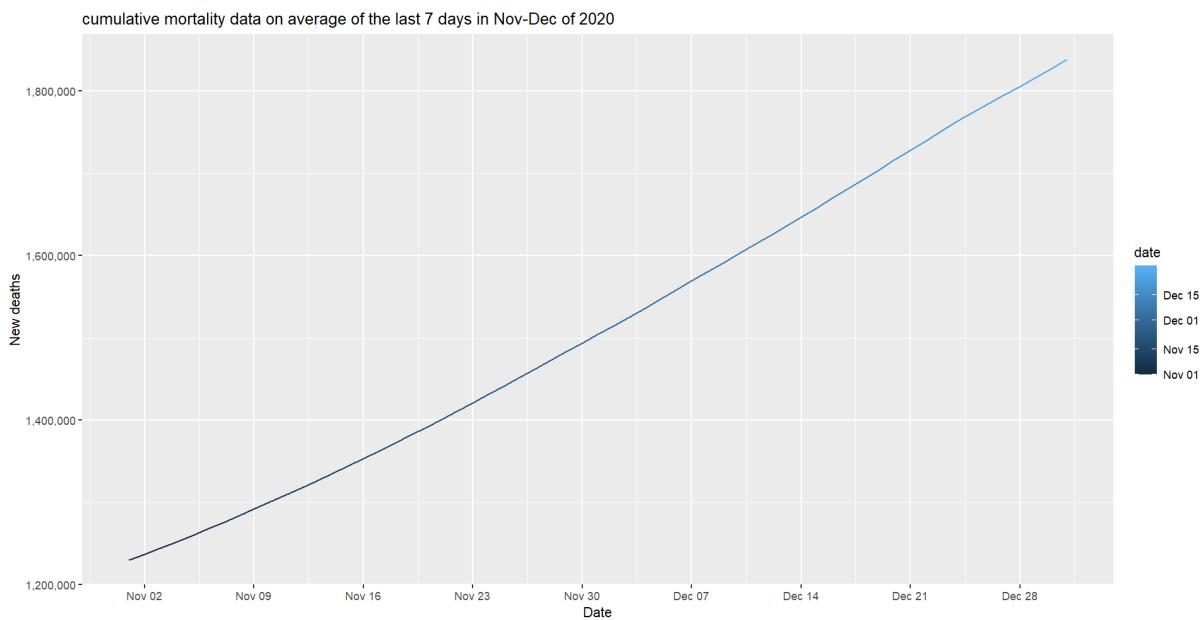
result1 <- vector(length = 760)
for(i in 1: 7){
  result1[i] <- mean(mydata2$new_deaths[1:i])
}
for (i in 8: 760){
  result1[i] <- mean(mydata2$new_deaths[(i-6):i])
}
mydata2$new_deaths <- result1

result2 <- cumsum(mydata2$new_deaths)
mydata2$new_deaths <- result2

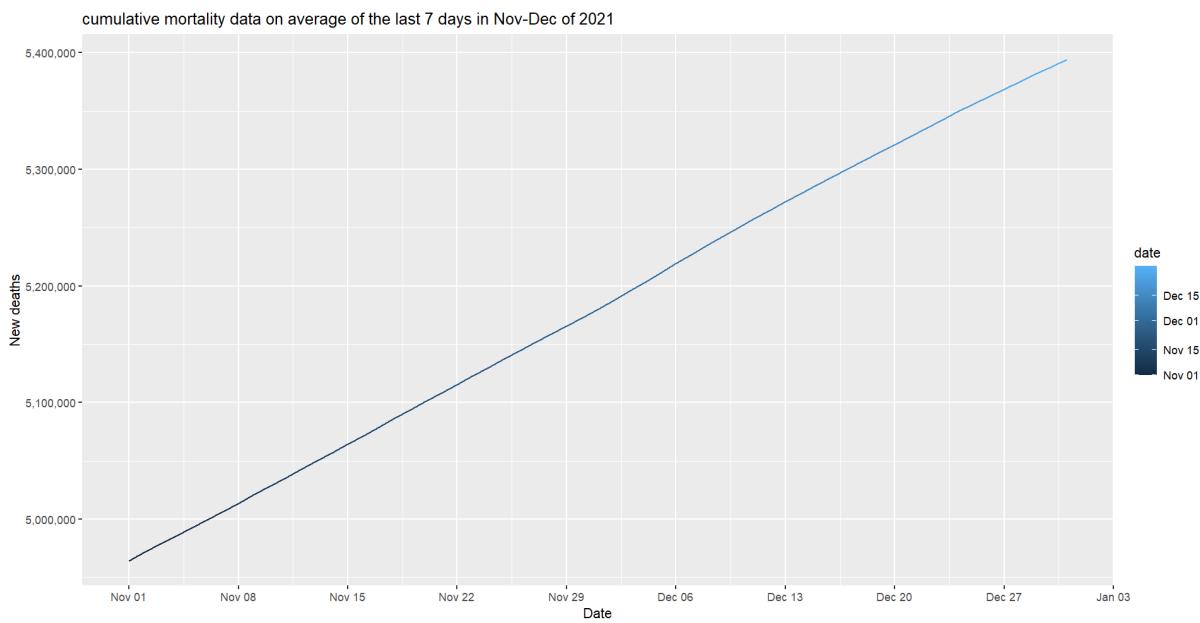
mydata2 <- mydata2[mydata2$date >= "2020-11-1" & mydata2$date <= "2020-12-31",]

ketqua <- ggplot() + geom_line(data = mydata2, mapping = aes(x = date,
  y = new_deaths, color = date))
require(scales)
dm <- ketqua + scale_y_continuous(labels = comma) + scale_x_date(date =
  breaks = "1 week", date_labels = "%b %d")
print(dm + ggtitle("Cumulative mortality data on average of the last 7
  days in Nov-Dec of 2020") + labs(x = "Date", y = "New deaths"))
```

- Output câu 6 năm 2020:



- Output câu 6 năm 2021:





## ix Nhóm câu hỏi liên quan đến sự tương quan giữa nhiễm bệnh và tử vong

1) Vẽ biểu đồ thể hiện phần trăm giữa nhiễm bệnh tích lũy trên tổng nhiễm bệnh và phần trăm tử vong tích lũy trên tổng số tử vong cho từng quốc gia theo thời gian. Vẽ 2 đường trên cùng biểu đồ.

Trên từng quốc gia riêng của nhóm hãy vẽ biểu đồ thể hiện trực Ox là nhiễm bệnh, trực Oy là tử vong. Hãy lấy 4 tháng theo 4 ký số mã để thể hiện. Nếu ký số là 0 thì lấy tháng là 10.

- Phương pháp làm: sẽ bao gồm những ý chính sau:

- Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file gốc lấy ra những cột location có tên là "Indonesia", "Japan", "Vietnam". Sau đó lấy giá trị tuyệt đối cho những giá trị new\_cases và new\_deaths (để chuyển những giá trị âm thành dương).
- Bước 2: Ta lọc bỏ đi những dữ liệu new\_cases và new\_deaths không được thống kê (NA).
- Bước 3: Sau đó ta sử dụng hàm sum để tính tổng số ca nhiễm và tổng số ca tử vong theo các cột new\_cases và new\_deaths.
- Bước 4: Tiếp theo ta sử dụng hàm cumsum để tích tổng tích lũy theo 2 cột new\_cases và new\_deaths.
- Bước 5: Tính phần trăm bằng cách 100\*tổng tích lũy/tổng số ca, sau đó lưu vào lại data frame
- Bước 6: Phần vẽ đồ thị: Ta sử dụng hàm ggplot từ thư viện ggplot2 để vẽ đồ thị theo từng data frame nhỏ đã lọc ra.

- Code mà nhóm thực hiện: (tương tự cho những dữ liệu của Japan và Vietnam)

```
mydata <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata, file = 'mydata.rda')
indo <- mydata[mydata$location == "Indonesia",]
attach(indo)

indo %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
indo$date <- as.Date(indo$date, "%m/%d/%Y")
indonhiem <- indo[indo$date <= "2022-2-18", ]
indotuvong <- indo[indo$date >= "2020-3-11", ]

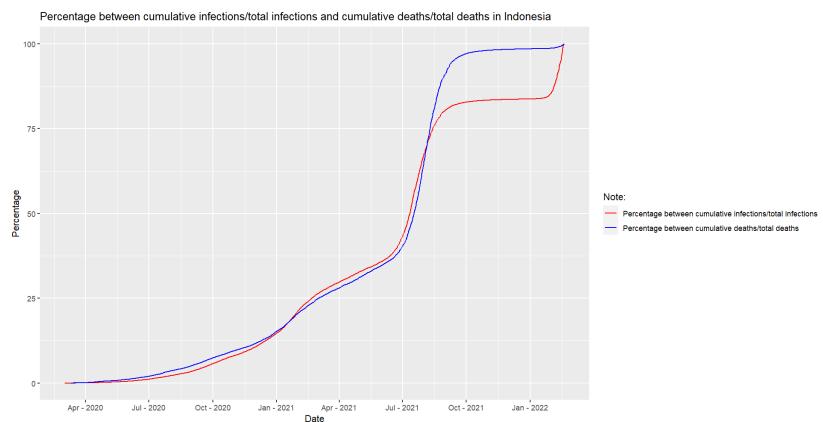
tongcanhiem <- sum(indo$new_cases, na.rm = TRUE)
tongcatuvong <- sum(indo$new_deaths, na.rm = TRUE)

tongnhiemtichluy <- cumsum(indonhiem$new_cases)
tongtuvongtichluy <- cumsum(indotuvong$new_deaths)

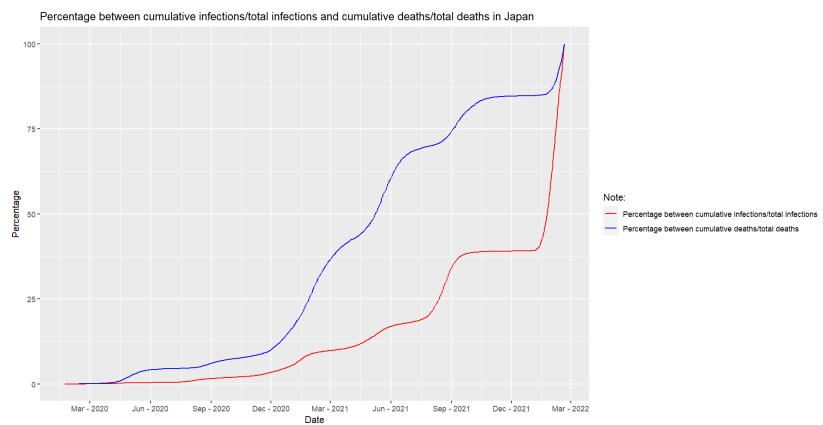
result1 <- 100*tongnhiemtichluy/tongcanhiem
indonhiem$new_cases <- result1
result2 <- 100*tongtuvongtichluy/tongcatuvong
indotuvong$new_deaths <- result2

ketqua <- ggplot()+
  geom_line(data = indonhiem, mapping = aes(x = date, y = new_cases,
    color = "Percentage between cumulative infections/total infections
    "))+ 
  geom_line(data = indotuvong, mapping = aes(x = date, y = new_deaths,
    color = "Percentage between cumulative deaths/total deaths"))+
  scale_color_manual(name = "Note:", values = c("Percentage between
    cumulative infections/total infections" = "red", "Percentage
    between cumulative deaths/total deaths" = "blue"))
require(scales)
res <- ketqua + scale_y_continuous(labels = comma) + scale_x_date(date_
  breaks = "3 month", date_labels = "%b - %Y")
print(res + ggtitle("Percentage between cumulative infections/total
  infections and cumulative deaths/total deaths in Indonesia") + labs(x
  = "Date", y = "Percentage"))
```

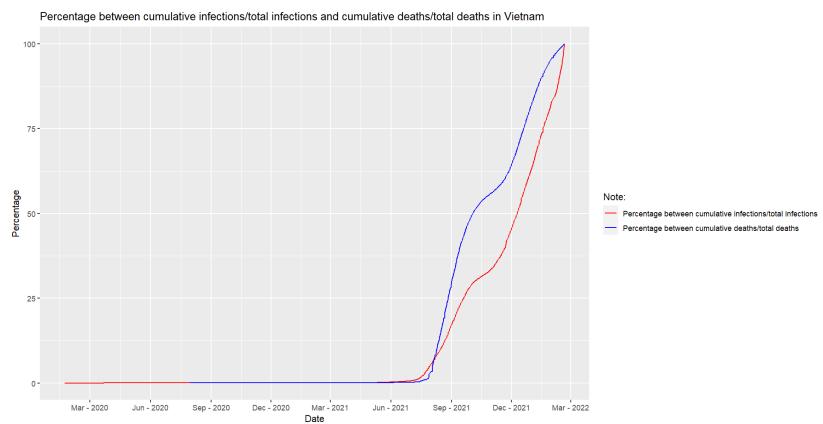
- Output câu 1 của Indonesia:



- Output câu 1 của Japan:



- Output câu 1 của Vietnam:





2) Xét tương quan trong mỗi tháng.

- Mục tiêu cần đạt được: Vẽ biểu đồ tương quan giữa số ca nhiễm bệnh và số ca tử vong theo ngày, tính hệ số tương quan, từ đó rút ra nhận xét về sự tương quan (hướng tương quan, tính tương quan mạnh hay yếu,...).

- Phương pháp làm:

- Bước 1: Đầu tiên ta sẽ lọc dữ liệu từ file gốc lấy ra những cột location có tên là "Indonesia", "Japan", "Vietnam". Sau đó lấy giá trị tuyệt đối cho những giá trị new\_cases và new\_deaths (để chuyển những giá trị âm thành dương).
- Bước 2: Ta thay những dữ liệu new\_cases và new\_deaths không được thống kê (NA) bằng các giá trị là 0.
- Bước 3: Vẽ đồ thị tương quan sử dụng hàm `ggplot` từ thư viện `ggplot2`.
- Bước 4: Tính hệ số tương quan qua hàm `cor()`.
- Bước 5: Rút ra nhận xét về sự tương quan.

- Code mà nhóm thực hiện: (tương tự cho 2 nước còn lại là Japan và Vietnam)

```
mydata <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata, file = 'mydata.rda')
indo <- mydata[mydata$location == "Indonesia", ]
attach(indo)

indo$new_cases[is.na(indo$new_cases)] <- 0
indo$new_deaths[is.na(indo$new_deaths)] <- 0

indo %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
indo$date <- as.Date(indo$date, "%m/%d/%Y")

indo$year <- strftime(indo$date, "%Y")
indo$month <- strftime(indo$date, "%m")
indo$dates<- strftime(indo$date, "%d")

indo_2020 <- subset(indo, (year == "2020"))
indo_2021 <- subset(indo, (year == "2021"))
indo_2022 <- subset(indo, (year == "2022"))

indo_2020_4 <- subset(indo_2020, (month == "04"))
indo_2020_10 <- subset(indo_2020, (month == "10"))

indo_2021_1 <- subset(indo_2021,(month=="01"))
indo_2021_2 <- subset(indo_2021,(month=="02"))
indo_2021_4 <- subset(indo_2021,(month=="04"))
indo_2021_10 <- subset(indo_2021,(month=="10"))

indo_2022_1 <- subset(indo_2022,(month=="01"))
indo_2022_2 <- subset(indo_2022,(month=="02"))

ggplot(indo_2020_4,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 4/2020 cua Indonesia") + geom_smooth()
print(cor(indo_2020_4$new_cases,indo_2020_4$new_deaths))

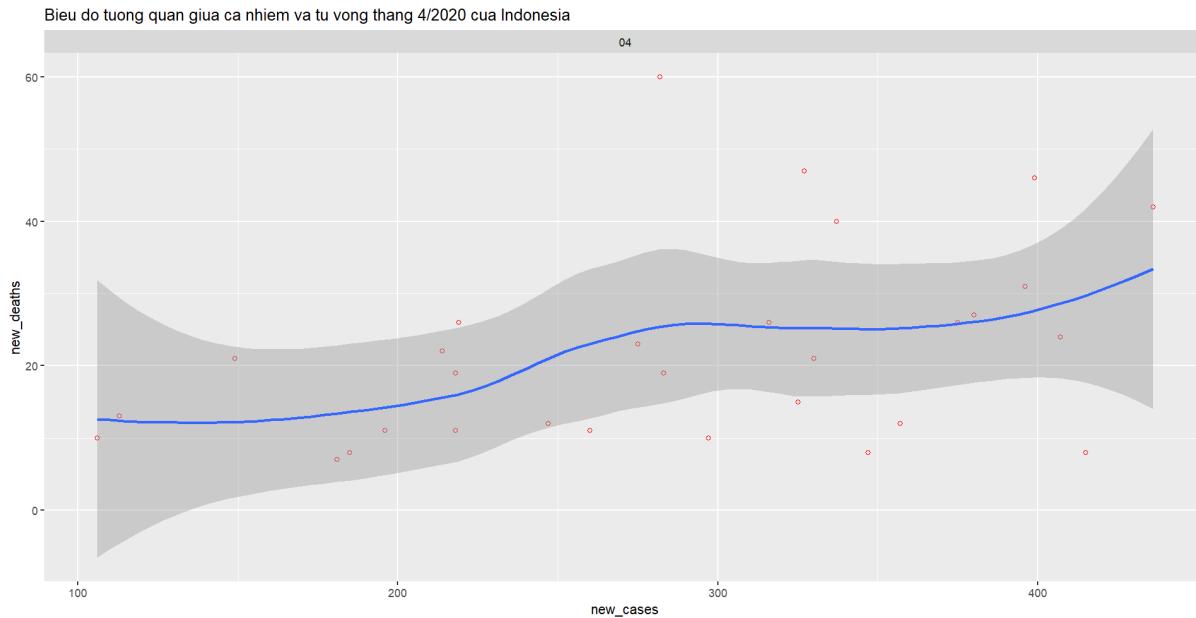
ggplot(indo_2020_10,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 10/2020 cua Indonesia") + geom_smooth()
print(cor(indo_2020_10$new_cases,indo_2020_10$new_deaths))
```



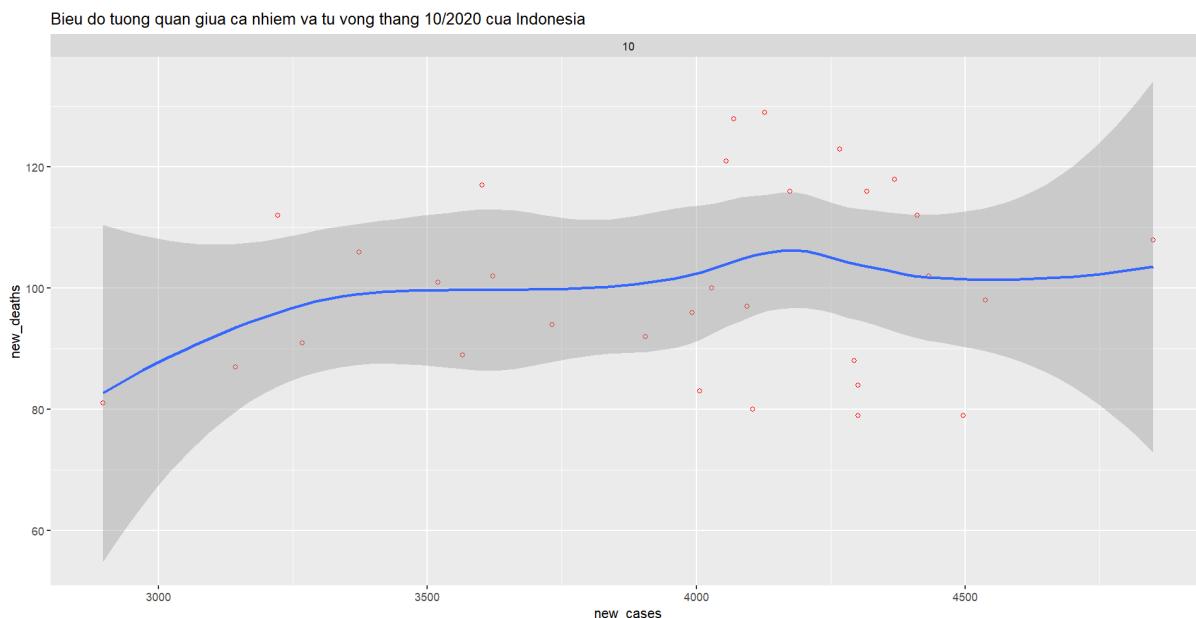
```
ggplot(indo_2021_1,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 1/2021 cua Indonesia") + geom_smooth()  
print(cor(indo_2021_1$new_cases,indo_2021_1$new_deaths))  
  
ggplot(indo_2021_2,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 2/2021 cua Indonesia") + geom_smooth()  
print(cor(indo_2021_2$new_cases,indo_2021_2$new_deaths))  
  
ggplot(indo_2021_4,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 4/2021 cua Indonesia") + geom_smooth()  
print(cor(indo_2021_4$new_cases,indo_2021_4$new_deaths))  
  
ggplot(indo_2021_10,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 10/2021 cua Indonesia") + geom_smooth()  
print(cor(indo_2021_10$new_cases,indo_2021_10$new_deaths))  
  
ggplot(indo_2022_1,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 1/2022 cua Indonesia") + geom_smooth()  
print(cor(indo_2022_1$new_cases,indo_2022_1$new_deaths))  
  
ggplot(indo_2022_2,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color="red")+facet_wrap(~ month)+ ggtitle("Bieu do tuong quan giua ca nham va tu vong thang 2/2022 cua Indonesia") + geom_smooth()  
print(cor(indo_2022_2$new_cases,indo_2022_2$new_deaths))
```

- Output câu 2 của Indonesia:

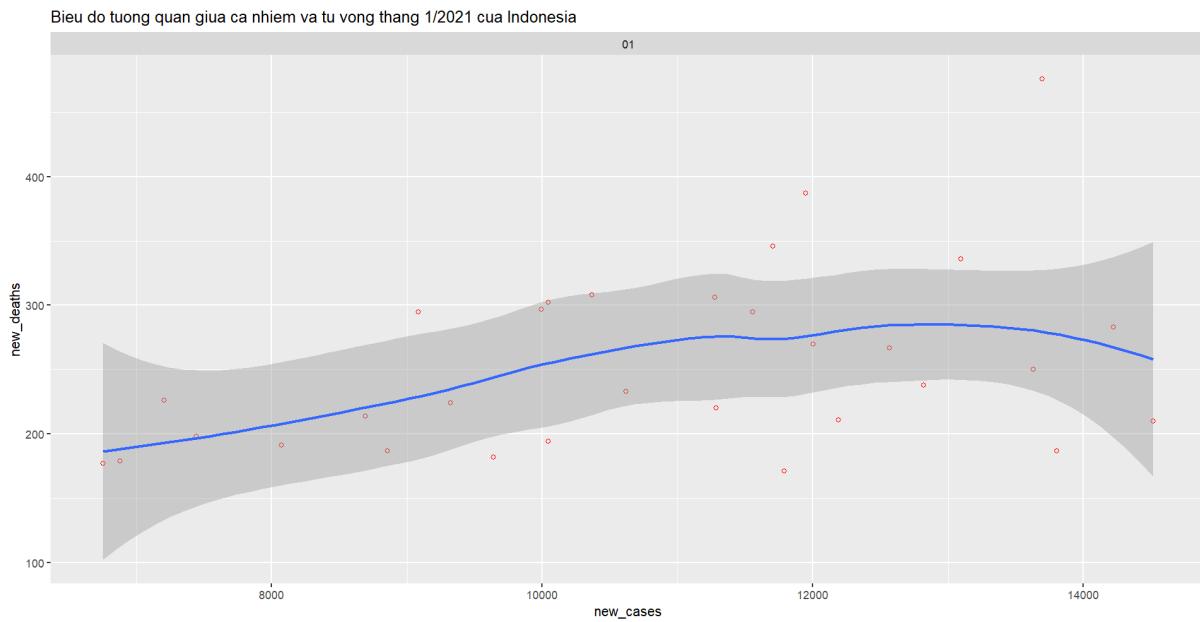
Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 4-2020 của Indonesia là 0.416476, hướng tương quan dương và có sự tương quan tuyến tính **yếu** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



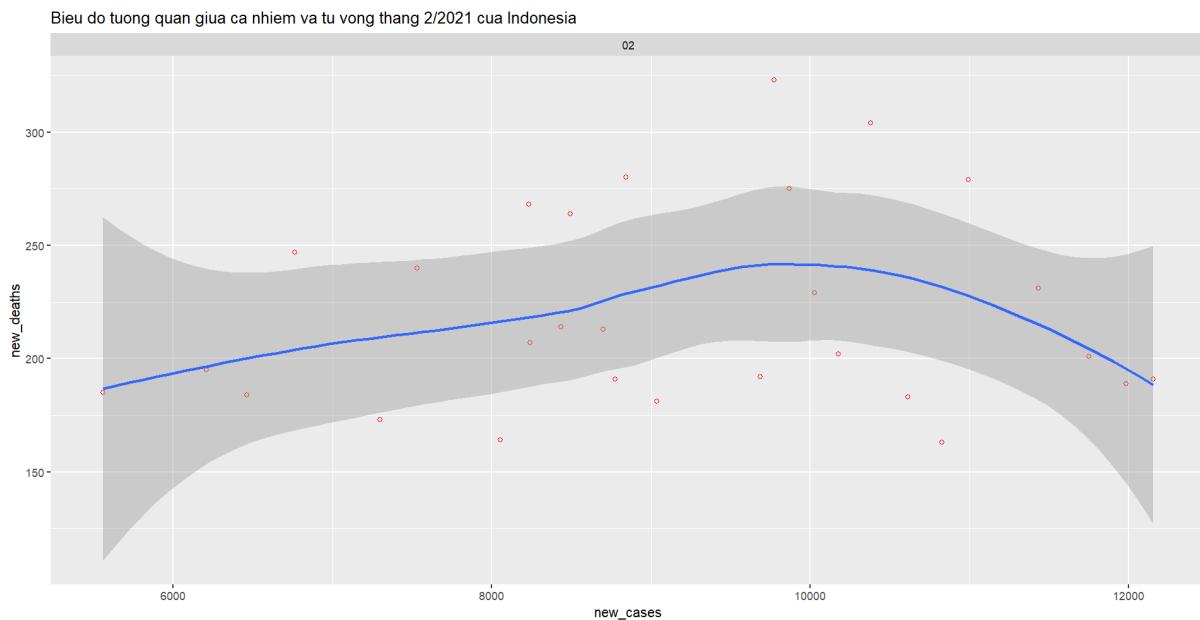
Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 10-2020 của Indonesia là 0.1741182, hướng tương quan dương và có sự tương quan tuyến tính **rất yếu** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



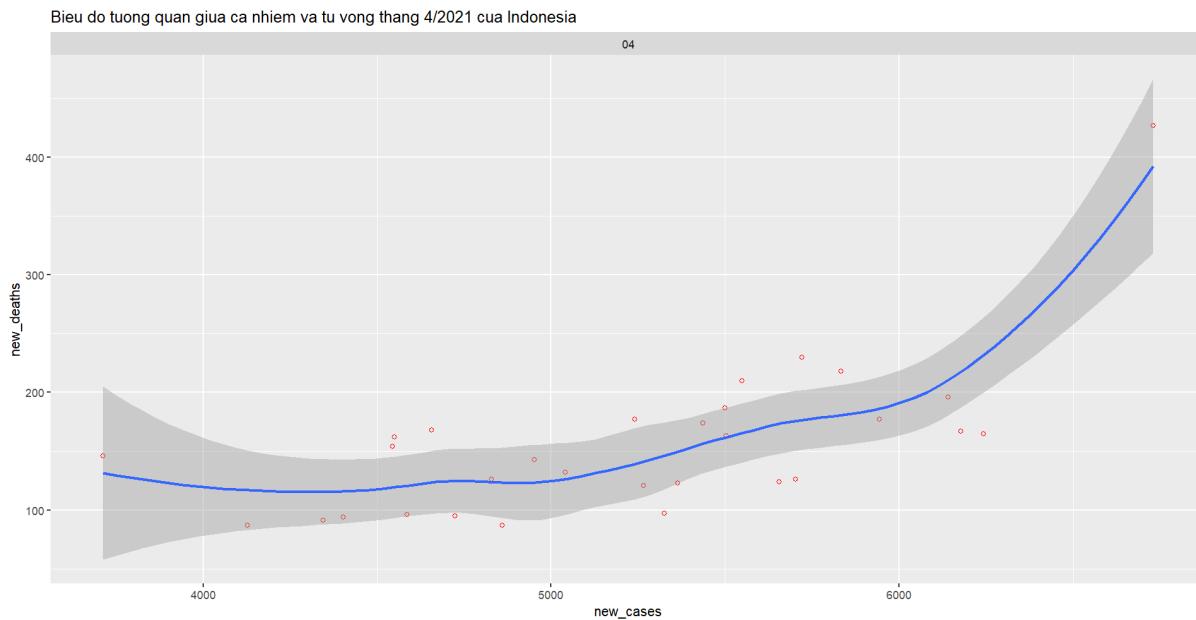
Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 1-2021 của Indonesia là 0.419901, hướng tương quan dương và có sự tương quan tuyến tính **ý êu** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



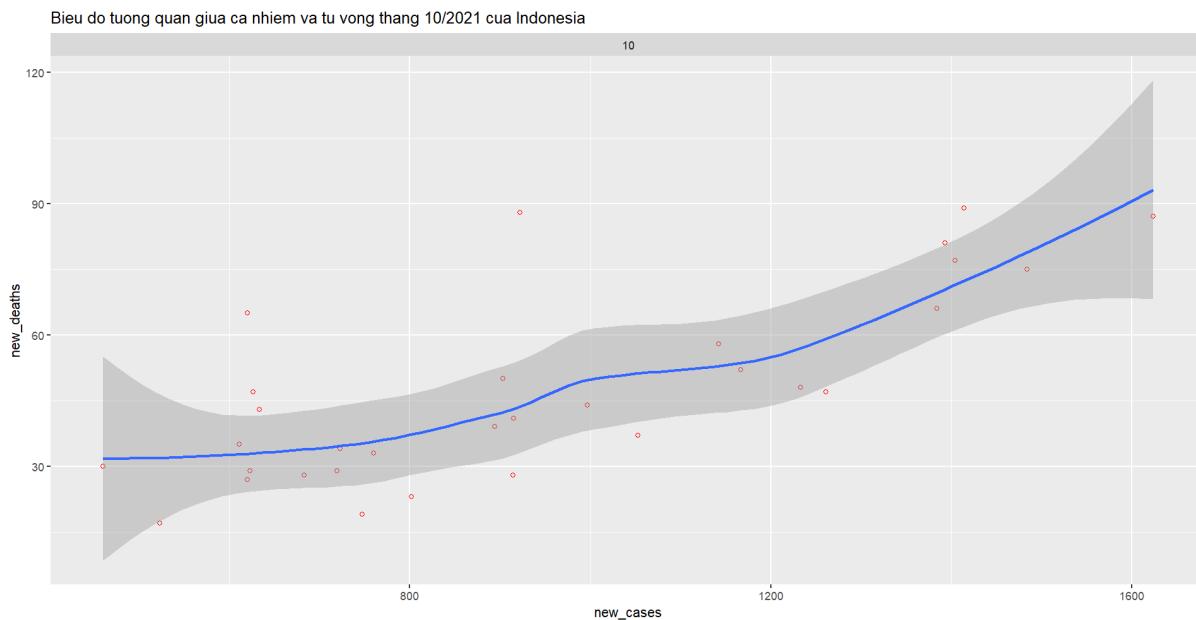
Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 2-2021 của Indonesia là 0.108853, hướng tương quan dương và có sự tương quan tuyến tính **rất ý êu** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



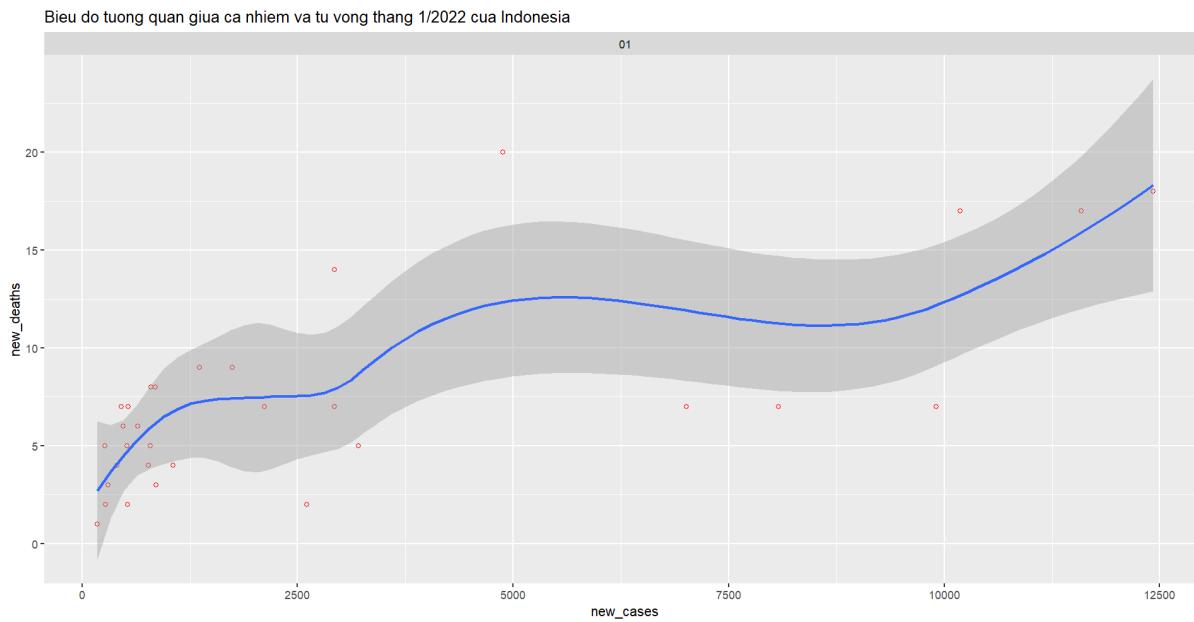
Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 4-2021 của Indonesia là 0.6400672, hướng tương quan dương và có sự tương quan tuyến tính **vừa** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



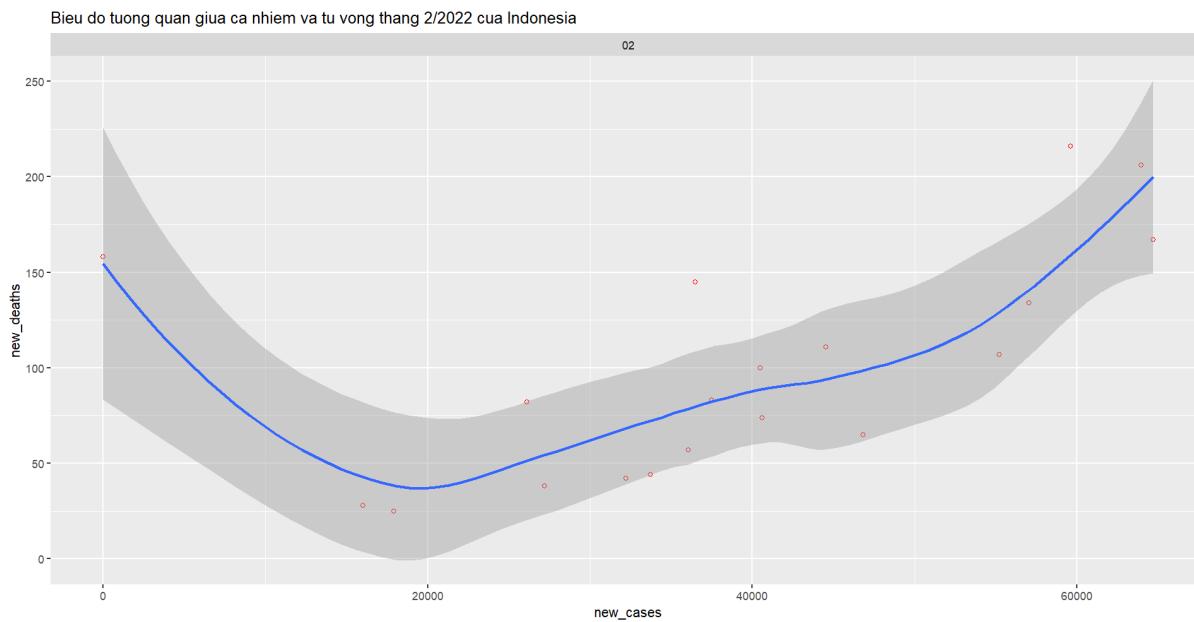
Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 10-2021 của Indonesia là 0.7554078, hướng tương quan dương và có sự tương quan tuyến tính **mạnh** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 1-2022 của Indonesia là 0.688997, hướng tương quan dương và có sự tương quan tuyến tính **vừa** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



Hệ số tương quan giữa số ca nhiễm và số ca tử vong vào tháng 2-2022 của Indonesia là 0.5494153, hướng tương quan dương và có sự tương quan tuyến tính **vừa** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:





- 3) Xét tương quan trong mỗi tháng theo trung bình 7 ngày gần nhất - Phương pháp làm: tương tự như câu 2) ở phía trên nhưng trước khi sử dụng data frame thì ta phải tính lại dữ liệu ở 2 cột `new_cases` và `new_deaths` theo trung bình 7 ngày gần nhất bằng cách sử dụng vòng lặp for. Khi đó ta tính theo 2 trường hợp:

- Trường hợp 1: Tại thời điểm xét số ngày trước đó ít hơn 7 ngày thí sẽ được tính như sau:

$$\text{day1} = \text{day1}/1$$

$$\text{day2} = (\text{day1} + \text{day2})/2$$

...

$$\text{day7} = (\text{day1} + \text{day2} + \dots + \text{day7})/7$$

- Trường hợp 2: từ ngày thứ 8 trong data frame trở đi ta tính như sau:

$$\text{day}(n) = (\text{day}(n-6) + \text{day}(n-5) + \dots + \text{day}(n))/7$$

- Code mà nhóm thực hiện: (tương tự cho 2 nước còn lại là Japan và Vietnam)

```
mydata <- read.csv("owid-covid-data.csv", header = TRUE)
save(mydata, file = 'mydata.rda')
indo <- mydata[mydata$location == "Indonesia", ]
attach(indo)

indo$new_cases[is.na(indo$new_cases)] <- 0
indo$new_deaths[is.na(indo$new_deaths)] <- 0

indo %>%
  select(c(new_cases, new_deaths)) %>%
  abs()
indo$date <- as.Date(indo$date, "%m/%d/%Y")

indo$year <- strftime(indo$date, "%Y")
indo$month <- strftime(indo$date, "%m")
indo$dates<- strftime(indo$date, "%d")

result1 <- vector(length = 720)
for(i in 1: 7){
  result1[i] <- mean(indo$new_cases[1:i])
}
for (i in 8: 720){
  result1[i] <- mean(indo$new_cases[(i-6):i])
}
indo$new_cases <- result1

result2 <- vector(length = 720)
for(i in 1: 7){
  result2[i] <- mean(indo$new_deaths[1:i])
}
for (i in 8: 720){
  result2[i] <- mean(indo$new_deaths[(i-6):i])
}
indo$new_deaths <- result2

indo_2020 <- subset(indo, (year == "2020"))
indo_2021 <- subset(indo, (year == "2021"))
indo_2022 <- subset(indo, (year == "2022"))

indo_2020_4 <- subset(indo_2020, (month == "04"))
indo_2020_10 <- subset(indo_2020, (month == "10"))

indo_2021_1 <- subset(indo_2021,(month=="01"))
indo_2021_2 <- subset(indo_2021,(month=="02"))
indo_2021_4 <- subset(indo_2021,(month=="04"))
```



```
indo_2021_10 <- subset(indo_2021,(month=="10"))

indo_2022_1 <- subset(indo_2022,(month=="01"))
indo_2022_2 <- subset(indo_2022,(month=="02"))

ggplot(indo_2020_4,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color=
  "red")+facet_wrap( ~ month)+ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 4/2020 cua Indonesia") +
  geom_smooth()
print(cor(indo_2020_4$new_cases,indo_2020_4$new_deaths))

ggplot(indo_2020_10,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color
  ="red")+facet_wrap( ~ month)+ ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 10/2020 cua Indonesia") +
  geom_smooth()
print(cor(indo_2020_10$new_cases,indo_2020_10$new_deaths))

ggplot(indo_2021_1,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color=
  "red")+facet_wrap( ~ month)+ ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 1/2021 cua Indonesia") +
  geom_smooth()
print(cor(indo_2021_1$new_cases,indo_2021_1$new_deaths))

ggplot(indo_2021_2,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color=
  "red")+facet_wrap( ~ month)+ ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 2/2021 cua Indonesia") + geom
  _smooth()
print(cor(indo_2021_2$new_cases,indo_2021_2$new_deaths))

ggplot(indo_2021_4,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color=
  "red")+facet_wrap( ~ month)+ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 4/2021 cua Indonesia")+ geom
  _smooth()
print(cor(indo_2021_4$new_cases,indo_2021_4$new_deaths))

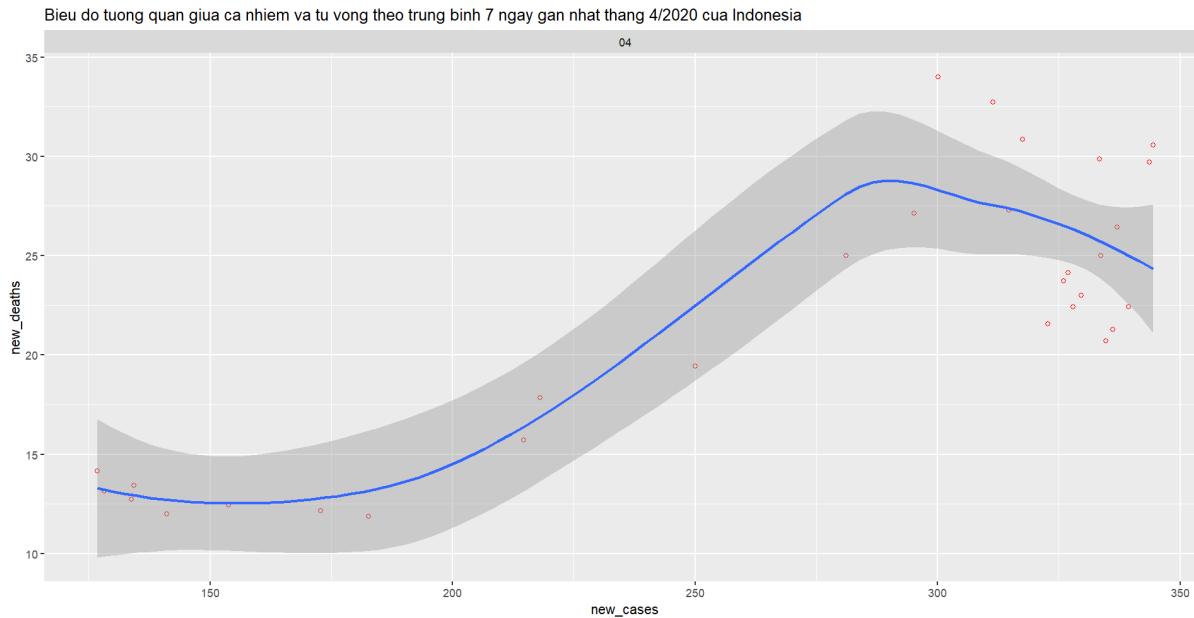
ggplot(indo_2021_10,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color
  ="red")+facet_wrap( ~ month)+ ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 10/2021 cua Indonesia")+
  geom_smooth()
print(cor(indo_2021_10$new_cases,indo_2021_10$new_deaths))

ggplot(indo_2022_1,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color=
  "red")+facet_wrap( ~ month)+ ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 1/2022 cua Indonesia") + geom
  _smooth()
print(cor(indo_2022_1$new_cases,indo_2022_1$new_deaths))

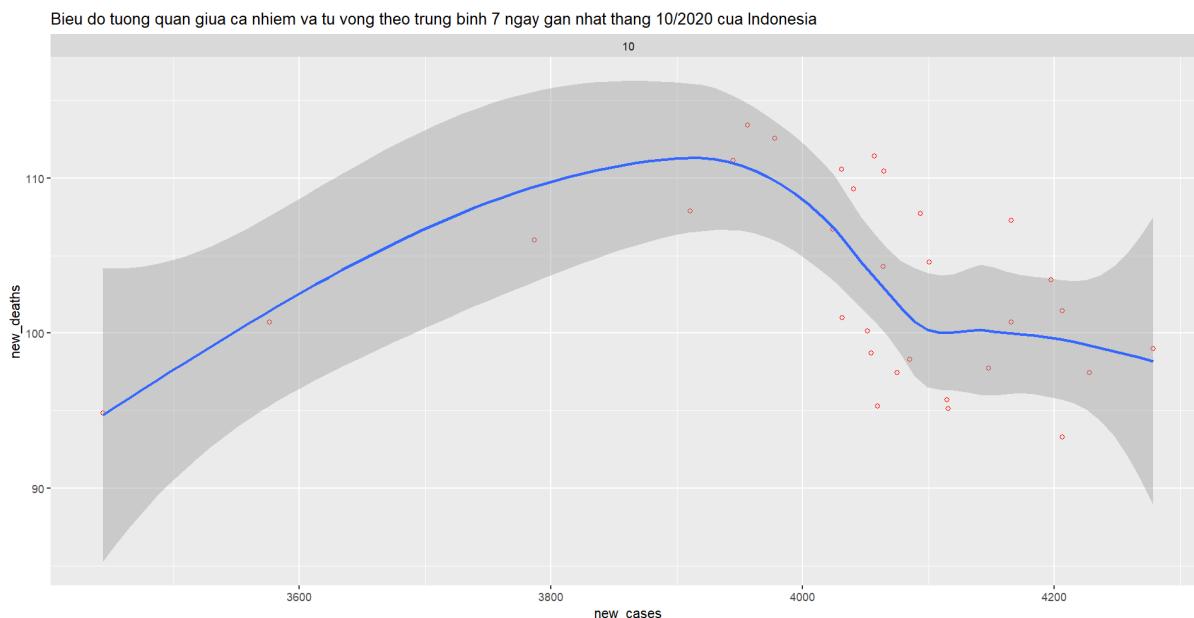
ggplot(indo_2022_2,(aes(x=new_cases,y=new_deaths)))+geom_point(shape=1,color
  ="red")+facet_wrap( ~ month)+ ggtitle("Bieu do tuong quan giua ca nham va
  tu vong theo trung binh 7 ngay gan nhat thang 2/2022 cua Indonesia") + geom
  _smooth()
print(cor(indo_2022_2$new_cases,indo_2022_2$new_deaths))
```

- Output câu 3 của Indonesia:

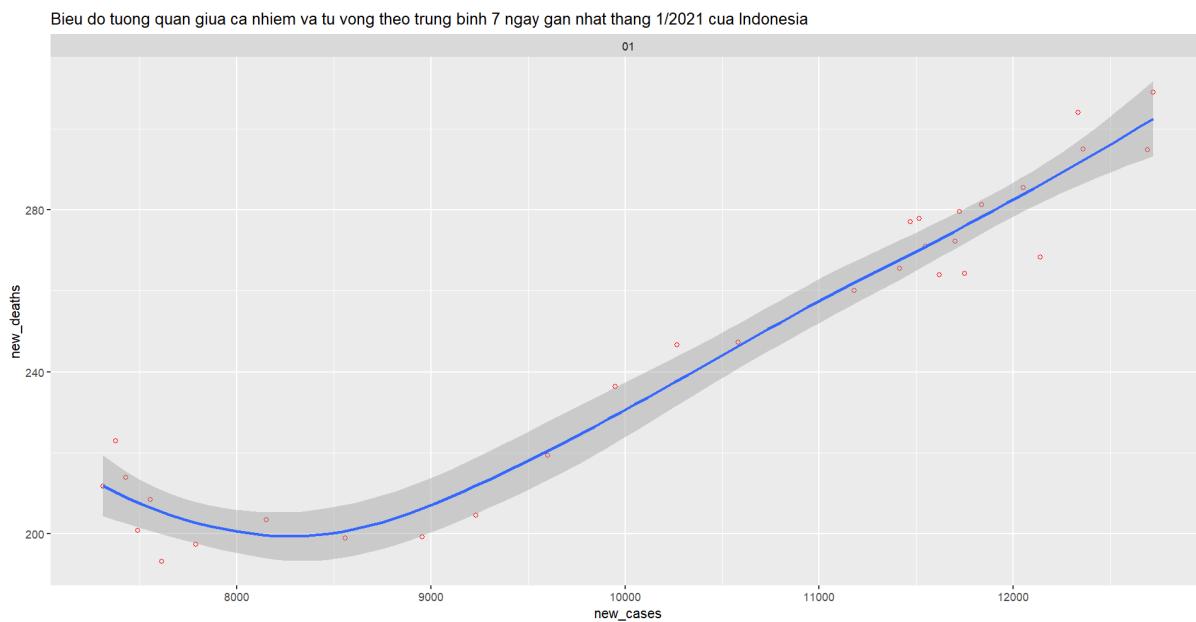
Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 4-2020 của Indonesia là 0.8372391, hướng tương quan dương và có sự tương quan tuyến tính **mạnh** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



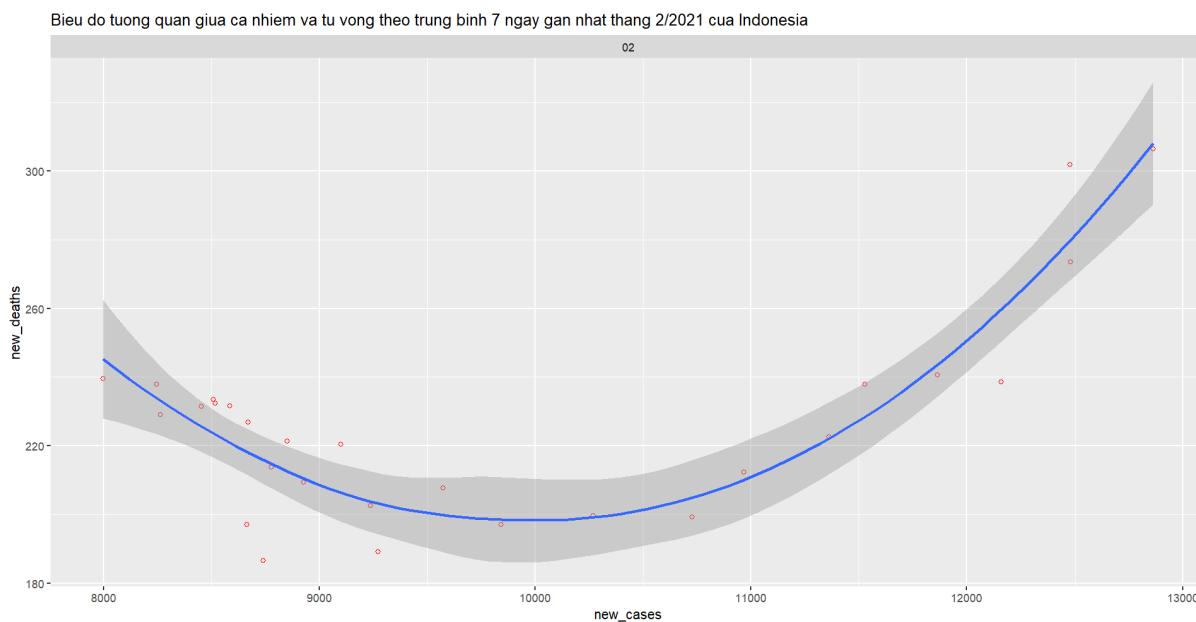
Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 10-2020 của Indonesia là -0.09688588, hướng tương quan âm và có sự tương quan tuyến tính **rất yếu** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



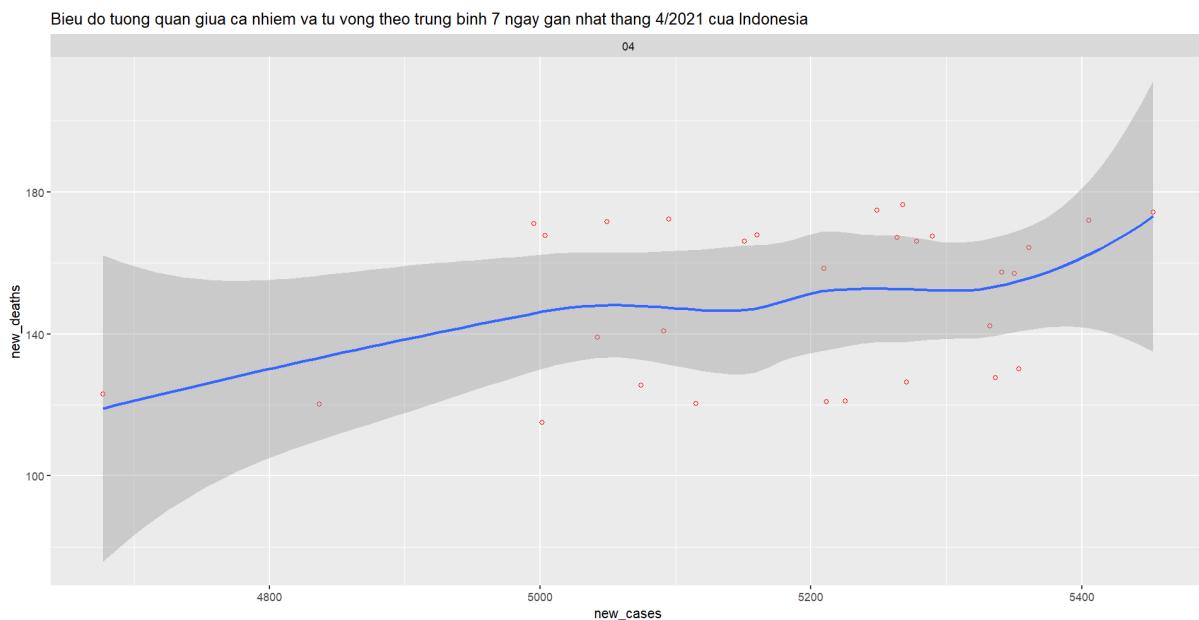
Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 1-2021 của Indonesia là 0.9401912, hướng tương quan dương và có sự tương quan tuyến tính **rất mạnh** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



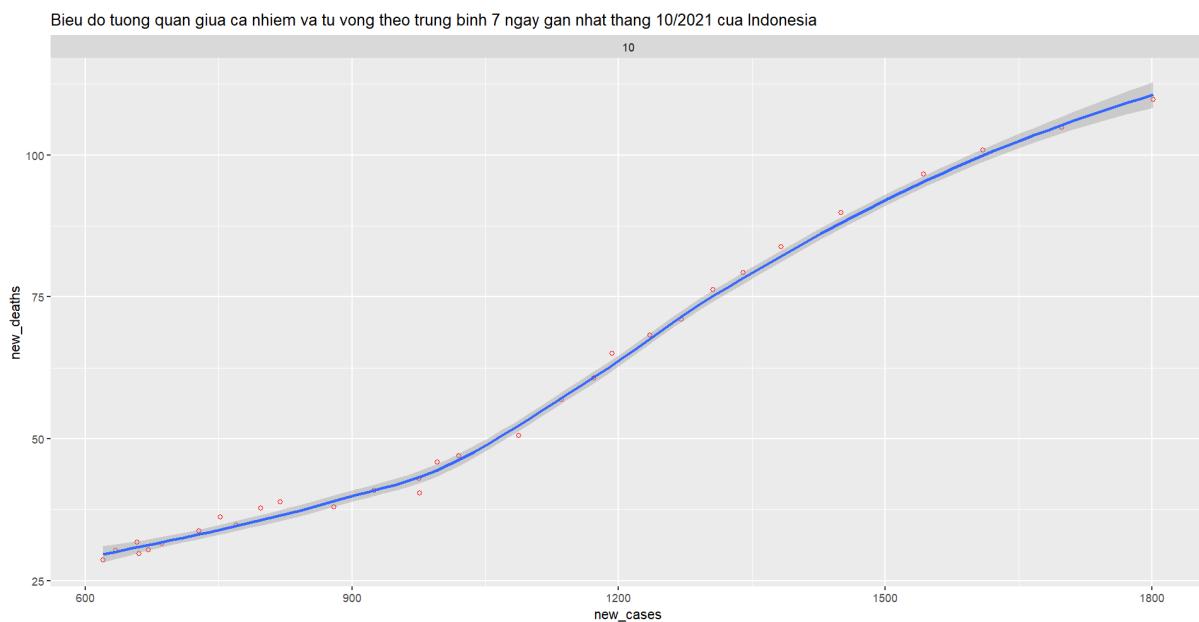
Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 2-2021 của Indonesia là 0.5482933, hướng tương quan dương và có sự tương quan tuyến tính **vừa** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



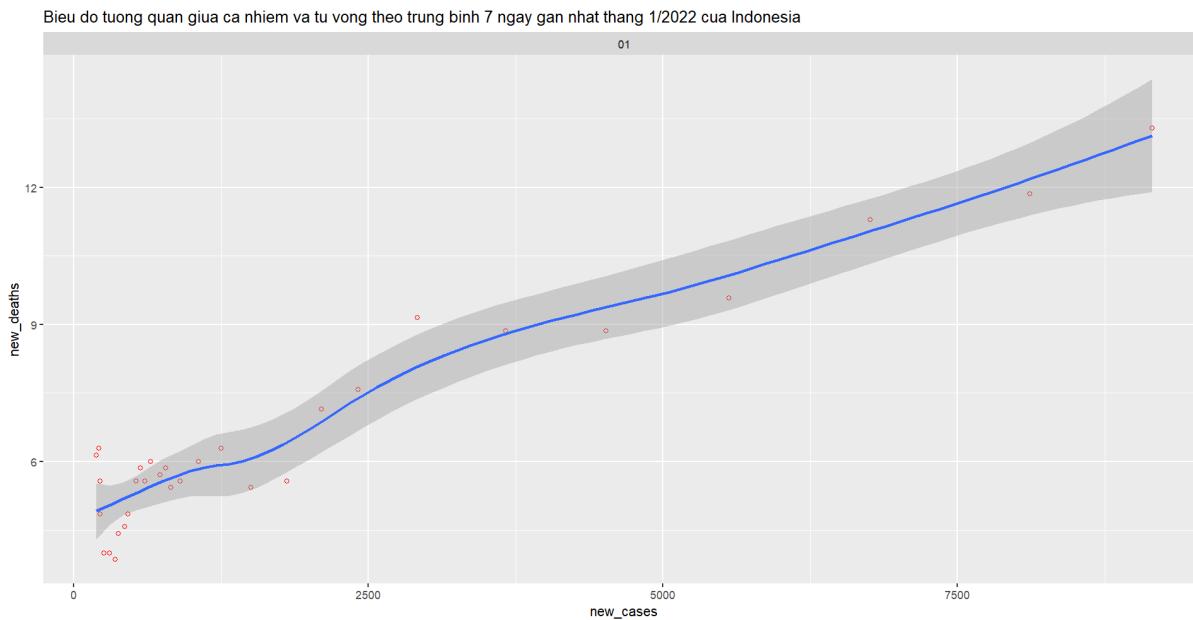
Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 4-2021 của Indonesia là 0.343823, hướng tương quan dương và có sự tương quan tuyến tính **yếu** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



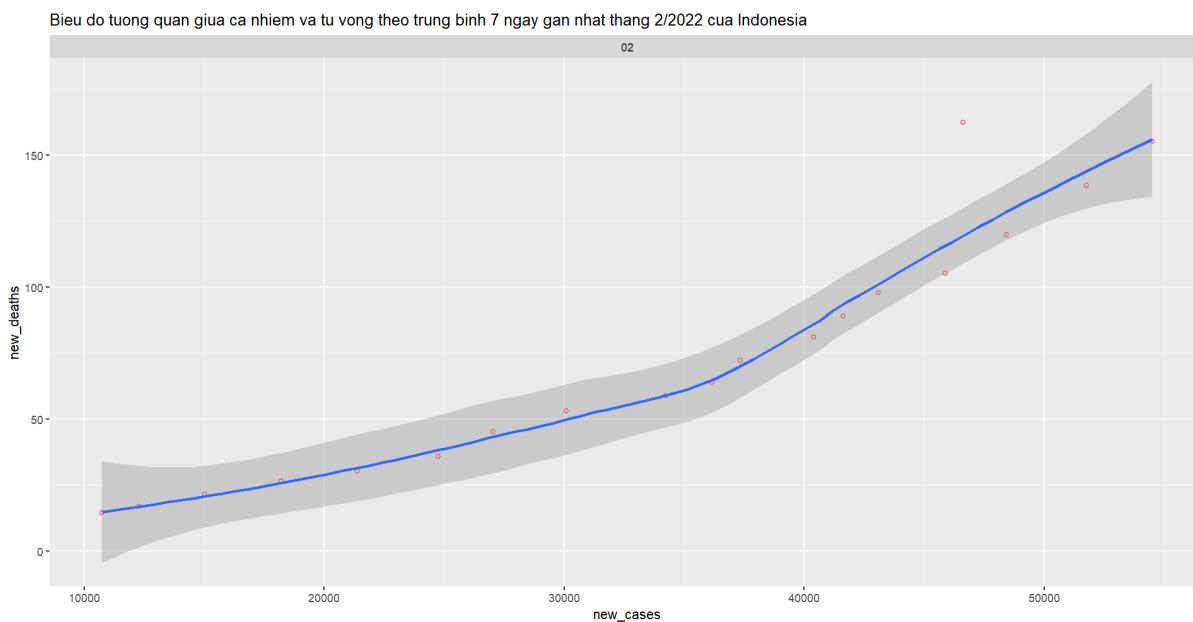
Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 10-2021 của Indonesia là 0.985146, hướng tương quan dương và có sự tương quan tuyến tính **rất mạnh** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 1-2022 của Indonesia là 0.955116, hướng tương quan dương và có sự tương quan tuyến tính **rất mạnh** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:



Hệ số tương quan giữa số ca nhiễm và số ca tử vong theo trung bình 7 ngày gần nhất vào tháng 2-2022 của Indonesia là 0.9436106, hướng tương quan dương và có sự tương quan tuyến tính **rất mạnh** giữa số ca nhiễm và số ca tử vong. Biểu đồ thể hiện sự tương quan như hình dưới:

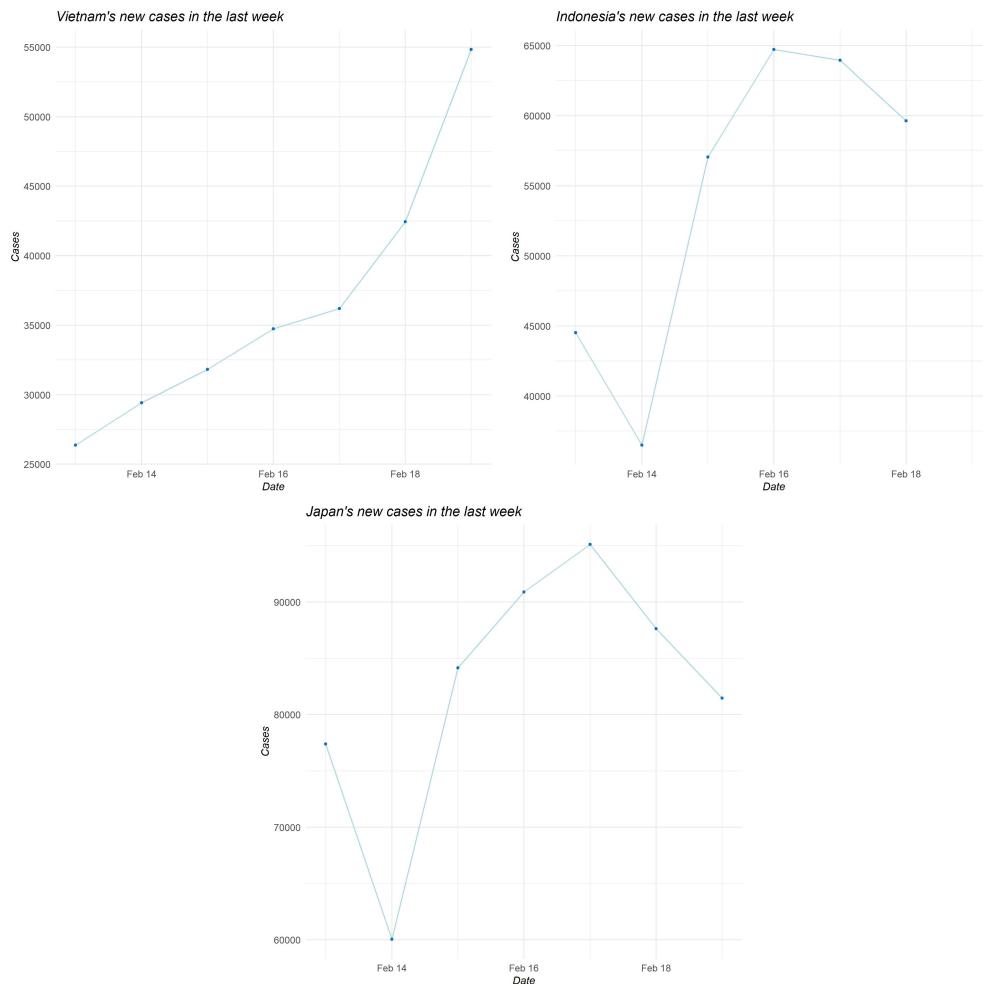


## x Nhóm câu hỏi riêng

Do MADE của nhóm là 1204 nên nhóm sẽ thực hiện các câu hỏi 1, 2, 4, 10.

1) So sánh tình trạng nhiễm bệnh của các quốc gia trong 7 ngày cuối của năm cuối cùng

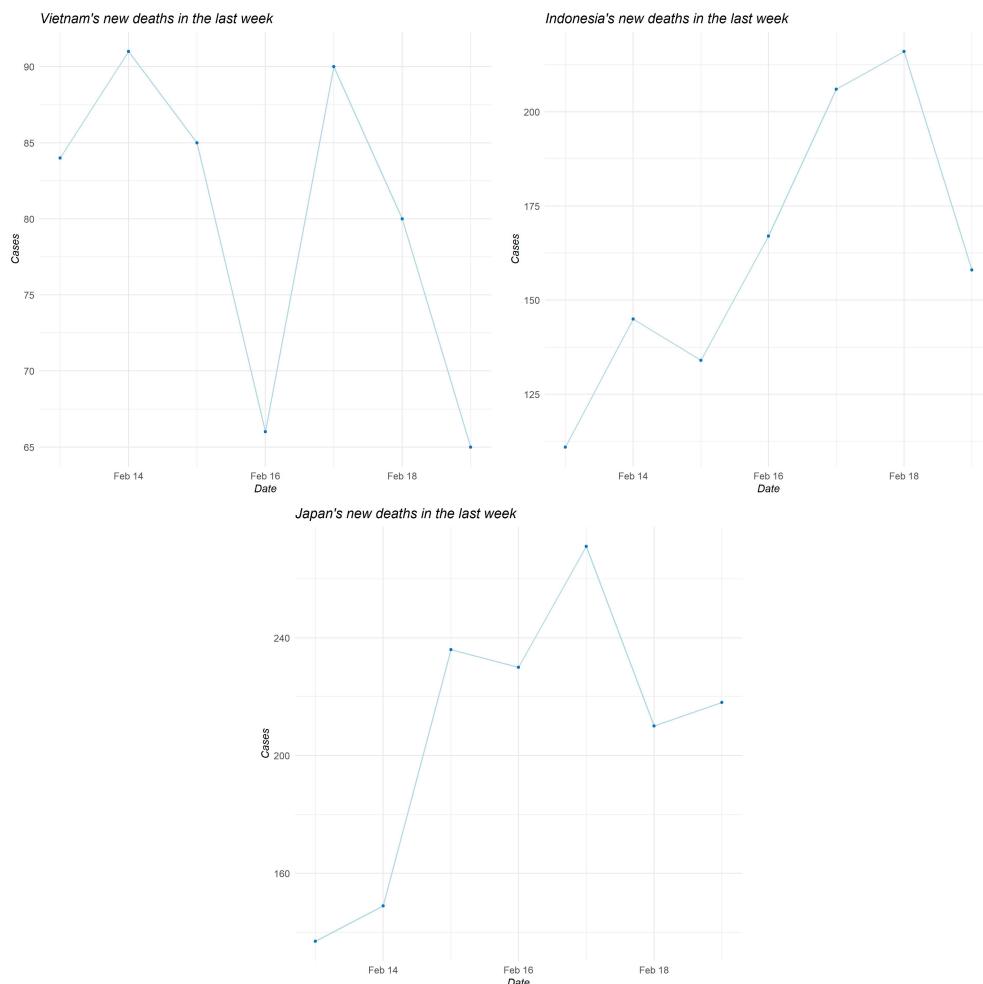
Chúng ta sẽ sử dụng biểu đồ đã vẽ được ở câu 3 phần *iv* để so sánh:



### Nhận xét:

- Trong tuần cuối năm, Japan là nước có số ca nhiễm bệnh **cao nhất** trong 3 nước với ngày cao nhất là khoảng 95000 ca nhiễm bệnh.
- Trong tuần cuối năm, Vietnam là nước có số ca nhiễm bệnh **thấp nhất** trong 3 nước với ngày cao nhất là khoảng 55000 ca nhiễm bệnh.
- Tuy nhiên, ở Vietnam, trong tuần cuối số ca nhiễm bệnh tăng dần, không có xu hướng giảm trong cả tuần.
- Ở Japan và Indonesia, trong tuần cuối số ca nhiễm bệnh có xu hướng tương tự nhau, khi mà số ca nhiễm giảm ở ngày đầu tiên của tuần, sau đó tăng mạnh ở 3 ngày tiếp theo, và có xu hướng giảm ở 2 ngày cuối tuần.

2) So sánh tình trạng tử vong của các quốc gia trong 7 ngày cuối của năm cuối cùng  
Chúng ta sẽ sử dụng biểu đồ đã vẽ được ở câu 4 phần *iv* để so sánh:



#### Nhận xét:

- Trong tuần cuối năm, Japan là nước có số ca tử vong **cao nhất** trong 3 nước với ngày cao nhất là khoảng 300 ca tử vong(17/02/2022) và thấp nhất là khoảng gần 140 ca tử vong vào ngày đầu tuần (13/02/2022).
- Trong tuần cuối năm, Vietnam là nước có số ca tử vong **thấp nhất** trong 3 nước với ngày cao nhất là khoảng 92 ca tử vong(14/02/2022) và thấp nhất là khoảng gần 65 ca tử vong vào ngày cuối tuần (19/02/2022).
- Ở Indonesia, số ca tử vong cao nhất trong ngày là khoảng 213 ca(18/02/2022) và thấp nhất là khoảng 112 ca(13/02/2022).

4) Với k là mốc bùng phát dịch, hãy xác định k và cho biết các khoảng thời gian bùng phát Nhóm lấy ví dụ quốc gia Việt Nam (tương tự với Indonesia và Japan) và k = 20000.

```
k = 20000

x4_Vietnam = mydata %>% filter(mydata$location == "Vietnam")
x4_Vietnam = x4_Vietnam[,4:6]
i = 1
while (i < nrow(x4_Vietnam)){
  temp = c()
  if (x4_Vietnam[i,2] >= k){
```

```

temp = cbind(temp, x4_Vietnam[i,1], ' - ')
while (x4_Vietnam[i,2] >= k){
  i = i + 1
  if (i > nrow(x4_Vietnam)) break
}
temp = cbind(temp, x4_Vietnam[i-1,1])
cat(temp, '\n')
}
else i = i+1
}

```

Kết quả:

```

11/23/2021 - 11/23/2021
12/16/2021 - 12/16/2021
12/30/2021 - 12/30/2021
1/4/2022 - 1/4/2022
1/6/2022 - 1/6/2022
1/9/2022 - 1/9/2022
1/12/2022 - 1/12/2022
1/18/2022 - 1/18/2022
1/30/2022 - 1/30/2022
2/8/2022 - 2/19/2022

```

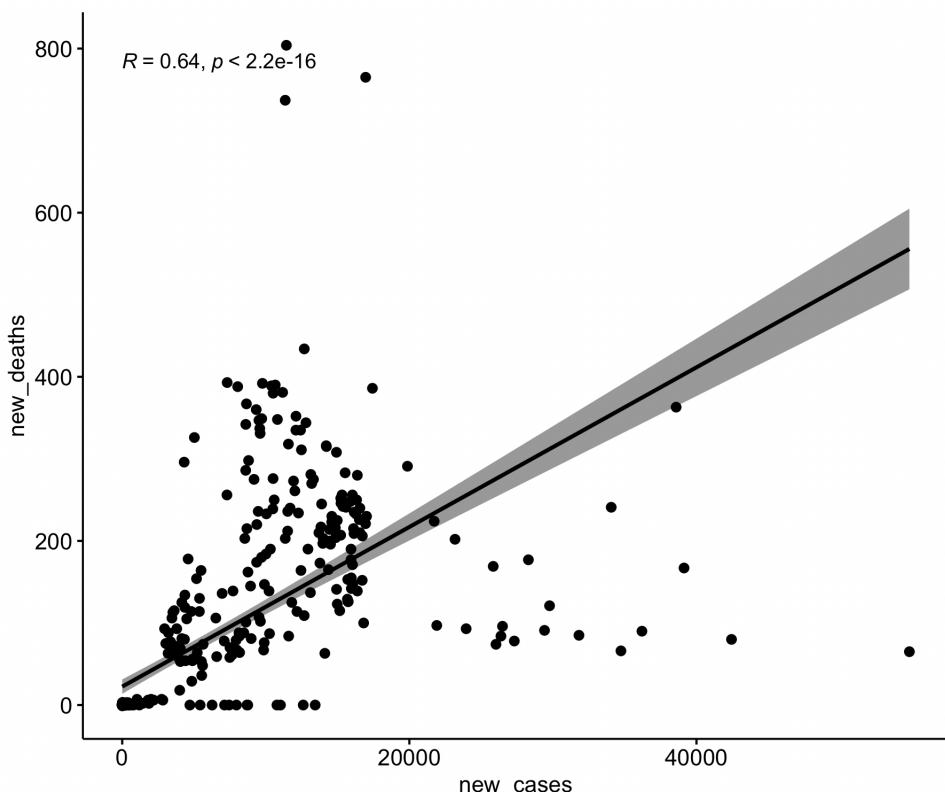
10) Hãy mô tả mối quan hệ tuyến tính giữa nhiễm bệnh và tử vong bằng cách đo độ kết hợp của mối quan hệ dùng correlation r (correlation coefficient) và hướng kết hợp.

```

new_cases = x4_Vietnam$new_cases
new_deaths = x4_Vietnam$new_deaths
ggscatter(x4_Vietnam, x="new_cases", y="new_deaths", add = "reg.line", conf.int
          = TRUE, cor.coef = TRUE, cor.method = "pearson")
cor.test(new_cases, new_deaths)

```

Kết quả:



Hình 4: Biểu đồ thể hiện quan hệ tuyến tính giữa nhiễm bệnh và tử vong của Việt Nam



#### Pearson's product-moment correlation

```
data: new_cases and new_deaths
t = 19.645, df = 567, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.584786 0.682870
sample estimates:
cor
0.6363934
```

**Hình 5:** Tính hệ số tương quan theo phương pháp Pearson

Nhận xét:

Dựa vào hàm cor.test(), ta tính được hệ số tương quan cor = 0.64, nghĩa là nhiễm bệnh và tử vong có mối quan hệ đồng biến và mức độ tương quan vừa ( $0.5 < |cor| < 0.7$ ).

Quan sát biểu đồ hình 4 kết hợp sử dụng kết quả tính toán ở hình 5, ta có thể tính được số ca nhiễm dựa vào số ca tử vong và ngược lại.

Ta cũng có thể dự đoán được các mốc bùng phát hay mốc tử vong với độ chính xác không quá cao, từ đó đưa ra các biện pháp phòng chống hợp lý.

## 6 Kết luận

### i Ngôn ngữ R

- Được biết R là một ngôn ngữ lập trình và môi trường phần mềm dành cho tính toán và đồ họa thống kê. Ngôn ngữ R đã trở thành một tiêu chuẩn trên thực tế giữa các nhà thống kê và được sử dụng rộng rãi để phát triển phần mềm thống kê và phân tích dữ liệu.
- Sau khi tiếp xúc và làm việc với R, nhóm đã có thể sử dụng thành thạo các thao tác cơ bản trên R để hoàn thành BTL CTRR về phân tích và thống kê dữ liệu COVID - 19.

### ii Phân tích và thống kê

- Hiểu được tầm quan trọng của thống kê. Thống kê là một công việc vô cùng khó khăn, đòi hỏi sự chính xác tuyệt đối, làm việc với khối lượng dữ liệu khổng lồ. Có vai trò quan trọng và là nền tảng cho các phân tích.
- Từ đó thực hiện các tính toán, phân tích, dự đoán xu hướng thay đổi và phát triển của dữ liệu. Cuối cùng, dựa vào các phân tích đó để đưa ra phương hướng xử lý vấn đề hiệu quả và an toàn nhất.