

DẠI 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)

Thống kê khảo sát kết quả Covid-19
môn Cấu trúc rời rạc

GVHD: Huỳnh Tường Nguyên
Nguyễn Ngọc Lễ

SV thực hiện: Lê Nhật Anh – 2153158
Kim Gia Bảo – 2152417
Đỗ Nhật Thái – 2152964
Nguyễn Thế Cường – 2153240

Tp. Hồ Chí Minh, Tháng 04/2022



Mục lục

1 Động cơ nghiên cứu	2
2 Mục tiêu	2
3 Cơ sở lý thuyết	2
3.1 Bách phân vị và tứ phân vị	2
3.1.1 Định nghĩa	2
3.1.2 Công thức	2
3.2 Giá trị trung bình	3
3.2.1 Định nghĩa	3
3.2.2 Công thức	3
3.3 Độ lệch chuẩn	3
3.3.1 Định nghĩa	3
3.3.2 Công thức	3
3.4 Dữ liệu ngoại lệ	3
3.4.1 Định nghĩa	3
3.5 Biểu đồ hộp	4
3.5.1 Định nghĩa	4
3.6 Tần số tích lũy và tần số tích lũy tương đối	4
3.6.1 Định nghĩa	4
3.6.2 Công thức	4
4 Mô tả dữ liệu	4
5 Nhiệm vụ	4
6 Hướng dẫn và yêu cầu	117
6.1 Hướng dẫn	117
6.2 Yêu cầu	118
6.3 Nộp bài	118
7 Cách đánh giá và xử lý gian lận	118
7.1 Dánh giá	118
7.2 Xử lý gian lận	118
Tài liệu	118



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 mới 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 <https://github.com/owid/covid-19-data/blob/master/public/data/README.md> 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 Cơ sở lý thuyết

3.1 Bách phân vị và tứ phân vị

3.1.1 Định nghĩa

- Bách phân vị (Percentile) là đại lượng dùng để ước tính tỷ lệ dữ liệu trong một tập số liệu rơi vào vùng cao hơn hoặc thấp hơn so với một giá trị cho trước. Bách phân vị chia dữ liệu có thứ tự theo hàng trăm.
- Tứ phân vị (Quartile) là một trường hợp đặc biệt của bách phân vị. Tứ phân vị có 3 giá trị, đó là tứ phân vị thứ nhất, thứ nhì, và thứ ba. Ba giá trị này chia một tập hợp dữ liệu đã sắp xếp theo thứ tự thành 4 phần có số lượng quan sát đều nhau.

3.1.2 Công thức

a) Xác định giá trị bách phân vị

Để xác định giá trị (vp) của phân vị thứ p trong một tập dữ liệu, ta thực hiện theo các bước sau:

- Sắp xếp dữ liệu theo thứ tự từ nhỏ nhất đến lớn nhất.
- Tính chỉ số i

$$i = \frac{p \times (n + 1)}{100}$$

Trong đó:

- *i* là vị trí của giá trị dữ liệu tại phân vị thứ p
- *p* là phân vị thứ p
- *n* là tổng số quan sát

b) Xác định giá trị tứ phân vị:

- Giá trị tứ phân vị thứ nhất Q1 bằng trung vị phần dưới, tương đương với bách phân vị thứ 25.
- Giá trị tứ phân vị thứ hai Q2 chính bằng giá trị trung vị, tương đương với bách phân vị thứ 50.
- Giá trị tứ phân vị thứ ba Q3 bằng trung vị phần trên, tương đương với bách phân vị thứ 75.



3.2 Giá trị trung bình

3.2.1 Định nghĩa

Giá trị trung bình là một loại trung bình được tính bằng cách chia tổng của một tập hợp số cho số lượng các số trong tập hợp đó.

3.2.2 Công thức

$$\bar{a} = \frac{(a_1 + a_2 + \dots + a_n)}{n} = \frac{(\sum a)}{n}$$

Trong đó:

- \bar{a} là giá trị trung bình
- a_1, a_2, \dots, a_n là các số trong tập hợp
- n là số các số lượng các số trong tập hợp

3.3 Độ lệch chuẩn

3.3.1 Định nghĩa

Độ lệch chuẩn (Standard deviation) là thước đo độ phân tán của các giá trị trong một tập dữ liệu đã cho từ giá trị trung bình của chúng. Nó cho biết trung bình mỗi giá trị nằm bao xa so với giá trị trung bình.

3.3.2 Công thức

a) Đối với dữ liệu là một tổng thể:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

Trong đó:

- x_i là giá trị của quan sát thứ i
- μ là giá trị trung bình tổng thể
- N là tổng số quan sát của tổng thể

b) Đối với dữ liệu là một mẫu từ tổng thể:

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

Trong đó:

- x_i là giá trị của quan sát thứ i
- \bar{x} là giá trị trung bình của mẫu dữ liệu
- n là số quan sát trong mẫu dữ liệu

3.4 Dữ liệu ngoại lệ

3.4.1 Định nghĩa

Dữ liệu ngoại lệ (Outliers) là một điểm dữ liệu có sự khác biệt đáng kể so với các quan sát khác. Dữ liệu ngoại lệ có thể xuất hiện do sự thay đổi thang đo hoặc do lỗi từ dữ liệu thu thập (thông thường dữ liệu ngoại lệ dạng này sẽ bị loại khỏi tập dữ liệu). Một giá trị ngoại lệ có thể gây ra vấn đề nghiêm trọng trong quá trình phân tích dữ liệu.



3.5 Biểu đồ hộp

3.5.1 Định nghĩa

Biểu đồ hộp (Box plot) hay còn gọi là biểu đồ hộp và râu (Box and whisker plot) là biểu đồ diễn tả 5 vị trí phân bố của dữ liệu, đó là: giá trị nhỏ nhất (min), tứ phân vị thứ nhất (Q1), trung vị (median), tứ phân vị thứ 3 (Q3) và giá trị lớn nhất (max).

3.6 Tần số tích lũy và tần số tích lũy tương đối

3.6.1 Định nghĩa

- Các tần số tích lũy là tổng các tần số tuyệt đối f , từ tần số thấp nhất đến tần số thấp nhất tương ứng với một giá trị nào đó của biến. Đổi lại, tần số tuyệt đối là số lần một quan sát xuất hiện trong tập dữ liệu
- AF_{Toi} còn được gọi là tần số tích lũy tuyệt đối. Nếu chia cho tổng dữ liệu, chúng ta có tần số tích lũy tương đối, có tổng cuối cùng phải bằng 1.

3.6.2 Công thức

- Tần suất tích lũy của một giá trị nhất định của biến X_{Toi} là tổng các tần số tuyệt đối f của tất cả các giá trị nhỏ hơn hoặc bằng nó:

$$F_{Toi} = f_1 + f_2 + f_3 + \dots + F_{Toi}$$

- Tần số tích lũy tương đối: thu được bằng cách chia tần số tích lũy tuyệt đối f_{Toi} cho tổng dữ liệu N :

$$F_r = \frac{f_{Toi}}{N}$$

4 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.

- iso_code*: Định danh đất nước
- continent*: Tên châu lục
- location*: Tên quốc gia
- date*: Ngày quan sát với định dạng Month-Day-Year
- new_cases*: Số trường hợp COVID-19 mới được xác nhận
- new_deaths*: Số tử vong mới do COVID-19

5 Nhiệm vụ

Gọi MD là mã đề riêng cho mỗi nhóm (gồm 4 ký số) không trùng nhau, nhóm sinh viên sẽ thực hiện các yêu cầu dưới đây với các giá trị xác định như sau: $MD = 2179$

- Mỗi nhóm sẽ dùng R để thao tác trên số file dữ liệu khác nhau được chọn theo cột "STT" theo cách tính $kq = (kytu1 + kytu2 + kytu3 + kytu4) \% 6$:
 - Nếu $kq = 0$ thì làm các stt là 1,2,3
 - Nếu $kq = 1$ thì làm các stt là 4,5,6
 - Nếu $kq = 2$ thì làm các stt là 7,8,9
 - Nếu $kq = 3$ thì làm các stt là 10,11,12



- Nếu $kq = 4$ thì làm các stt là 13,14,15
- Nếu $kq = 5$ thì làm các stt là 16,17,18
- $kq = (2 + 1 + 7 + 9) \% 6 = 1$ làm các stt là 4,5,6

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

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

Source code

```
library(lubridate)
library(tidyverse)
library(Hmisc)
library(data.table)
library(gridExtra)

#data_input_global_declare
rm(list = ls(all.names = TRUE))
source <- read.csv(file.choose())
data <- source[source[,2]!=""]
data[,5] <- abs(data[,5])
data[,6] <- abs(data[,6])
options("scipen"=10)

world_data <- subset(source, source$location == "World")
iso_code <- c(data$iso_code)
continent <- c(data$continent)
new_cases <- c(data$new_cases)
death_cases <- c(data$new_deaths)
```

- Xóa các data từ lần sử dụng trước bằng lệnh $rm(list = ls(all.names = TRUE))$.
- Khai báo các thư viện cần thiết.
- Tiến hành chọn file dữ liệu và đưa vào biến data.
- Loại các dòng không phải đất nước ra khỏi dữ liệu.

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

Source code

```
i1 <- function()
{
  date_format <- data
  date_format$date <- as.POSIXct(date_format$date, format = "%m/%d/%Y")
  outputi1 <- rbind(unique(format(date_format$date, format = "%Y")))
  View(outputi1)
}
```



- Đầu tiên ta đưa dữ liệu vào biến data_format
- Sau đó ta định dạng lại cột date lại thành kiểu dữ liệu *Date*
- Sử dụng hàm *unique* theo format là *%Y* để lấy ra các năm theo yêu cầu đề bài.

	V1	V2	V3
1	2020	2021	2022

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

iso_code:	Country
AFG	Afghanistan
OWID_AFR	Africa
ALB	Albania
Count	Số đất nước

Source code

```
i2 <- function()
{
  iso_code1 <- unique(data[,1])
  location1 <- unique(data[,3])
  output_i2 <- rbind(cbind(iso_code1[1:10],location1[1:10]),
                     cbind("Count",length(iso_code1)))
  colnames(output_i2) <- c("iso_code","location")
  View(output_i2)
}
```

- Ta dùng hàm *unique* tìm các đất nước và iso_code chỉ xuất hiện đúng 1 lần trong data gán vào 2 biến. Sau đó ta dùng *rbind* để tạo data chứa 10 giá trị đầu tiên ở 2 biến iso_code1 và location1.

	iso_code	location
1	AFG	Afghanistan
2	ALB	Albania
3	DZA	Algeria
4	AND	Andorra
5	AGO	Angola
6	AIA	Anguilla
7	ATG	Antigua and Barbuda
8	ARG	Argentina
9	ARM	Armenia
10	ABW	Aruba
11	Count	225

3) Số lượng chủng tộc trong tập mẫu



Continent : Số châu lục
Africa: Châu phi
Asia: Châu Á

Source code

```
i3 <- function()
{
  conti <- cbind( unique(data$continent) )
  conti_trans <- rbind("Chau_A", "Chau_Au",
                       "Chau_Phi", "Bac_Mi",
                       "Nam_Mi", "Chau_Dai_Duong")
  output_i3 <- data.frame(
    conti,
    conti_trans
  )
  output_i3 = rbind(c("Continent", length(conti)), output_i3)
  View(output_i3)
}
```

- Ta sử dụng hàm `unique` để các châu lục chỉ xuất hiện duy nhất 1 lần rồi gán cho biến `conti`, sử dụng hàm `rbind` để tạo một cột `conti_trans` chứa phần phiên dịch tên các châu lục. Cuối cùng ta dùng hàm `length` để đếm số lượng các phần tử của biến `conti`

	conti	conti_trans
1	Continent	6
2	Asia	Chau A
3	Europe	Chau Au
4	Africa	Chau Phi
5	North America	Bac Mi
6	South America	Nam Mi
7	Oceania	Chau Dai Duong

- 4) Số lượng dữ liệu thể hiện thu thập dữ liệu được trong từng từng châu lục và tổng số

Continent:	Observations
Africa	value1
Asia	value2
Tổng:	giá trị tổng

Source code

```
i4 <- function()
{
  con_data <- as.numeric(table(data$continent))
  con <- sort(unique(data$continent))
  sum <- c("Tong:", sum(con_data))
  con_data <- data.frame(rbind(cbind(con, con_data), sum))
  colnames(con_data) <- c("Continent:", "Observations")
  rownames(con_data) <- c(1:nrow(con_data))
  View(con_data)
}
```



- Ta dùng hàm `table` để đếm số lần xuất hiện của mỗi châu lục tại cột `Continent`, đây chính là số lượng dữ liệu thu thập đc của mỗi châu lục. Ta biến đổi dữ liệu này thành số và ghép với một cột chứa tên các châu lục tương ứng và tính tổng của chúng rồi ghép vào hàng cuối cùng.

	Continent:	Observations
1	Africa	38647
2	Asia	35528
3	Europe	36375
4	North America	24438
5	Oceania	8993
6	South America	9335
7	Tổng:	153316

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

iso_code	Observations
AFG	value1
OWID_AFR	value2
ALB	value3
Tổng:	giá trị tổng

Source code

```
con_data <- as.numeric(table(data$iso_code))
coun <- sort(unique(data$iso_code))
sum <- c("Tổng:", sum(con_data))
con_data <- data.frame(rbind(cbind(coun, con_data), sum))
colnames(con_data) <- c("iso_code", "Observations")
rownames(con_data) <- c(1:nrow(con_data))
View(tail(con_data, n = 11))
```

- Ta dùng hàm `table` để đếm số lần xuất hiện `iso_code` của mỗi quốc gia tại cột `iso_code`, đây chính là số lượng dữ liệu thu thập đc của mỗi quốc gia. Ta biến đổi dữ liệu này thành số và ghép với một cột chứa các `iso_code` tương ứng và tính tổng của chúng rồi ghép vào hàng cuối cùng sau đó xài hàm `tail()` để xuất ra 10 quốc gia cuối và hàng tổng.

	iso_code	Observations
216	VEN	708
217	VGB	694
218	VNM	759
219	VUT	467
220	WLF	489
221	WSM	459
222	YEM	681
223	ZAF	744
224	ZMB	704
225	ZWE	702
226	Tổng:	153316

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

5	Oceania	8993
---	---------	------

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

1	Africa	38647
---	--------	-------

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

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

Source code

```
coun_data <- as.numeric(table(data$location))
coun <- sort(unique(data$location))
coun_data <- data.frame(cbind(coun, coun_data))
colnames(coun_data) <- c("Country", "Observations")
rownames(coun_data) <- c(1:nrow(coun_data))
coun_data[, 2] <- as.numeric(coun_data[, 2])
View(rbind(subset(coun_data,
                  Observations == min(coun_data$Observations)),
           subset(coun_data,
                  Observations == max(coun_data$Observations))))
```

- Ta cũng dùng hàm `table` tương tự như câu i-5 tuy nhiên thay vì cột `iso_code` thì lần này ta sẽ dùng cột `Location` để lấy tên của đất nước. Sau đó dùng `subset` để lọc ra các hàng đạt giá trị nhó nhất và lớn nhất của data chứa số lượng dữ liệu thu thập được của mỗi nước và xuất ra màn hình.



	Country	Observations
161	Pitcairn	85
8	Argentina	781
129	Mexico	781

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

Source code

```
i10_i11 <- function()
{
  date <- c(data$date)
  iso_code <- c(data$iso_code)
  Date <- data.frame(iso_code, date)

  Date <- Date %>% arrange(mdy(Date$date))
  df <- data.frame(setDT(Date)[, list(numData=N), Date$date])
#i10
  output_i10 = subset(df, numData == min(numData))
  View(output_i10)
#i11
  output_i11 = subset(df, numData == max(numData))
  View(output_i11)
}
```

- Hàm `arrange` từ thư viện `lubridate` có tác dụng sắp xếp ngày trong data frame Date theo thứ tự từ ngày cũ nhất đến ngày mới nhất
- Dòng code tiếp theo có tác dụng sử dụng data frame Date truyền vào, tạo một danh sách `numData` là số lượng dữ liệu theo từng ngày của data frame Date sau đó tạo một data frame df mới chứa ngày `Date` và số lượng dữ liệu thu nhập được `numData`
- Cuối cùng ta lần lượt tạo các biến `output_i10, output_i11` gán lần lượt các subset (tập con) là những ngày có số lượng thu thập dữ liệu thấp nhất và những ngày có lượng thu thập dữ liệu cao nhất

	Date	numData
1	1/1/2020	2
2	1/2/2020	2
3	1/3/2020	2



	Date	numData
600	8/22/2021	225
601	8/23/2021	225
602	8/24/2021	225
603	8/25/2021	225
604	8/26/2021	225
605	8/27/2021	225
606	8/28/2021	225
607	8/29/2021	225

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

Source code

```
i12_i13_i14 <- function ()  
{  
  df <- data.frame(data)  
  df <- subset(df, continent != "")  
  date_sort <- df %>% arrange(mdy(df$date))  
  #View(date_sort)  
  output_i12 <- date_sort %>% group_by(date, continent)  
              %>% summarise(value = n())  
  output_i12 <- output_i12 %>% arrange(mdy(output_i12$date))  
  #i12  
  View(output_i12)  
  #i13  
  output_i13 <- subset(output_i12, value == max(value))  
  View(output_i13)  
  #i14  
  output_i14 <- subset(output_i12, value == min(value))  
  View(output_i14)  
}
```

- Biến df chứa dữ liệu của tất cả continent có giá trị khác "".
- Biến date_sort chứa dữ liệu ngày tháng năm được sắp xếp bằng hàm `arrange` theo date.
- Biến output_i12 chứa dữ liệu đầu ra cho mục i câu 12. Bằng cách nhóm date và continent lại và sau đó sử dụng hàm `summarise(value = n())` để xét trong ngày đó có châu lục nào xuất hiện và nó xuất hiện bao nhiêu lần trong ngày.
- Sử dụng lại dữ liệu từ output_i12 để tìm giá trị lớn nhất và nhỏ nhất theo yêu cầu đề bài.



	date	continent	value
4587	2/17/2022	Oceania	16
4588	2/17/2022	South America	13
4589	2/18/2022	Africa	55
4590	2/18/2022	Asia	48
4591	2/18/2022	Europe	49
4592	2/18/2022	North America	34
4593	2/18/2022	Oceania	16
4594	2/18/2022	South America	13
4595	2/19/2022	Africa	55
4596	2/19/2022	Asia	48
4597	2/19/2022	Europe	49
4598	2/19/2022	North America	34
4599	2/19/2022	Oceania	16
4600	2/19/2022	South America	13
	date	continent	value
518	2/6/2022	Africa	55
519	2/7/2022	Africa	55
520	2/8/2022	Africa	55
521	2/9/2022	Africa	55
522	2/10/2022	Africa	55
523	2/11/2022	Africa	55
524	2/12/2022	Africa	55
525	2/13/2022	Africa	55
526	2/14/2022	Africa	55
527	2/15/2022	Africa	55
528	2/16/2022	Africa	55
529	2/17/2022	Africa	55
530	2/18/2022	Africa	55
531	2/19/2022	Africa	55
	date	continent	value
80	2/9/2020	South America	1
81	2/10/2020	South America	1
82	2/11/2020	South America	1
83	2/12/2020	South America	1
84	2/13/2020	South America	1
85	2/14/2020	South America	1
86	2/15/2020	South America	1
87	2/16/2020	South America	1
88	2/17/2020	South America	1
89	2/18/2020	South America	1
90	2/19/2020	South America	1
91	2/20/2020	South America	1
92	2/21/2020	South America	1
93	2/22/2020	South America	1

- 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.

Source code

```
df <- data.frame(iso_code, continent, date, new_cases, death_cases)
df$date<-as.Date(df$date, format="%m/%d/%Y")
k=readline(prompt="Enter the k date with format %Y/%m/%d:")
t=readline(prompt="Enter the t continent:")
datefilter = as.Date(k)
df <- df %>% filter(continent == t & date == datefilter)
print(df)
```



- Ta tạo một data frame df mới, hàm `as.Date` có tác dụng chuyển vector date trong df từ kiểu dữ liệu `char` sang `Date`.
- hàm `readline` để đọc dữ liệu từ bàn phím sau đó gán giá trị dữ liệu vừa nhập vào các biến k,t
- sử dụng biến datefilter gán giá trị k vừa được chuyển đổi sang kiểu dữ liệu `Date`. Sau đó dùng hàm `filter` để lọc dữ liệu các châu lục và ngày theo k và t cho trước

	iso_code	continent	date	new_cases	death_cases
1	AFG	Asia	2020-12-06	253	18
2	ARM	Asia	2020-12-06	978	17
3	AZE	Asia	2020-12-06	4356	39
4	BHR	Asia	2020-12-06	198	0
5	BGD	Asia	2020-12-06	1756	31
6	BTN	Asia	2020-12-06	4	NA
7	BRN	Asia	2020-12-06	0	0
8	KHM	Asia	2020-12-06	2	NA
9	CHN	Asia	2020-12-06	15	0
10	GEO	Asia	2020-12-06	4321	42
11	HKG	Asia	2020-12-06	95	0
12	IND	Asia	2020-12-06	32981	391
13	IDN	Asia	2020-12-06	6089	151
14	IRN	Asia	2020-12-06	11561	294
15	IRQ	Asia	2020-12-06	1680	21
16	ISR	Asia	2020-12-06	1080	8
17	JPN	Asia	2020-12-06	2038	31
18	JOR	Asia	2020-12-06	2576	46

- 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.

Source code

```
sub_con_data <- subset(con_data, duplicated(con_data[,2]) |  
                           duplicated(con_data[,2], fromLast=TRUE))  
sub_con_data <- sub_con_data[order(sub_con_data[,2]),]  
View(sub_con_data)
```

- Ta tái sử dụng lại dữ liệu từ biến `con_data` của câu i-5. Thông qua điều kiện `duplicated(con_data[,2]) | duplicated(con_data[,2], fromLast = TRUE)` sẽ giúp ta lấy được tất cả các hàng có dữ liệu trùng lặp với nhau. Ta đưa dữ liệu mới này vào biến `sub_con_data` sau đó sắp xếp lại theo thứ tự tăng dần để dễ nhận biết các nước có số lượng dữ liệu giống nhau.
- Vì có rất nhiều nước có lượng dữ liệu thu thập giống nhau nên bên dưới đây chỉ là ảnh minh họa cho một vài nước.



	iso_code	Observations
182	SPM	686
184	SSD	686
14	BDI	691
178	SLE	691
4	AIA	694
194	TCA	694
217	VGB	694
78	GNB	697
108	KNA	697
131	MLI	697

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

Source code

```
i17 <- function()
{
  temp <- cbind( unique(data$iso_code) , unique(data$location) )
  output_i17 <- temp[ nchar( unique(data$iso_code)) > 3 , ]
  View(output_i17)
}
```

- Ta sử dụng hàm `unique` để lấy các giá trị iso_code và location chỉ xuất hiện 1 lần trong data và dùng `cbind` gán vào biến temp. Cuối cùng gán các hàng có iso_code lớn hơn 3 dùng `nchar` để đặt điều kiện.

	iso_code	location
1	OWID_KOS	Kosovo
2	OWID_CYN	Northern Cyprus

- 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:

- 1) Tính giá trị nhỏ nhất, lớn nhất
- 2) Tính tứ phân vị thứ nhất(Q1), thứ hai(Q2), thứ ba(Q3)
- 3) Tính giá trị trung bình (Avg)
- 4) Tính giá trị độ lệch chuẩn (Std)
- 5) Đếm xem có bao nhiêu outliers, một quan sát mà giá trị của nó nằm trong khoảng sau:
 $IQR = Q3 - Q1$
 $outliers < Q1 - 1.5 * IQR$ hoặc $outliers > Q3 + 1.5 * IQR$
- 6) Lập bảng mô tả số liệu thống kê cho từng đất nước thuộc về nhóm:



Countries	Min	Q1	Q2	Q3	Max	Avg	Std	Outlier
ctr_i	?	?	?	?	?	?	?	?

Source code

```
ii <- function(subdata , col)
{
  sum <- summary(subdata[,col])
  min <- as.numeric(sum[1])
  max <- as.numeric(sum[6])
  Q1 <- as.numeric(sum[2])
  Q2 <- as.numeric(sum[3])
  Q3 <- as.numeric(sum[5])
  avg <- as.numeric(sum[4])
  std <- sd(subdata[,col],na.rm = TRUE)
  outlier <- 0
  for(i in 1:nrow(subdata))
  {
    if(is.na(subdata[i,col])) next
    if(subdata[i,col] < (Q1 - (1.5*(Q3 - Q1))) ||
       subdata[i,col] > (Q3 + (1.5*(Q3 - Q1))))
    {
      outlier <- outlier + 1
    }
    else {}
  }
  return(as.numeric(c(min,Q1,Q2,Q3,max,avg,std,outlier)))
}
dfCountry <- data.frame(Countries = c("Indonesia",
                                         "Japan", "Vietnam"))
id_data <- subset(data, location == "Indonesia")
jp_data <- subset(data, location == "Japan")
vn_data <- subset(data, location == "Vietnam")

tmp <- data.frame(rbind(ii(id_data,5),ii(jp_data,5),ii(vn_data,5)))
colnames(tmp) <- c("Min", "Q1", "Q2", "Q3",
                   "Max", "Avg", "Std", "Outlier")
dfCountry_case <- cbind(dfCountry,tmp)
View(dfCountry_case)

tmp <- data.frame(rbind(ii(id_data,6),ii(jp_data,6),ii(vn_data,6)))
colnames(tmp) <- c("Min", "Q1", "Q2", "Q3",
                   "Max", "Avg", "Std", "Outlier")
dfCountry_death <- cbind(dfCountry,tmp)
View(dfCountry_death)
```

- Ta thực hiện viết hàm `ii()` để nhận vào data frame của quốc gia và cột tương ứng cần tính. Hàm này sẽ trả về 1 vector có các giá trị tương ứng theo thứ tự là `Min`, `Q1`, `Q2`, `Q3`, `Max`, `Average`, `StandardDeviation`, `Outlier`. Với đó, ta sẽ làm đc cùng lúc toàn bộ các câu từ 1 đến 6 của phần ii.
- Bên trong hàm `ii`, ta thực hiện hàm `summary()` cho cột thứ `col` nhập vào của dữ liệu đưa vào biến `subdata`. Hàm này sẽ cho ta biết được các giá trị nhỏ nhất, lớn nhất, tứ phân vị và giá trị trung bình của dữ liệu nhập vào. Ta chỉ việc trích xuất các giá trị này vào các biến `min`, `max`, `Q1`, `Q2`, `Q3`, `avg` tương ứng.
- Dối với độ lệch chuẩn `std` ta thực hiện hàm `sd()` với `na.rm = TRUE` để tính độ lệch chuẩn của dữ liệu trong khi bỏ qua các số liệu NA.



- Đối với *outlier*, ta khởi tạo giá trị 0. Sau đó thực hiện một vòng lặp for chạy xuyên suốt các hàng của dữ liệu nhập vào. Nếu dữ liệu không phải là NA và thỏa mãn điều kiện $outliers < Q1 - 1.5 * IQR$ hoặc $outliers > Q3 + 1.5 * IQR$ với $IQR = Q3 - Q1$ thì ta tăng giá trị của *outlier* lên 1.
- Ghép các giá trị này vào 1 vector để hàm trả về.
- Bên ngoài hàm, ta thực hiện việc tạo 1 vector *dfCountry* chứ giá trị là tên của 3 nước thuộc về nhóm cần tính số liệu, sau đó tạo ra các data frame riêng để chứa dữ liệu của từng nước thông qua hàm *subset()*.
- Ta sử dụng hàm vừa làm ban nãy để tính tất cả các số liệu cần thiết cho cột *new_case* của cả 3 quốc gia thuộc về nhóm cần tính số liệu và ghép lại thành 1 data frame với tên biến *tmp*. Sau đó đặt tên từng cột theo các số liệu tương ứng.
- Ta ghép cột *dfCountry* với data frame vừa tạo để hoàn chỉnh bảng số liệu và xuất ra màn hình.
- Đối với cột *new_death* cũng thực hiện giống như trên chỉ cần thay giá trị thành số thứ tự của cột *new_death* vào hàm *ii*

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

Hình 1: Số liệu thống kê cho ca nhiễm của từng đất nước

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

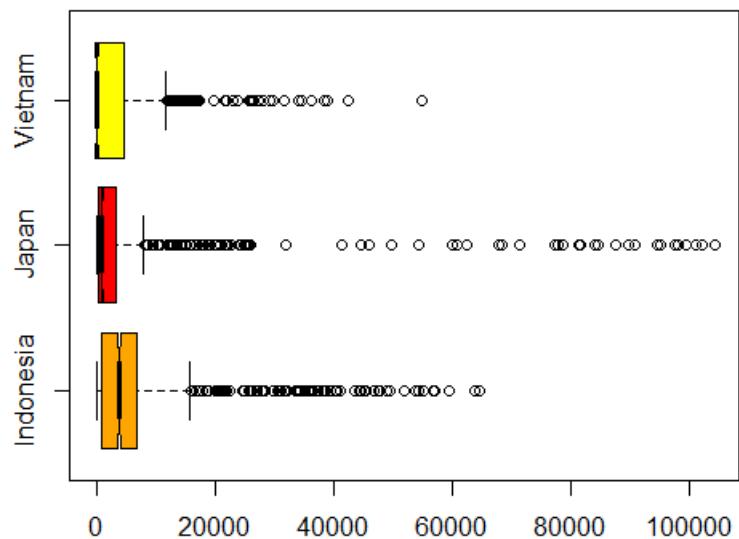
Hình 2: Số liệu thống kê cho tử vong của từng đất nước

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

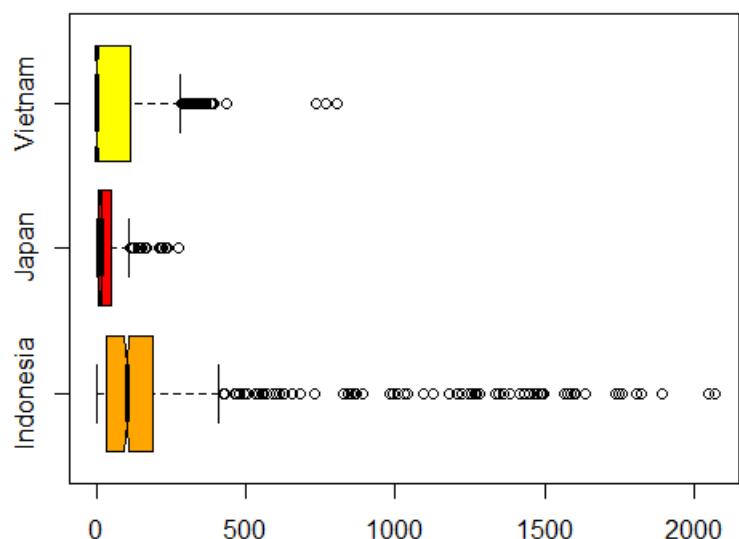
Source code

```
boxplot(id_data[,5], jp_data[,5], vn_data[,5],
        main = "Plotbox_for_new_cases",
        at = c(1,2,3),
        names = c("Indonesia", "Japan", "Vietnam"),
        col = c("orange", "red", "yellow"),
        border = "black",
        horizontal = TRUE,
        notch = TRUE
)
boxplot(id_data[,6], jp_data[,6], vn_data[,6],
        main = "Plotbox_for_new_deaths",
        at = c(1,2,3),
        names = c("Indonesia", "Japan", "Vietnam"),
        col = c("orange", "red", "yellow"),
        border = "black",
        horizontal = TRUE,
        notch = TRUE
)
```

Plotbox for new cases



Plotbox for new deaths



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:

Source code

```
indo <- data[data[,1]=="IDN",]
vn <- data[data[,1]=="VNM",]
jp <- data[data[,1]=="JPN",]
```

- Tạo data của mỗi quốc gia indonesia, vietnam và japan.
- 1) Có bao nhiêu ngày có số lần dữ liệu không được báo cáo mới

Source code

```
nnreportcindo <- indo[ is.na(indo[,5]) | indo[,5]==0,]
indo1.1 <- nrow(nnreportcindo)
view(indo1.1)
nnreportdindo <- indo[ is.na(indo[,6]) | indo[,6]==0,]
indo1.2 <- nrow(nnreportdindo)
view(indo1.2)

nnreportcvn <- vn[ is.na(vn[,5]) | vn[,5]==0,]
vn1.1 <- nrow(nnreportcvn)
view(vn1.1)
nnreportdvn <- vn[ is.na(vn[,6]) | vn[,6]==0,]
vn1.2 <- nrow(nnreportdvn)
view(vn1.1)

nnreportcj <- jp[ is.na(jp[,5]) | jp[,5]==0,]
jp1.1 <- nrow(nnreportcj)
view(jp1.1)
nnreportdjp <- jp[ is.na(jp[,6]) | jp[,6]==0,]
jp1.2 <- nrow(nnreportdjp)
view(jp1.1)
)
```

- Tìm các dữ liệu không được báo cáo mới bằng cách đặt điều kiện bằng 0 và hàm `is.na` gán vào biến. Sau đó dùng hàm `nrow` để đếm số hàng của biến.

	x			x			x	
1	8			1	139		1	11
	x			x			x	
1	15			1	489		1	77

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

Source code

```
newreportindo <- indo[ !is.na(indo[,5]) & !is.na(indo[,6])
& indo[,5]!=0 & indo[,6]!=0,]
newreportvn <- vn[ !is.na(vn[,5]) & !is.na(vn[,6])
& vn[,5]!=0 & vn[,6]!=0,]
newreportjp <- jp[ !is.na(jp[,5]) & !is.na(jp[,6])
& jp[,5]!=0 & jp[,6]!=0,]
newreport <- rbind(newreportindo, newreportvn, newreportjp)

indominc <- newreportindo[ newreportindo[,5]==min( newreportindo[,5]),4]
indo2.1 <- length(indominc)
view(indo2.1)
indomind <- newreportindo[ newreportindo[,6]==min( newreportindo[,6]),4]
indo2.2 <- length(indomind)
view(indo2.2)
```

```

vnminc <- newreportvn [ newreportvn[,5]==min( newreportvn[,5]),4]
vn2.1 <- length(vnminc)
view(vn2.1)
vnmind <- newreportvn [ newreportvn[,6]==min( newreportvn[,6]),4]
vn2.2 <- length(vnmind)
view(vn2.2)

jpminc <- newreportjp [ newreportjp[,5]==min( newreportjp[,5]),4]
jp2.1 <- length(jpminc)
view(jp2.1)
jpmind <- newreportjp [ newreportjp[,6]==min( newreportjp[,6]),4]
jp2.2 <- length(jpmind)
view(jp2.2)

```

- Tương tự tìm các dữ liệu được báo cáo mới gán vào biến. Sau đó dùng hàm *min* để tìm giá trị thấp nhất và đặt điều kiện trong matrix để tìm các ngày bằng giá trị đó. Cuối cùng dùng hàm *length* để tìm số ngày có giá trị thấp nhất được báo cáo mới.

1	1			1	2		1	1
	x				x		x	
1	5			1	26		1	54

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

Source code

```

indomaxc <- newreportindo [ newreportindo[,5]==max( newreportindo[,5]),4]
indo3.1 <- length(indomaxc)
view(indo3.1)
indomaxd <- newreportindo [ newreportindo[,6]==max( newreportindo[,6]),4]
indo3.2 <- length(indomaxd)
view(indo3.2)

vnmaxc <- newreportvn [ newreportvn[,5]==max( newreportvn[,5]),4]
vn3.1 <- length(vnmaxc)
view(vn3.1)
vnmaxd <- newreportvn [ newreportvn[,6]==max( newreportvn[,6]),4]
vn3.2 <- length(vnmaxd)
view(vn3.2)

jpmaxc <- newreportjp [ newreportjp[,5]==max( newreportjp[,5]),4]
jp3.1 <- length(jpmaxc)
view(jp3.1)
jpmaxd <- newreportjp [ newreportjp[,6]==max( newreportjp[,6]),4]
jp3.2 <- length(jpmaxd)
view(jp3.2)

```



	x		x		x	
1	1		1	1	1	1
	x		x		x	
1	1		1	1	1	1

4) Thể hiện bảng số liệu như sau:

Không được báo cáo mới:

Countries ctr_i	Infections value	Deaths value
--------------------	---------------------	-----------------

Báo cáo mới:

Countries ctr_i	Infections value	Deaths value
--------------------	---------------------	-----------------

Source code

```

nnreport <- rbind(indo[is.na(indo[,5]) | is.na(indo[,6])
| indo[,6]==0 | indo[,5]==0,],
vn[is.na(vn[,5]) | is.na(vn[,6])
| vn[,5]==0 | vn[,6]==0,],
jp[is.na(jp[,5]) | is.na(jp[,6])
| jp[,5]==0 | jp[,6]==0,])
iii4.1 <- nnreport[,-c(1,2,4)]
colnames(iii4.1) <- c("Countries","Infections_value","Deaths_value")
view(iii4.1)
iii4.2 <- newreport[,-c(1,2,4)]
colnames(iii4.2) <- c("Countries","Infections_value","Deaths_value")
view(iii4.2)

```

- Đặt điều kiện trong matrix để tìm các dữ liệu không được báo cáo mới của 3 nước và dùng `rbind` để gán các dữ liệu đó vào biến nnreport. Sau đó lấy các cột 1, 2, 4 của data nnreport và gán vào biến mới. Cuối cùng dùng hàm `colnames` để đổi tên cho cột.

	Countries	Infections value	Deaths value
69098	Indonesia	21	0
69099	Indonesia	17	0
69100	Indonesia	38	0
69634	Indonesia	0	0
69749	Indonesia	0	0
69804	Indonesia	NA	158
158996	Vietnam	2	NA
158997	Vietnam	0	NA
158998	Vietnam	0	NA
158999	Vietnam	0	NA
159000	Vietnam	0	NA
159001	Vietnam	0	NA
159002	Vietnam	0	NA
	Countries	Infections value	Deaths value
159746	Vietnam	26487	96
159747	Vietnam	27311	78
159748	Vietnam	26379	84
159749	Vietnam	29413	91
159750	Vietnam	31814	85
159751	Vietnam	34737	66
159752	Vietnam	36200	90
159753	Vietnam	42439	80
159754	Vietnam	54830	65
75647	Japan	5	1
75652	Japan	12	1
75659	Japan	13	1
75661	Japan	26	1



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

Source code

```
indo5_case <- 0
for (i in 1:nrow(indo)){
  if (is.na(indo[i,5])){
    indo5_case <- indo5_case + 1
    break
  }
}
view(indo5_case)
indo5_death <- 0
for (i in 1:nrow(indo)){
  if (is.na(indo[i,6])){
    indo5_death <- indo5_death + 1
    break
  }
}
view(indo5_death)

vn5_case <- 0
for (i in 1:nrow(vn)){
  if (is.na(vn[i,5])){
    vn5_case <- vn5_case + 1
    break
  }
}
view(vn5_case)
vn5_death <- 0
for (i in 1:nrow(vn)){
  if (is.na(vn[i,6])){
    vn5_death <- vn5_death + 1
    break
  }
}
view(vn5_death)

jp5_case <- 0
for (i in 1:nrow(jp)){
  if (is.na(jp[i,5])){
    jp5_case <- jp5_case + 1
    break
  }
}
view(jp5_case)
jp5_death <- 0
for (i in 1:nrow(jp)){
  if (is.na(jp[i,6])){
    jp5_death <- jp5_death + 1
    break
  }
}
view(jp5_death)
```

- Ta chạy vòng for và dùng hàm `is.na` để kiểm tra giá trị của cột 5,6 của data mỗi nước. Nếu `TRUE` thì kết quả là 1, ngược lại kết quả bằng 0.

	▲	X	▼		▲	X	▼		▲	X	▼	
	1	1			1	0			1	1		
		▲	X	▼		▲	X	▼		▲	X	▼
	1	1			1	1			1	1		

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

Source code

```

indo6_case <- 0
temp <- 0
for (i in 1:nrow(indo)){
  if (is.na(indo[i,5])){
    temp <- temp + 1
    if (temp>indo6_case) {
      indo6_case <- temp
    }
  }
  else{
    temp <- 0
  }
}
view(indo6_case)
indo6_death <- 0
temp <- 0
for (i in 1:nrow(indo)){
  if (is.na(indo[i,6])){
    temp <- temp + 1
    if (temp>indo6_death) {
      indo6_death <- temp
    }
  }
  else{
    temp <- 0
  }
}
view(indo6_death)

vn6_case <- 0
temp <- 0
for (i in 1:nrow(vn)){
  if (is.na(vn[i,5])){
    temp <- temp + 1
    if (temp>vn6_case) {
      vn6_case <- temp
    }
  }
  else{
    temp <- 0
  }
}
view(vn6_case)
vn6_death <- 0
temp <- 0

```

```

for (i in 1:nrow(vn)){
  if (is.na(vn[i,6])){
    temp <- temp + 1
    if (temp>vn6_death) {
      vn6_death <- temp
    }
  } else{
    temp <- 0
  }
}
view(vn6_death)

jp6_case <- 0
temp <- 0
for (i in 1:nrow(jp)){
  if (is.na(jp[i,5])){
    temp <- temp + 1
    if (temp>jp6_case) {
      jp6_case <- temp
    }
  } else{
    temp <- 0
  }
}
view(jp6_case)
jp6_death <- 0
temp <- 0
for (i in 1:nrow(jp)){
  if (is.na(jp[i,6])){
    temp <- temp + 1
    if (temp>jp6_death) {
      jp6_death <- temp
    }
  } else{
    temp <- 0
  }
}
view(jp6_death)

```

- Ta chạy vòng for và dùng hàm `is.na` để kiểm tra dữ liệu cột 5,6 của mỗi nước, nếu `TRUE` biến temp tăng và so sánh vs số ngày dài nhất liên tiếp, nếu `FALSE` reset biến temp.

	x		x		x	
1	1		1	0	1	1
	x		x		x	
1	9		1	190	1	22

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

Source code

```
indo7 <- 0
for (i in 1:nrow(nnreportcindo)){
  if (nnreportcindo[i,5]==0){
    indo7 <- indo7 + 1
    break
  }
}
view(indo7)

vn7 <- 0
for (i in 1:nrow(nnreportcvn)){
  if (nnreportcvn[i,5]==0){
    vn7 <- vn7 + 1
    break
  }
}
view(vn7)

jp7 <- 0
for (i in 1:nrow(nnreportcj7)){
  if (nnreportcj7[i,5]==0){
    jp7 <- jp7 + 1
    break
  }
}
view(jp7)
```

▲	x	▼	▲	x	▼	▲	x	▼
1	1		1	1		1	1	

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

Source code

```
indo8 <- 0
temp <- 0
for (i in 1:nrow(indo)){
  if (!is.na(indo[i,5]) & indo[i,5]==0){
    temp <- 0
  }
  else{
    temp <- temp + 1
    if (temp>indo8) {
      indo8 <- temp
    }
  }
}
view(indo8)

vn8 <- 0
temp <- 0
for (i in 1:nrow(vn)){
  if (!is.na(vn[i,5]) & vn[i,5]==0){
```

```

        temp <- 0
    }
else{
    temp <- temp + 1
    if (temp>vn8) {
        vn8 <- temp
    }
}
view(vn8)

jp8 <- 0
temp <- 0
for (i in 1:nrow(jp)){
    if (!is.na(jp[i,5]) & jp[i,5]==0){
        temp <- 0
    }
else{
        temp <- temp + 1
        if (temp>jp8) {
            jp8 <- temp
        }
    }
}
view(jp8)

```

	x			x		x
1	538			1	167	1

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

Prep for iv1-2

```

freg <- data[!duplicated(data[,c('location')]) ,]
coun_num <- nrow(freg)
x_con <- c("Africa", "Asia", "Europe",
          "North_America", "Oceania", "South_America")

```

- Ta dùng câu lệnh `data[!duplicated(data[,c('location')]) ,]` để loại bỏ các dòng có giá trị lặp ở cột `Location` để cho ta 1 data mới với mỗi nước chỉ xuất hiện duy nhất 1 lần. Ta bỏ data mới này vào biến `freg`.
- Dùng hàm `nrow()` để đếm tổng số đất nước và bỏ vào biến `coun_num`.
- Tiến hành khởi tạo trước cột x cho biểu đồ với giá trị là tên của từng châu lục.

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

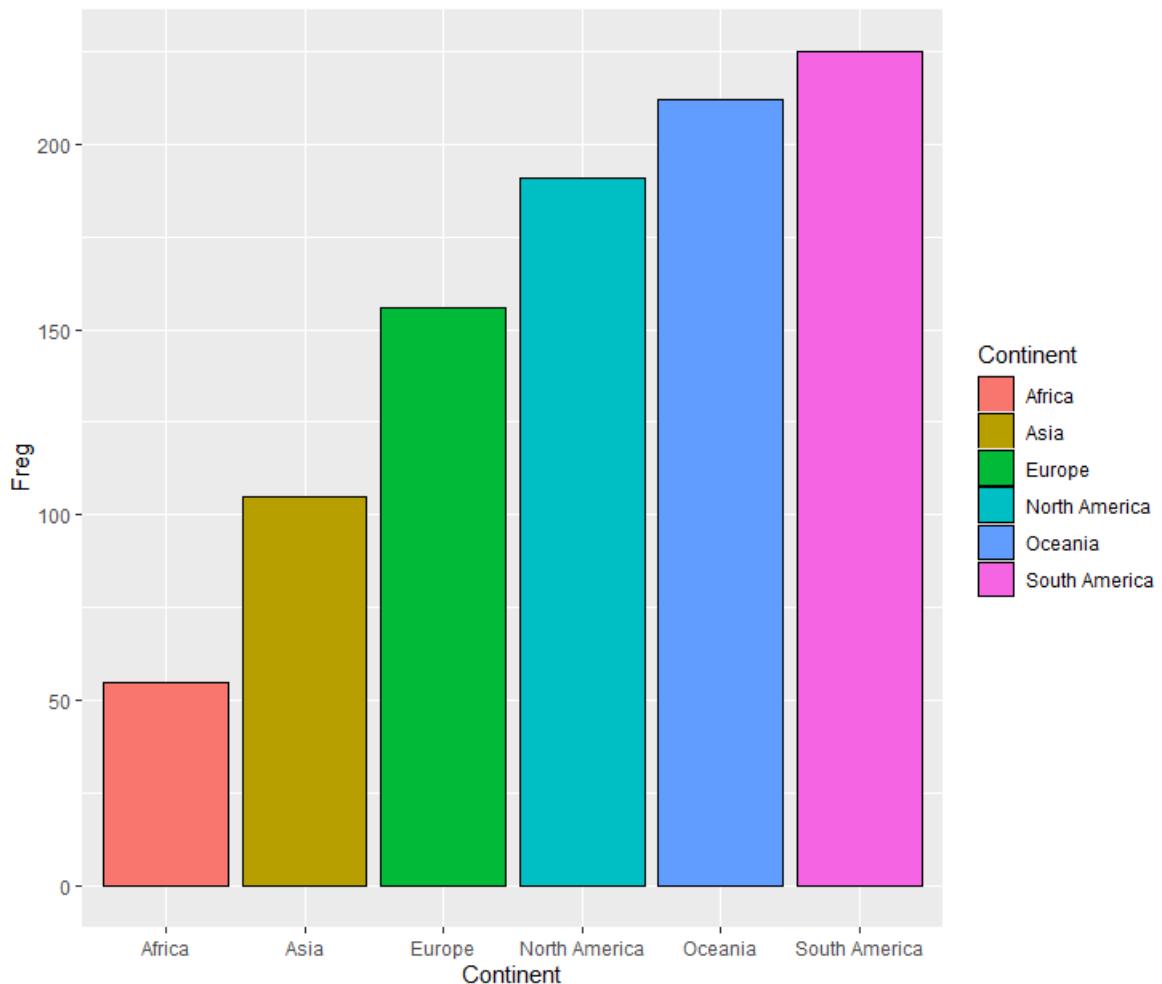
Source code

```

y_con <- cumsum(as.numeric(table(freg$continent)))
df_iv1 <- data.frame(Continent = x_con, Freq = y_con)
ggplot(data = df_iv1, aes(x = Continent, y = Freq, fill = Continent)) +
  geom_bar(stat = "Identity", colour = "black")

```

- Sử dụng hàm `table()` để đếm số lần xuất hiện của các châu lục. Vì mỗi quốc gia trong data vừa xử lý chỉ xuất hiện duy nhất một lần nên việc ta vừa làm chính là đếm số quốc gia cho từng châu lục
- Ta tiếp tục dùng hàm `cumsum()` để biến đổi vector này thành dạng tích lũy. Chuyển dữ liệu này thành dạng số và bỏ vào `vector` cho cột y của biểu đồ.

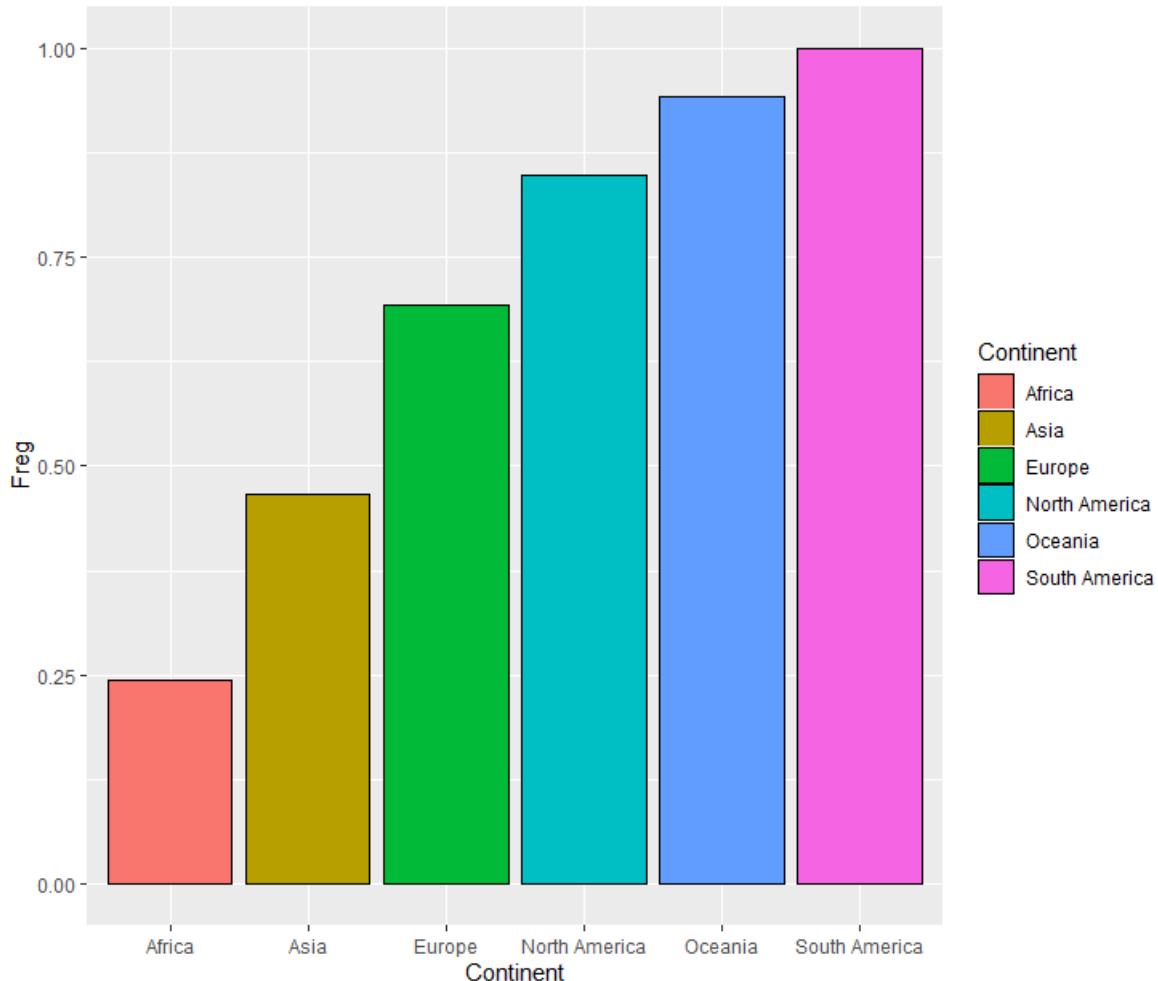


2) Vẽ biểu đồ tần số tương đối quốc gia cho các châu lục

Source code

```
y_con <- y_con/coun_num
df_iv2 <- data.frame(Continent = x_con, Freg = y_con)
ggplot(data = df_iv2, aes(x = Continent, y = Freg, fill = Continent))
  + geom_bar(stat = "Identity", colour = "black")
```

- Biến đổi `vector` cho cột y ở câu trước thành tần số tích lũy bằng cách chia nó cho giá trị cuối cùng của `vector` hoặc chia cho tổng số quốc gia và tiến hành vẽ biểu đồ như câu trên.



Prep for iv3-4

```
x_coun <- c("Indonesia", "Japan", "Vietnam")
id_data <- subset(data, location == "Indonesia")
jp_data <- subset(data, location == "Japan")
vn_data <- subset(data, location == "Vietnam")

id_date <- tail(id_data[order(as.Date(id_data$date,
format="%d/%m/%Y")),], n=7)
jp_date <- tail(jp_data[order(as.Date(jp_data$date,
format="%d/%m/%Y")),], n=7)
vn_date <- tail(vn_data[order(as.Date(vn_data$date,
format="%d/%m/%Y")),], n=7)

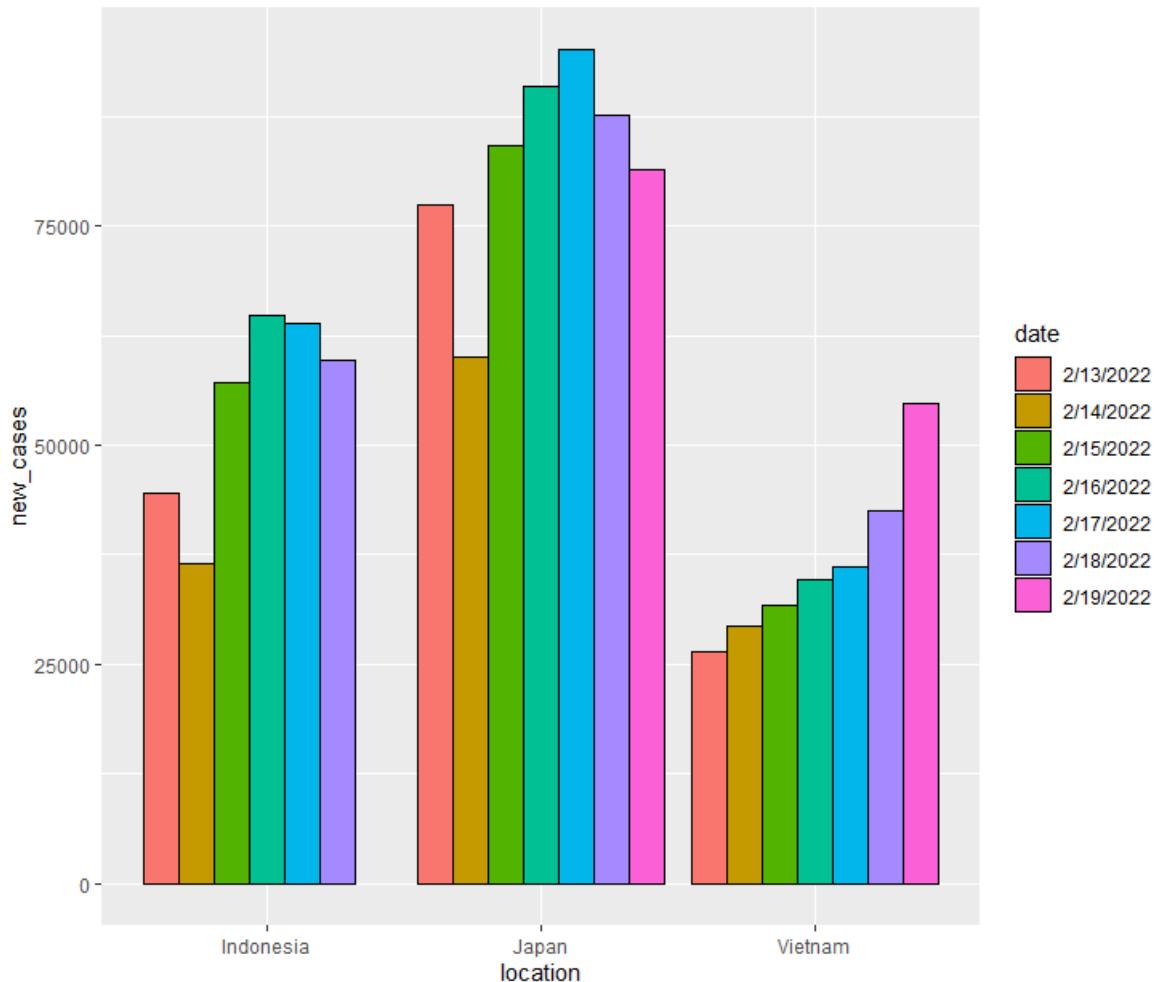
dfDate <- data.frame(rbind(id_date, jp_date, vn_date))
```

- Thực hiện tạo 1 vector `x_coun` chứa tên của các quốc gia thuộc về nhóm cần tính số liệu.
- Lấy ra dữ liệu của từng quốc gia rồi sử dụng hàm `order(as.Date())` với format là `dd-mm-yyyy` để sắp xếp dữ liệu của từng quốc gia theo ngày tăng dần và dùng hàm `tail()` với `n = 7` để lấy ra 7 ngày cuối của năm cuối cùng.
- Tiến hành ghép tất cả dữ liệu vừa xử lý của 3 quốc gia lại với nhau. Qua đó ta có thể dễ dàng vẽ biểu đồ nhiễm bệnh hay tử vong của các quốc gia khi chỉ cần thay giá trị của trục y trong câu lệnh `ggplot()` thành cột dữ liệu tương ứng.

- 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

Source code

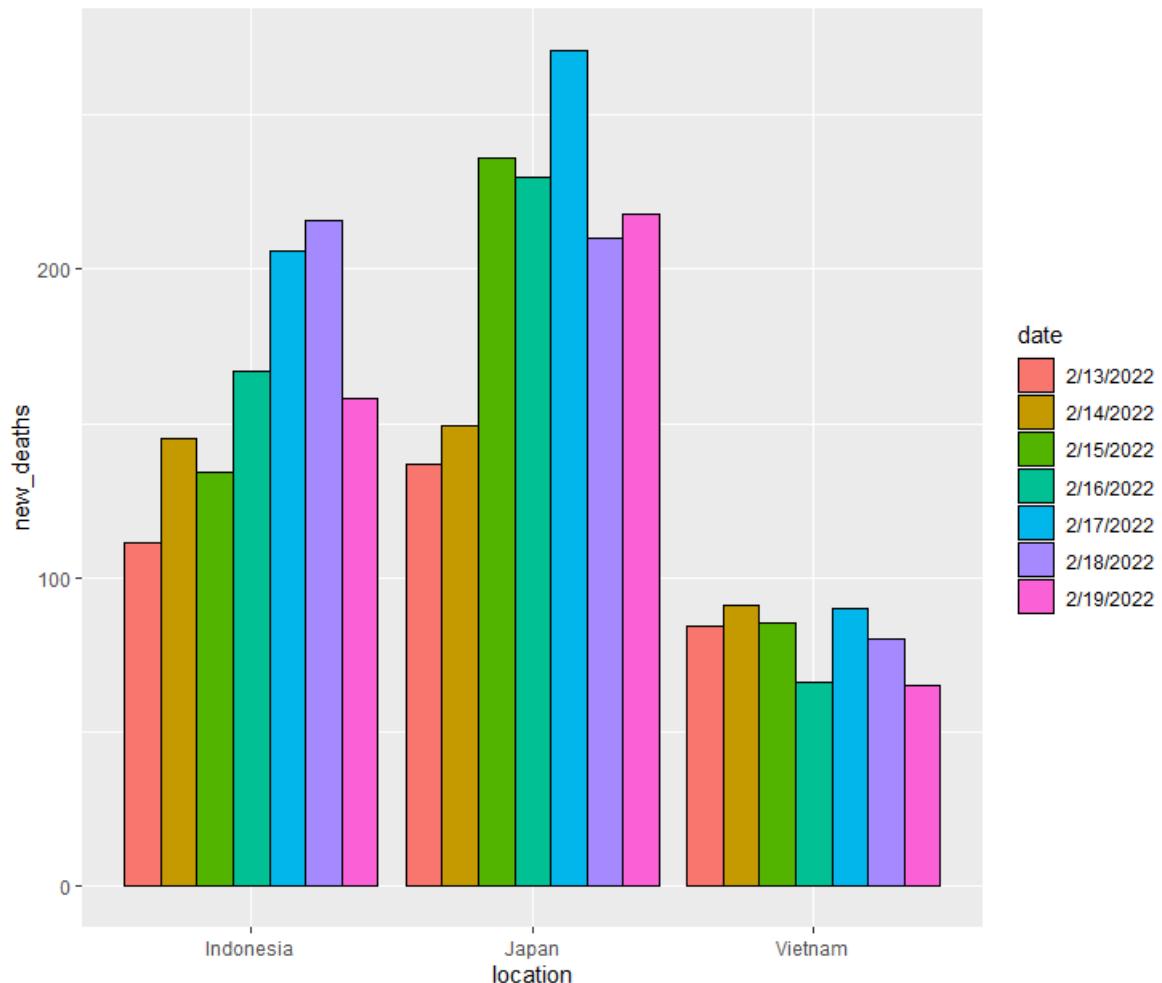
```
ggplot(data = dfDate, aes(x = location, y = new_cases, fill = date))  
+ geom_bar(stat = "Identity", colour = "black", position = "dodge")
```



- 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

Source code

```
ggplot(data = dfDate, aes(x = location, y = new_deaths, fill = date))  
+ geom_bar(stat = "Identity", colour = "black", position = "dodge")
```



Function and Prep for iv5-6

```
iv_5_6 <- function(data, name, col)
{
  subdata <- subset(data, location==name)
  sum <- summary(subdata[,col])
  Q1 <- as.numeric(sum[2])
  Q3 <- as.numeric(sum[5])
  outlier <- 0
  for(i in 1:nrow(subdata))
  {
    if(is.na(subdata[i,col])) next
    if(subdata[i,col] < (Q1 - (1.5*(Q3 - Q1)))
       || subdata[i,col] > (Q3 + (1.5*(Q3 - Q1))))
    {
      outlier <- outlier + 1
    }
  }
  return(c(name, outlier))
}

coun_name <- unique(data[,3])
```

- Hàm `iv_5_6()` có tác dụng nhận vào toàn bộ dữ liệu data, tên của quốc gia và cột cần tính outlier. Hàm sẽ trả về 1 `vector` với 2 giá trị là tên của quốc gia và số outlier của quốc gia đó.



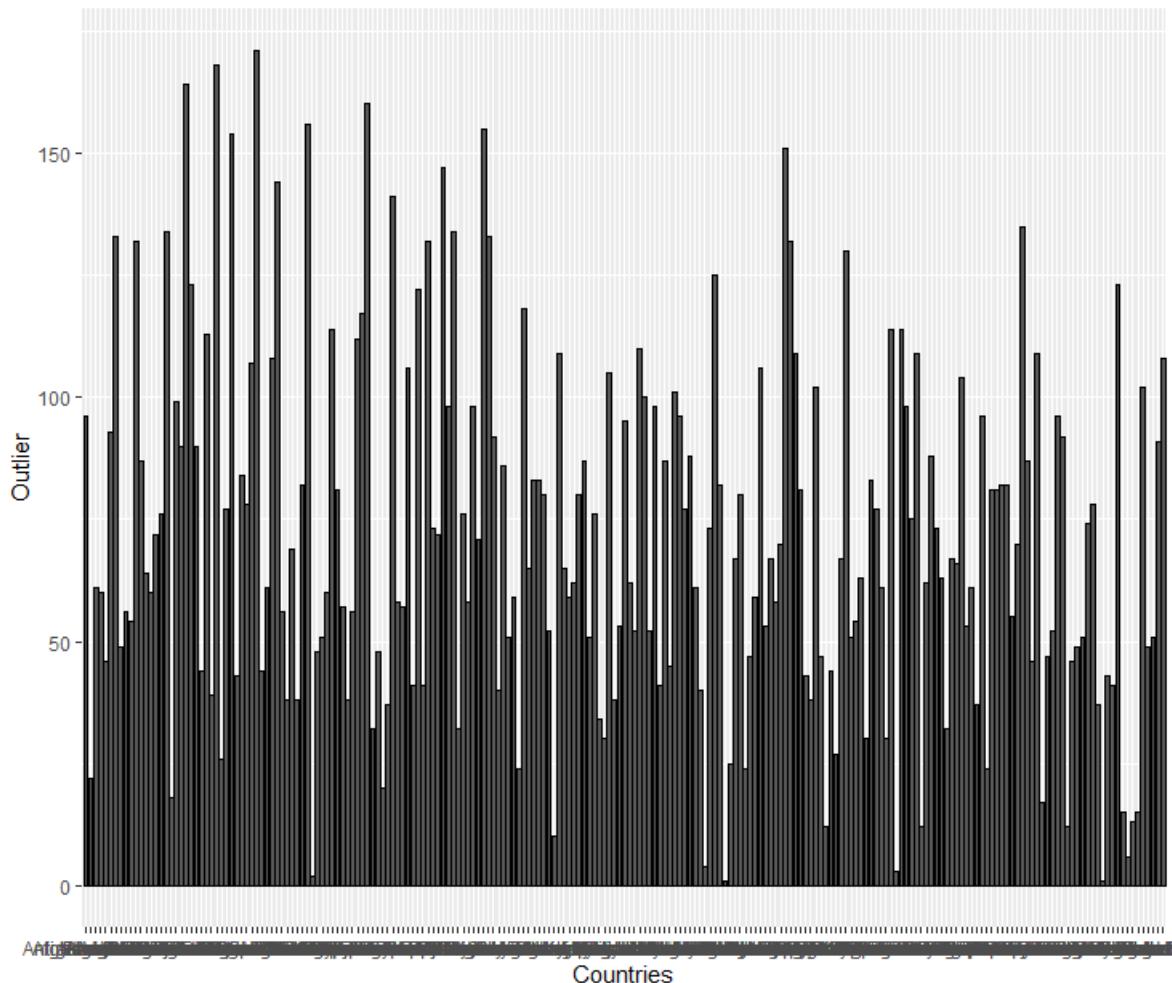
- Bên trong hàm sẽ lấy ra những dòng dữ liệu từ `data` của đất nước cần tính trong biến `name` và bỏ tất cả vào một `subdata` mới. Ta lại tiếp tục dùng hàm `summary()` để lấy tứ phân vị thứ nhất `Q1` và thứ ba `Q3`. Ta thực hiện chạy 1 vòng lặp for qua tất cả các dòng trong `subdata` để bắt đầu việc tính outlier.
- Ta khởi tạo giá trị của biến `outlier` bằng 0. Với mỗi hàng ta đi qua, nếu dữ liệu không phải là NA và thỏa mãn điều kiện $outliers < Q1 - 1.5 * IQR$ hoặc $outliers > Q3 + 1.5 * IQR$ với $IQR = Q3 - Q1$ thì ta tăng giá trị của `outlier` lên 1.
- Bên ngoài hàm, ta thực hiện việc lấy danh sách tên các quốc gia bằng hàm `unique()`.

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

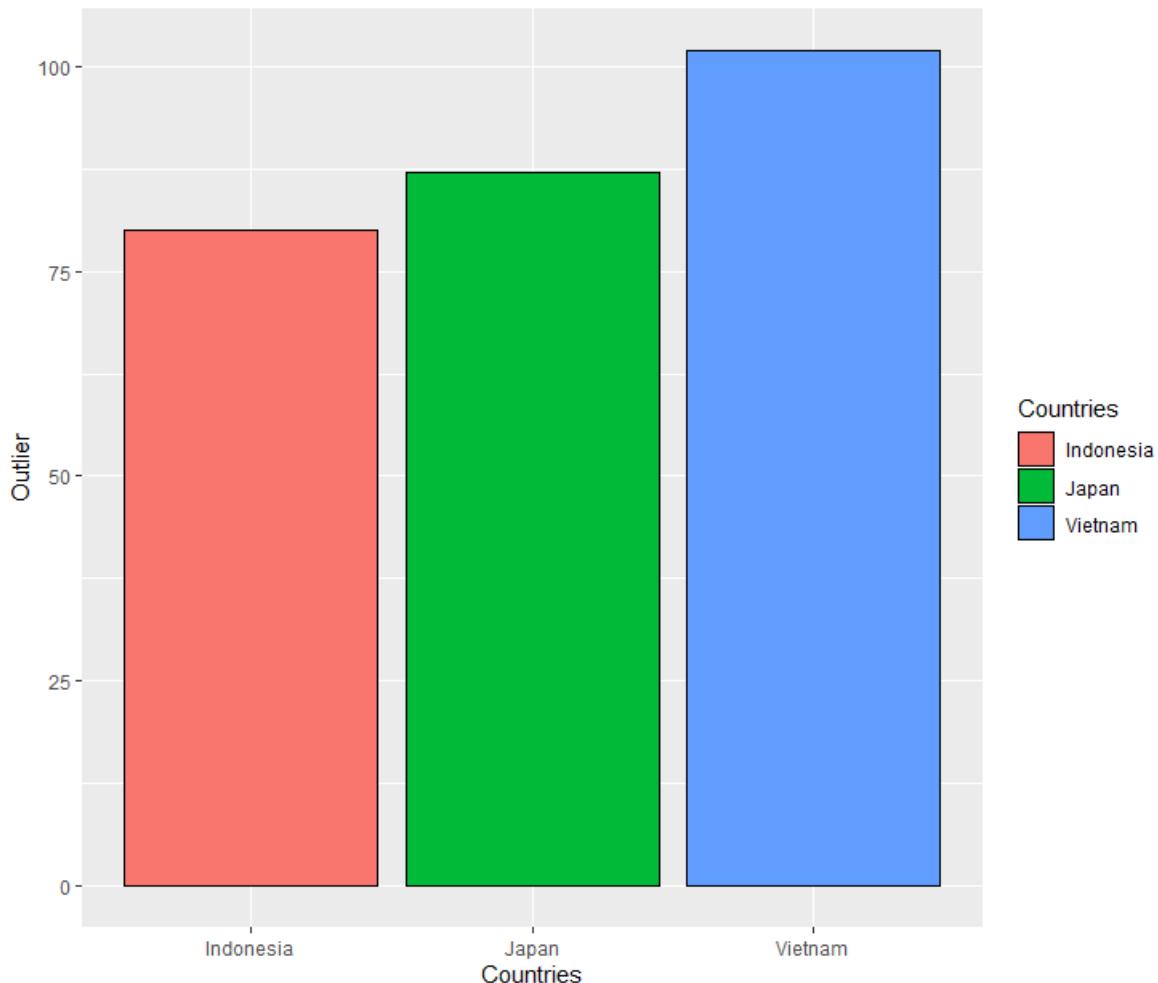
Source code

```
dfOutlier <- data.frame(Countries = c("tmp"), Outlier = c(0))
for(i in 1:length(coun_name))
{
  dfOutlier <- rbind(dfOutlier, iv_5_6(data, coun_name[i], 5))
}
dfOutlier[, 2] <- as.numeric(dfOutlier[, 2])
dfOutlier <- subset(dfOutlier, Outlier != 0)
view_dfOutlier <- data.frame(rbind(subset(dfOutlier,
                                             Countries == "Indonesia"),
                                     subset(dfOutlier,
                                             Countries == "Japan"),
                                     subset(dfOutlier,
                                             Countries == "Vietnam")))
ggplot(data = dfOutlier, aes(x = Countries, y = Outlier))
  + geom_bar(stat = "Identity", colour = "black")
ggplot(data = view_dfOutlier,
       aes(x = Countries, y = Outlier, fill = Countries))
  + geom_bar(stat = "Identity", colour = "black")
```

- Ta tiến hành tạo 1 data frame mới tên là `dfOutlier` với 2 cột là `Countries` và `Outlier`. Khởi tạo giá trị cho dòng đầu tiên của data frame này là `tmp` và `0` tương ứng với 2 cột.
- Thực hiện 1 vòng lặp for chạy qua toàn bộ từng phần tử của vector `coun_name`. Với mỗi lần lặp, ta sẽ thực hiện hàm `iv_5_6()` để tính outlier cho ca nhiễm của mỗi đất nước trong vector `coun_name` và gắn vector trả về của hàm vào data frame `dfOutlier`. Như vậy sau khi chạy hết vòng for thì `dfOutlier` đã chứa toàn bộ tên đất nước cũng như số outlier của từng đất nước tương ứng.
- Ta tiếp tục biến đổi cột `Outlier` của `dfOutlier` thành số để có thể vẽ được biểu đồ đồng thời loại các dòng mà tại quốc gia đó có số outlier là 0. Việc này sẽ giúp ta loại được cả dòng đầu tạm thời của data frame ta đặt khi nãy.
- Tiến hành vẽ biểu đồ cột dựa trên dữ liệu có được bằng hàm `ggplot()`.



- Ta dễ dàng nhận thấy vì có quá nhiều đất nước nên việc hiển thị toàn bộ dữ liệu trên biểu đồ sẽ rất khó nhìn. Vậy nên ta sẽ rút gọn lại chỉ với 3 nước thuộc về nhóm cần tính số liệu bằng việc đưa dữ liệu outlier của 3 nước này vào một biến `view_dfOutlier` và vẽ biểu đồ cho data frame này.

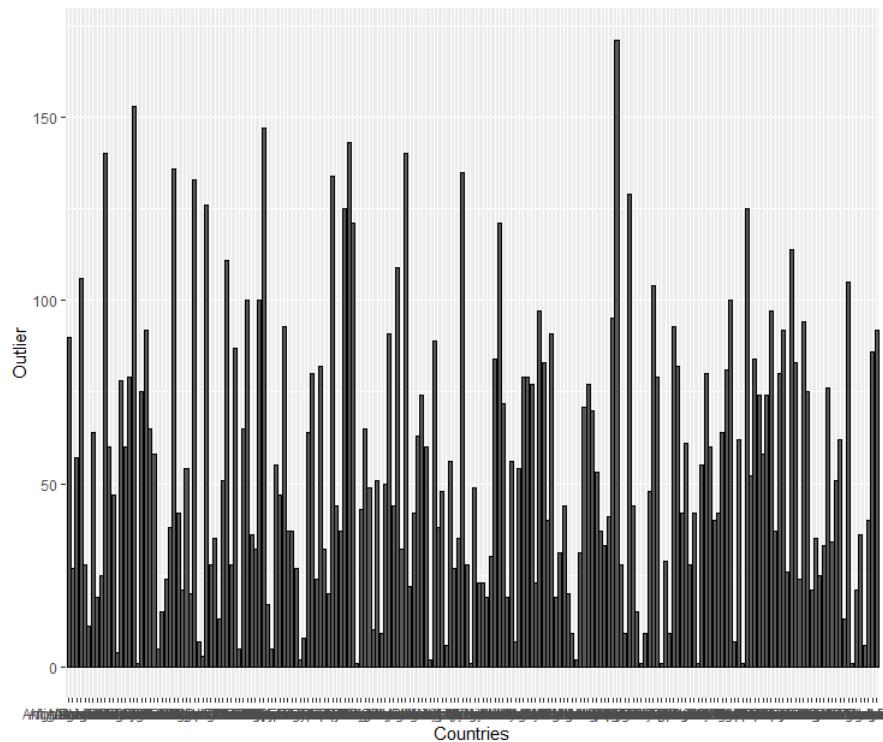


6) Vẽ biểu đồ phô đất nước xuất hiện outliers cho tử vong

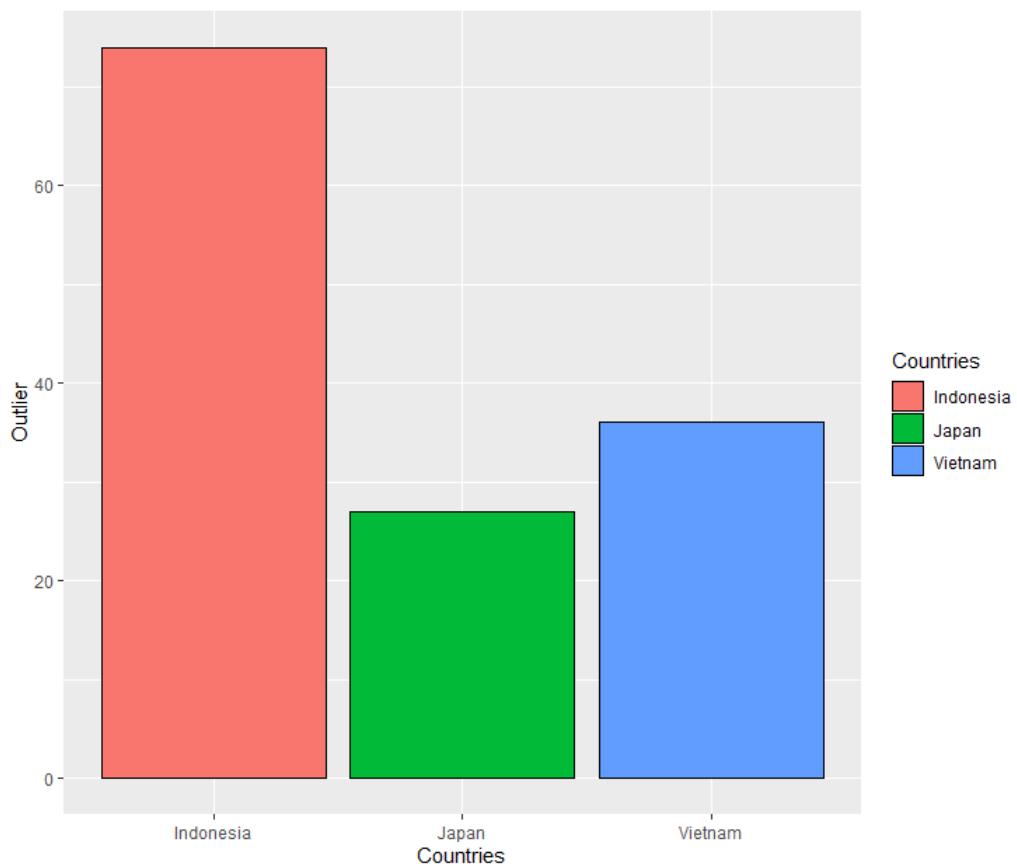
Source code

```
dfOutlier <- data.frame(Countries = c("tmp"), Outlier = c(0))
for(i in 1:length(coun_name))
{
  dfOutlier <- rbind(dfOutlier, iv_5_6(data, coun_name[i], 6))
}
dfOutlier[,2] <- as.numeric(dfOutlier[,2])
dfOutlier <- subset(dfOutlier, Outlier != 0)
view_dfOutlier <- data.frame(rbind(subset(dfOutlier,
                                             Countries == "Indonesia"),
                                     subset(dfOutlier,
                                             Countries == "Japan"),
                                     subset(dfOutlier,
                                             Countries == "Vietnam")))
ggplot(data = dfOutlier, aes(x = Countries, y = Outlier))
  + geom_bar(stat = "Identity", colour = "black")
ggplot(data = view_dfOutlier, aes(x = Countries, y = Outlier,
                                   fill = Countries))
  + geom_bar(stat = "Identity", colour = "black")
```

- Ta cũng tiến hành tạo data frame và vẽ biểu đồ tương tự như câu *iv - 5* nhưng là với outlier cho tử vong của mỗi đất nước bằng hàm *ggplot()*.



- Ta cũng dễ dàng nhận thấy vì có quá nhiều đất nước nên việc hiển thị toàn bộ dữ liệu trên biểu đồ sẽ rất khó nhìn. Vậy nên ta sẽ rút gọn lại chỉ với 3 nước thuộc về nhóm cần tính số liệu và vẽ biểu đồ.





Function and data for v vi vii viii

```
#MADE
data2 <- data[ data[,3] %in% c( "Vietnam" , "Japan" , "Indonesia" ), ]
data2 <- rbind( data2 , world_data )
data2 [ , 4 ] <- as.POSIXct( data2 [ , 4 ] , format = "%m/%d/%Y" )
y2020 <- data2 [ format( data2 [ , 4 ] , format="%Y")=="2020" &
                  ( format( data2 [ , 4 ] , format="%m")=="02" |
                    format( data2 [ , 4 ] , format="%m")=="01" |
                    format( data2 [ , 4 ] , format="%m")=="07" |
                    format( data2 [ , 4 ] , format="%m")=="09" ) , ]
y2021 <- data2 [ format( data2 [ , 4 ] , format="%Y")=="2021" &
                  ( format( data2 [ , 4 ] , format="%m")=="02" |
                    format( data2 [ , 4 ] , format="%m")=="01" |
                    format( data2 [ , 4 ] , format="%m")=="07" |
                    format( data2 [ , 4 ] , format="%m")=="09" ) , ]
y2022 <- data2 [ format( data2 [ , 4 ] , format="%Y")=="2022" &
                  ( format( data2 [ , 4 ] , format="%m")=="02" |
                    format( data2 [ , 4 ] , format="%m")=="01" |
                    format( data2 [ , 4 ] , format="%m")=="07" |
                    format( data2 [ , 4 ] , format="%m")=="09" ) , ]

#last_month
y2020_1 <- data2 [ format( data2 [ , 4 ] , format="%Y")=="2020" &
                  ( format( data2 [ , 4 ] , format="%m")=="11" |
                    format( data2 [ , 4 ] , format="%m")=="12" ) , ]
y2021_1 <- data2 [ format( data2 [ , 4 ] , format="%Y")=="2021" &
                  ( format( data2 [ , 4 ] , format="%m")=="11" |
                    format( data2 [ , 4 ] , format="%m")=="12" ) , ]
y2022_1 <- data2 [ format( data2 [ , 4 ] , format="%Y")=="2022" &
                  ( format( data2 [ , 4 ] , format="%m")=="11" |
                    format( data2 [ , 4 ] , format="%m")=="12" ) , ]

#function
draw_chart <- function( year_data , month_data , yyyy ,
                        cases_or_deaths , avg_or_not="" )
{
  if ( dim( year_data ) == 0 )
    return( ggplot() + labs( x= "" , y=paste( cases_or_deaths , " " , yyyy )) +
      theme( legend . position="top" ) + ggtitle( "NA" ) )

  amedumb <- 5
  pain <- "Cases"
  if ( cases_or_deaths == "deaths" ){
    pain <- "Deaths"
    amedumb <- 6
  }
  if ( avg_or_not == "avg" )
  {
    tmp <- ave_handle( year_data , month_data , amedumb )
    year_data [ , amedumb ] <- tmp
  }
}
```



```
}

chart_out <- ggplot(data=year_data ,
aes(x=format(year_data[,4],format="%d"),y=year_data[,amedumb],
color=month_data,group=month_data))+ 
  geom_line(lwd=1)+ 
  labs(x="",y=paste(pain,"",yyyy))+ 
  theme(legend.position="top")
return(chart_out)
}

cum_rel <- function(year_data, month_data, yyyy,
                     cases_or_deaths, avg_or_not="")
{
  if(dim(year_data) == 0)
    return(ggplot() + labs(x="",y=paste(cases_or_deaths,"",yyyy))+
    theme(legend.position="top") + ggtitle("NA"))
  amedumb <- 5
  pain <- "Cases"
  if(cases_or_deaths == "deaths"){
    pain <- "Deaths"
    amedumb <- 6
  }
  if(avg_or_not == "avg")
  {
    tmp <- ave_handle(year_data,month_data,amedumb)
    year_data[,amedumb] <- tmp
  }

  cum_sum_data <- cbind(cumsum(year_data[,amedumb]))
  prob <- cum_sum_data/sum(year_data[,amedumb])
  year_data[,amedumb] <- prob
  chart_out <- ggplot(data=year_data, aes(x=year_data[,4],
  y=year_data[,amedumb],group=1))+ 
    geom_line(lwd=1)+ 
    labs(x="Dates",y=paste(pain,"_crf",yyyy))+ 
    scale_x_datetime(date_labels = "%m/%d/%Y",date_breaks = "1_week")+
    theme(legend.position="top")
  return(chart_out)
}

cum <- function(year_data, month_data, yyyy, cases_or_deaths, avg_or_not="")
{
  if(dim(year_data) == 0) return(ggplot() + labs(x="",y=paste(cases_or_deaths,"",yyyy))+
  theme(legend.position="top") + ggtitle("NA"))
  amedumb <- 5
  pain <- "Cases"
  if(cases_or_deaths == "deaths"){
    pain <- "Deaths"
    amedumb <- 6
  }
  if(avg_or_not == "avg")
  {
    tmp <- ave_handle(year_data, month_data, amedumb)
    year_data[,amedumb] <- tmp
  }
```



```
cum_sum_data <- cbind(cumsum(year_data[,amedumb]))  
prob <- cum_sum_data  
year_data[,amedumb] <- prob  
chart_out <- ggplot(data=year_data,  
aes(x=format(year_data[,4],format="%d"),y=year_data[,amedumb],  
color=month_data,group=month_data))+  
  geom_line(lwd=1)+  
  labs(x="",y=paste("Cumulavite_of", pain, "", yyyy))+  
  theme(legend.position="top")  
return(chart_out)  
}  
  
two_line_chart <- function(year_data, month_data, yyyy,  
cases_or_deaths, avg_or_not="")  
{  
  if(dim(year_data) == 0)  
    return(ggplot() + labs(x="",y=paste("Cases_and_Deaths","",yyyy)) +  
           theme(legend.position="top") + ggtitle("NA"))  
  cases <- c()  
  deaths <- c()  
  mon_uni <- cbind(month_data)  
  if(avg_or_not == "avg")  
  {  
    tmp <- ave_handle(year_data, month_data, 5)  
    year_data[,5] <- tmp  
    tmp <- ave_handle(year_data, month_data, 6)  
    year_data[,6] <- tmp  
  }  
  
  for(x in 1:length(mon_uni))  
  {  
    cases <- rbind(cases, paste("New_cases", toString(mon_uni[x])))  
    deaths <- rbind(deaths, paste("New_deaths", toString(mon_uni[x])))  
  }  
  cases_and_deaths <- rbind(cases, deaths)  
  chart_out <- ggplot(data=year_data,  
aes(x=format(year_data[,4],format="%d"),group=month_data))+  
  geom_line(lwd=1, aes(y = year_data[,5], colour = cases))+  
  geom_line(lwd=1, aes(y = year_data[,6], colour = deaths))+  
  labs(x="",y=paste("Cases_and_Deaths","",yyyy))+  
  theme(legend.position="top")  
  return(chart_out)  
}  
  
ave_7days <- function(mon)  
{  
  i <- 1  
  j <- 1  
  arr <- cbind(mon)  
  arr[is.na(arr)] <- 0  
  wah <- arr  
  lenlen <- length(arr)  
  while(i < lenlen + 1)  
  {  
    wah[i] <- arr[j:i]/(i - j + 1)  
    if(i >= 7)  
    {  
      wah[i] <- arr[j:i]/(i - j + 1)  
    }  
  }  
}
```



```
j <- j + 1
}
i <- i + 1
}
while(j < i)
{
    wah[j] <- arr[j:i]/(i - j + 1)
    #print(i - j + 1)
    j <- j + 1
}
#View(cases)
#cases <- cases[!(cases %in% NA)]
#print(sum(cases))
return(wah)
}

ave_handle <- function(df, mon, amedumb)
{
    mon_uni <- cbind(unique(mon))
    df[is.na(df)] <- 0
    arr <- c()
    for(x in mon_uni)
    {
        tmp <- df[format(df[,4], format="%m") == x,]
        arr <- rbind(arr, ave_7days(tmp[,amedumb]))
    }
    return(arr)
}

country_chart <- function(country, type_w, made, cases_or_deaths = "", chart_name, avg_or_not = "")
{
    Months_2020<-format(y2020[y2020[,3]==country,4], format="%m")
    Months_2021<-format(y2021[y2021[,3]==country,4], format="%m")
    Months_2022<-format(y2022[y2022[,3]==country,4], format="%m")

    Last_months_2020<-format(y2020_1[y2020_1[,3]==country,4], format="%m")
    Last_months_2021<-format(y2021_1[y2021_1[,3]==country,4], format="%m")
    Last_months_2022<-format(y2022_1[y2022_1[,3]==country,4], format="%m")
    if(type_w == "line_chart")
    {
        if(made == "2_1_7_9")
        {
            chart_2020 <- draw_chart(y2020[y2020[,3] == country,],
            Months_2020, "2020", cases_or_deaths, avg_or_not)
            chart_2021 <- draw_chart(y2021[y2021[,3] == country,],
            Months_2021, "2021", cases_or_deaths, avg_or_not)
            chart_2022 <- draw_chart(y2022[y2022[,3] == country,],
            Months_2022, "2022", cases_or_deaths, avg_or_not)
            ggsave(filename = paste(chart_name, country, ".jpeg"),
            plot = arrangeGrob(chart_2020, chart_2021, chart_2022),
            device = "jpeg", scale = 1, width = 9, height = 9)
        }
        else
        {
            chart_2020 <- draw_chart(y2020_1[y2020_1[,3] == country,],
            Last_months_2020, "2020", cases_or_deaths, avg_or_not)
        }
    }
}
```



```
chart_2021 <- draw_chart(y2021_1[y2021_1[,3] == country,],  
Last_months_2021, "2021", cases_or_deaths, avg_or_not)  
chart_2022 <- draw_chart(y2022_1[y2022_1[,3] == country,],  
Last_months_2022, "2022", cases_or_deaths, avg_or_not)  
ggsave(filename = paste(chart_name, country, ".jpeg"),  
plot = arrangeGrob(chart_2020, chart_2021, chart_2022),  
device = "jpeg", scale = 1, width = 9, height = 9)  
}  
}  
else if(type_w == "two_line")  
{  
if(made == "2_1_7_9")  
{  
chart_2020 <- two_line_chart(y2020[y2020[,3] == country,],  
Months_2020, "2020", cases_or_deaths, avg_or_not)  
chart_2021 <- two_line_chart(y2021[y2021[,3] == country,],  
Months_2021, "2021", cases_or_deaths, avg_or_not)  
chart_2022 <- two_line_chart(y2022[y2022[,3] == country,],  
Months_2022, "2022", cases_or_deaths, avg_or_not)  
ggsave(filename = paste(chart_name, country, ".jpeg"),  
plot = arrangeGrob(chart_2020, chart_2021, chart_2022),  
device = "jpeg", scale = 1, width = 9, height = 9)  
}  
else  
{  
chart_2020 <- two_line_chart(y2020_1[y2020_1[,3] == country,],  
Last_months_2020, "2020", cases_or_deaths, avg_or_not)  
chart_2021 <- two_line_chart(y2021_1[y2021_1[,3] == country,],  
Last_months_2021, "2021", cases_or_deaths, avg_or_not)  
chart_2022 <- two_line_chart(y2022_1[y2022_1[,3] == country,],  
Last_months_2022, "2022", cases_or_deaths, avg_or_not)  
ggsave(filename = paste(chart_name, country, ".jpeg"),  
plot = arrangeGrob(chart_2020, chart_2021, chart_2022),  
device = "jpeg", scale = 1, width = 9, height = 9)  
}  
}  
else if(type_w == "cum")  
{  
if(made == "2_1_7_9")  
{  
chart_2020 <- cum(y2020[y2020[,3] == country,],  
Months_2020, "2020", cases_or_deaths, avg_or_not)  
chart_2021 <- cum(y2021[y2021[,3] == country,],  
Months_2021, "2021", cases_or_deaths, avg_or_not)  
chart_2022 <- cum(y2022[y2022[,3] == country,],  
Months_2022, "2022", cases_or_deaths, avg_or_not)  
ggsave(filename = paste(chart_name, country, ".jpeg"),  
plot = arrangeGrob(chart_2020, chart_2021, chart_2022),  
device = "jpeg", scale = 1, width = 9, height = 9)  
}  
else if(made == "11_12")  
{  
chart_2020 <- cum(y2020_1[y2020_1[,3] == country,],  
Last_months_2020, "2020", cases_or_deaths, avg_or_not)  
chart_2021 <- cum(y2021_1[y2021_1[,3] == country,],  
Last_months_2021, "2021", cases_or_deaths, avg_or_not)  
chart_2022 <- cum(y2022_1[y2022_1[,3] == country,],
```



```
Last_months_2022, "2022", cases_or_deaths, avg_or_not)
ggsave(filename = paste(chart_name, country, ".jpeg"),
plot = arrangeGrob(chart_2020, chart_2021, chart_2022),
device = "jpeg", scale = 1, width = 9, height = 9)
}
}
else
{
  if(made == "2_1_7_9")
  {
    chart_2020 <- cum_rel(y2020[y2020[,3] == country,],
    Months_2020, "2020", cases_or_deaths, avg_or_not)
    chart_2021 <- cum_rel(y2021[y2021[,3] == country,],
    Months_2021, "2021", cases_or_deaths, avg_or_not)
    chart_2022 <- cum_rel(y2022[y2022[,3] == country,],
    Months_2022, "2022", cases_or_deaths, avg_or_not)
    ggsave(filename = paste(chart_name, country, ".jpeg"),
    plot = arrangeGrob(chart_2020, chart_2021, chart_2022),
    device = "jpeg", scale = 1, width = 9, height = 9)
  }
  else if(made == "11_12")
  {
    chart_2020 <- cum_rel(y2020_1[y2020_1[,3] == country,],
    Last_months_2020, "2020", cases_or_deaths, avg_or_not)
    chart_2021 <- cum_rel(y2021_1[y2021_1[,3] == country,],
    Last_months_2021, "2021", cases_or_deaths, avg_or_not)
    chart_2022 <- cum_rel(y2022_1[y2022_1[,3] == country,],
    Last_months_2022, "2022", cases_or_deaths, avg_or_not)
    ggsave(filename = paste(chart_name, country, ".jpeg"),
    plot = arrangeGrob(chart_2020, chart_2021, chart_2022),
    device = "jpeg", scale = 1, width = 9, height = 9)
  }
}
}
```

- Dữ liệu cho câu v vi vii viii được lọc và lưu vào biến *data2*(bao gồm dữ liệu của 3 nước theo mã đê và toàn thế giới)
- Dữ liệu của các tháng theo các ký số trên mã đê của từng năm được đưa vào các biến *y2020*, *y2021*, *y2022*. Biến *y2020_1*, *y2021_1*, *y2022_1* nhận dữ liệu 2 tháng cuối của từng năm
- Hàm *draw_chart* dùng để trả về biểu đồ một đường theo từng tháng trong năm.
- Hàm *two_line_chart* dùng để trả về biểu hai đồ đường theo từng tháng trong năm.
- Hàm *cum* dùng để trả về biểu đồ tích lũy.
- Hàm *cum_rel* dùng để trả về biểu đồ tương đối tích lũy.
- Các hàm *draw_chart*, *cum_rel*, *cum*, *two_line_chart* chứa các tham số giống nhau:
 - *year_data*: tham số nhận vào dữ liệu của năm theo yêu cầu đê bài.



- *month_data*: tham số nhận vào dữ liệu của từng tháng theo yêu cầu đề bài.
- *yyyy*: tham số nhận vào năm để hiển thị lên biểu đồ.
- *cases_or_deaths*: nếu tham số nhận giá trị "cases" thì function sẽ vẽ biểu đồ theo nhiễm bệnh, còn nếu nhận giá trị "deaths" thì function sẽ vẽ biểu đồ theo tử vong. Nếu yêu cầu đề bài là cả 2 thì biến này có thể nhập hoặc để trống.
- *avg_or_not*: tham số có giá trị mặc định là "" nếu đưa giá trị "avg" vào tham số thì sẽ tính trung bình nếu đề bài có yêu cầu.
- Hàm *ave_7days* hỗ trợ việc tính trung bình 7 ngày gần nhất, hàm *ave_handle* để tính trung bình 7 ngày gần nhất theo từng tháng.
- Hàm *country_chart* dùng để xuất ra biểu đồ theo yêu cầu của đề bài. Hàm chứa các tham số:
 - *country*: tham số nhận vào tên của dữ liệu cần xuất biểu đồ.
 - *type_w*: loại biểu đồ cần vẽ:
 - * Biểu đồ một đường: "draw_chart"
 - * Biểu đồ hai đường: "two_line_chart"
 - * Biểu đồ tích lũy: "cum"
 - * Biểu đồ tương đối tích lũy: "cum_rel"
 - *made*: tham số quản lý việc xuất biểu đồ theo các tháng theo các ký số trong mã đề hoặc là 2 tháng cuối năm, nếu made nhận vào giá trị "2_1_7_9" thì nó sẽ xuất ra biểu đồ theo tháng theo các ký số trong mã đề của từng năm, nếu made nhận vào giá trị "11_12" thì sẽ xuất ra biểu đồ theo 2 tháng cuối của từng năm.
 - *cases_or_deaths*: tham số quản lý việc xuất ra biểu đồ theo nhiễm bệnh, tử vong hoặc cả 2. Nếu tham số nhận vào giá trị "cases" thì sẽ xuất ra biểu đồ theo nhiễm bệnh, còn nếu nhận vào giá trị "deaths" thì sẽ xuất ra biểu đồ theo tử vong, còn nếu trường hợp xuất ra biểu đồ chứa cả 2 thì có thể để trống hoặc nhập vào một giá trị bất kỳ.
 - *chart_name*: tham số nhận vào tên của biểu đồ.
 - *avg_or_not*: tham số quản lý việc có xuất ra biểu đồ với giá trị trung bình 7 ngày gần nhất hay không, tham số với giá trị mặc định là "", nếu tham số nhận vào giá trị "avg" thì sẽ xuất ra biểu đồ với giá trị trung bình 7 ngày gần nhất theo tháng của từng năm.

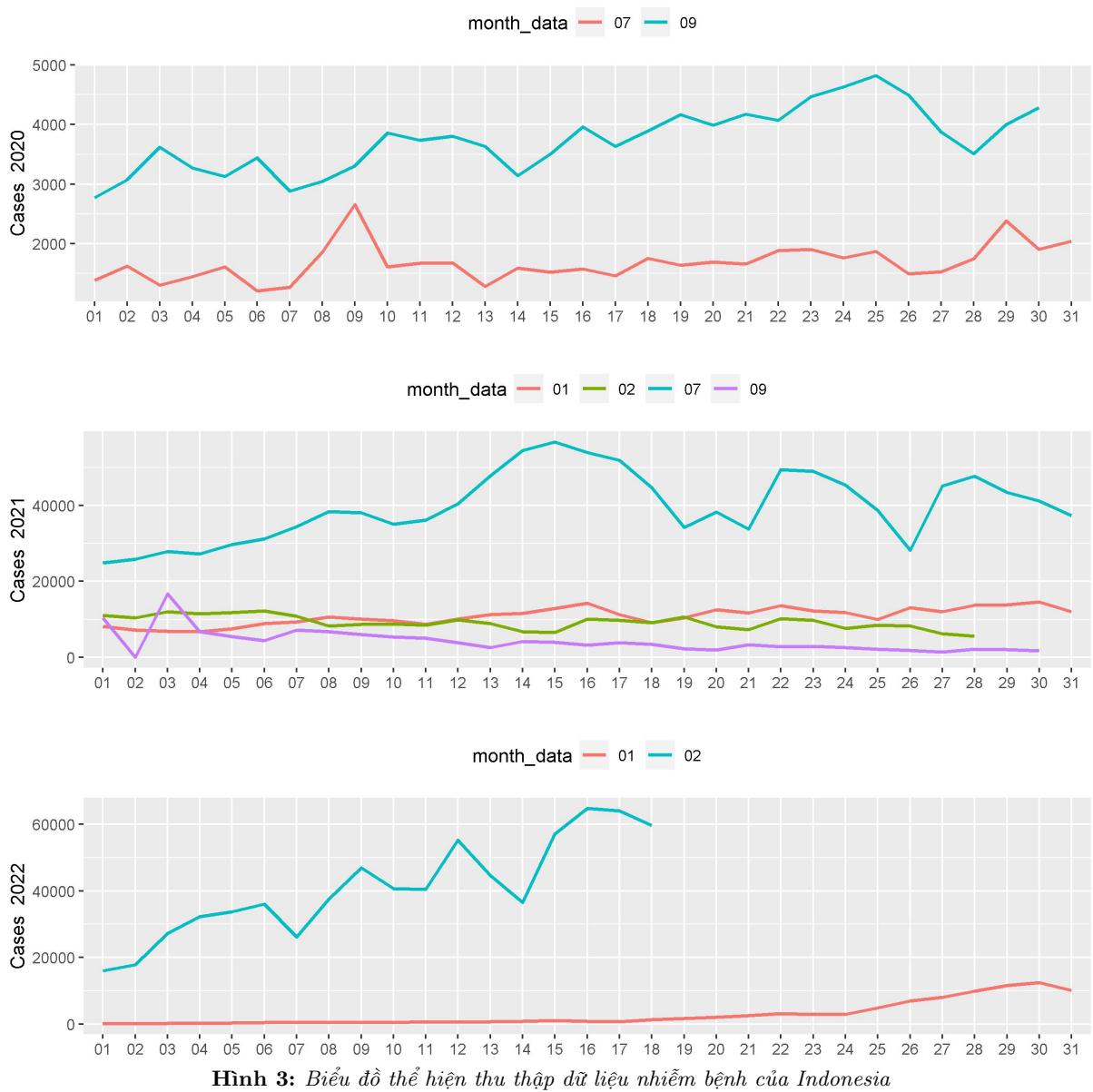
v) Nhóm câu hỏi liên quan đến trực quan dữ liệu theo thời gian là tháng

Với mỗi quốc gia mà thuộc về nhóm, trên từng năm hãy vẽ biểu đồ thể hiện trực Ox là thời gian, trực 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.

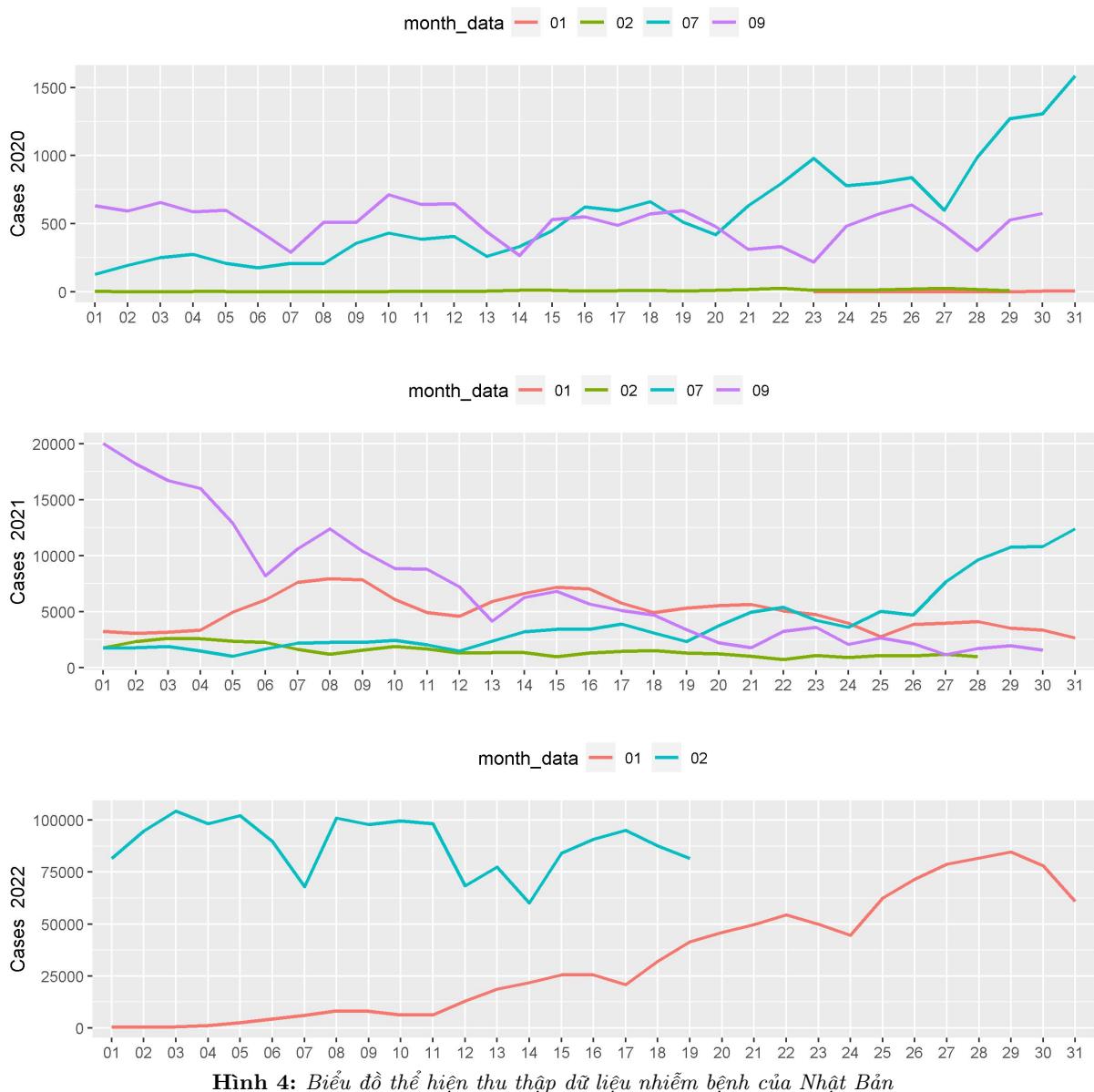
1) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh cho từng tháng

Source code

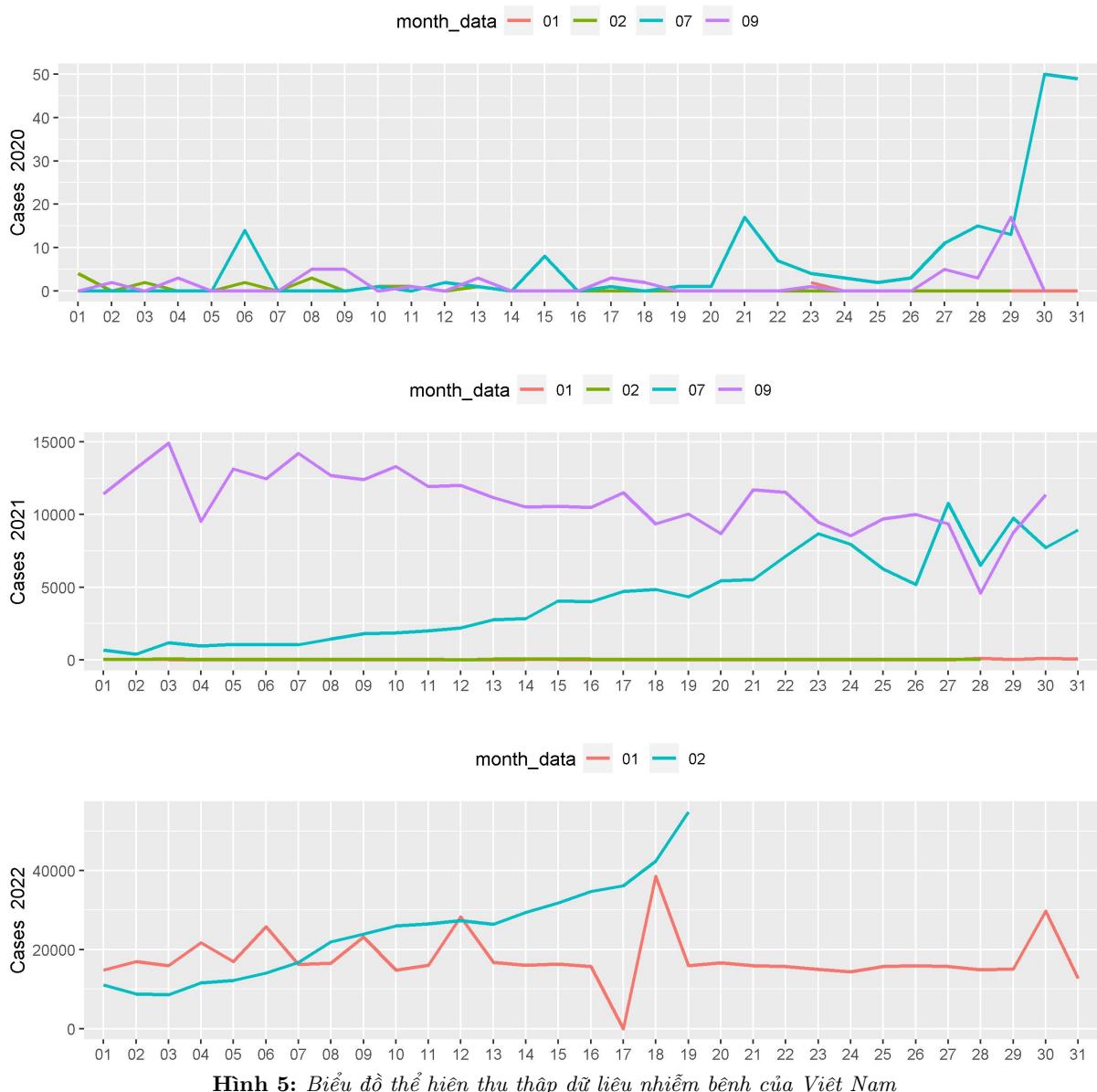
```
#v1
country_chart ("Vietnam", "line_chart", "2_1_7_9", "cases", "v1")
country_chart ("Japan", "line_chart", "2_1_7_9", "cases", "v1")
country_chart ("Indonesia", "line_chart", "2_1_7_9", "cases", "v1")
```



Hình 3: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh của Indonesia



Hình 4: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh của Nhật Bản

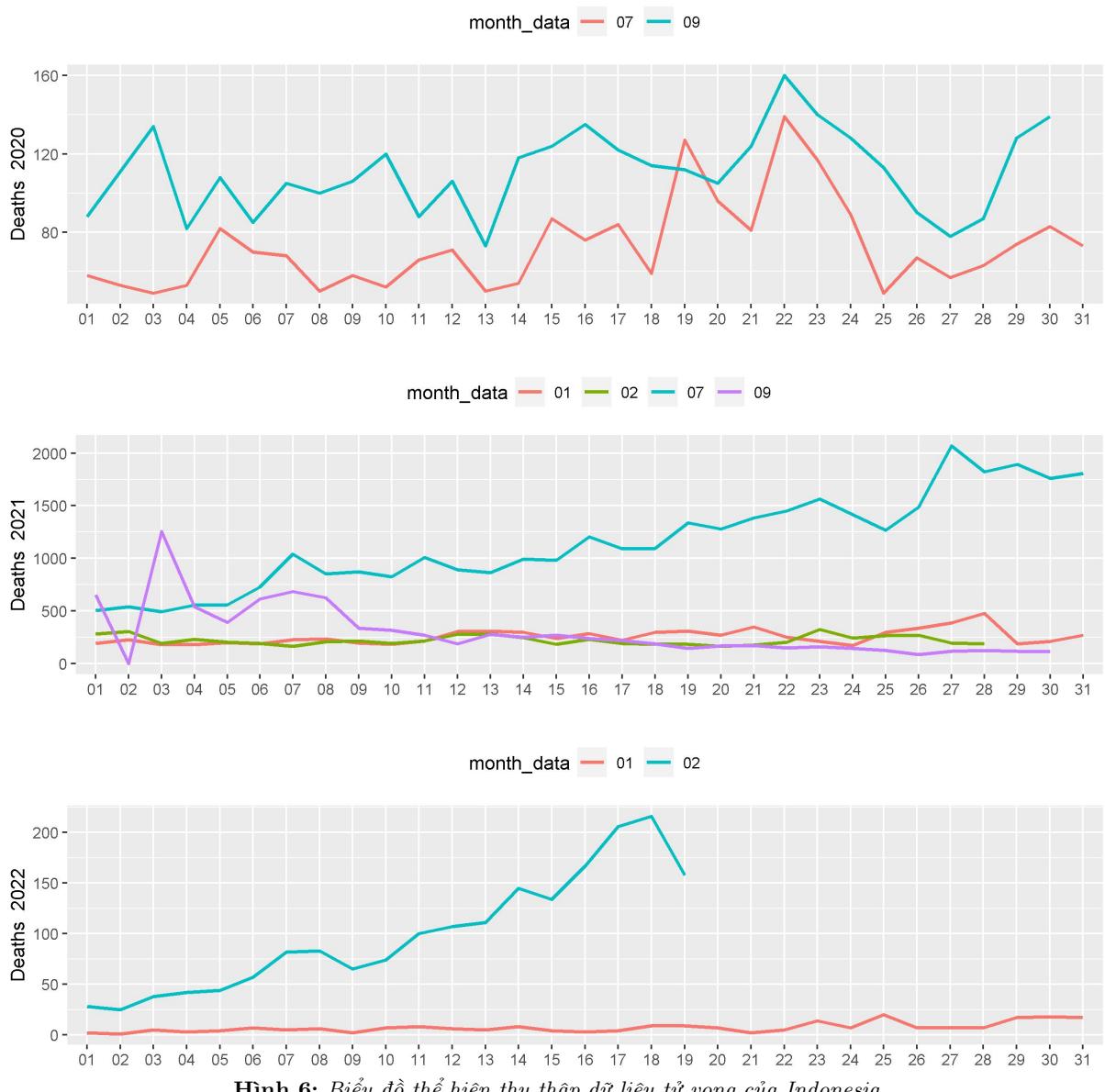


Hình 5: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh của Việt Nam

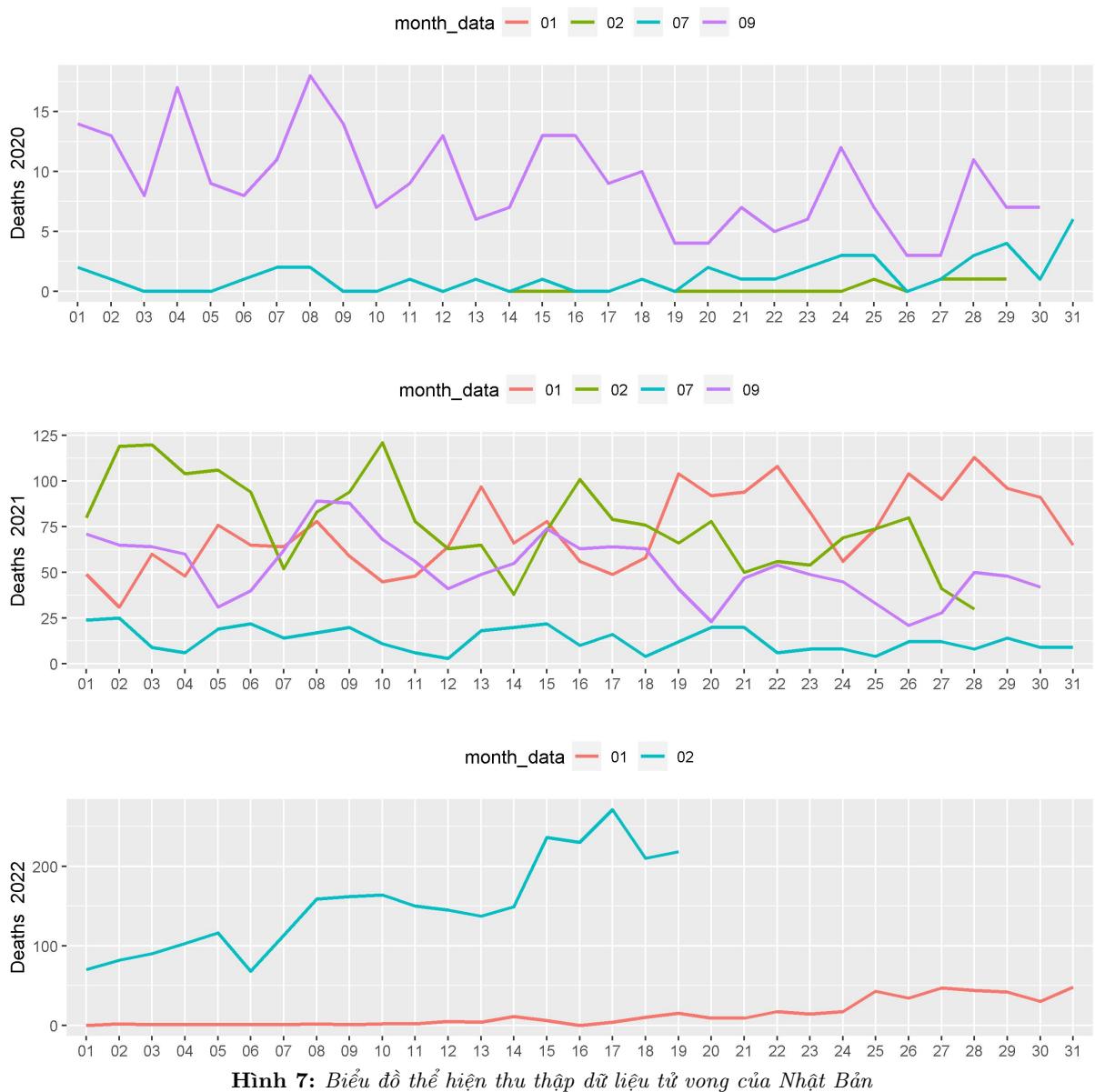
2) Biểu đồ thể hiện thu thập dữ liệu tử vong cho từng tháng

Source code

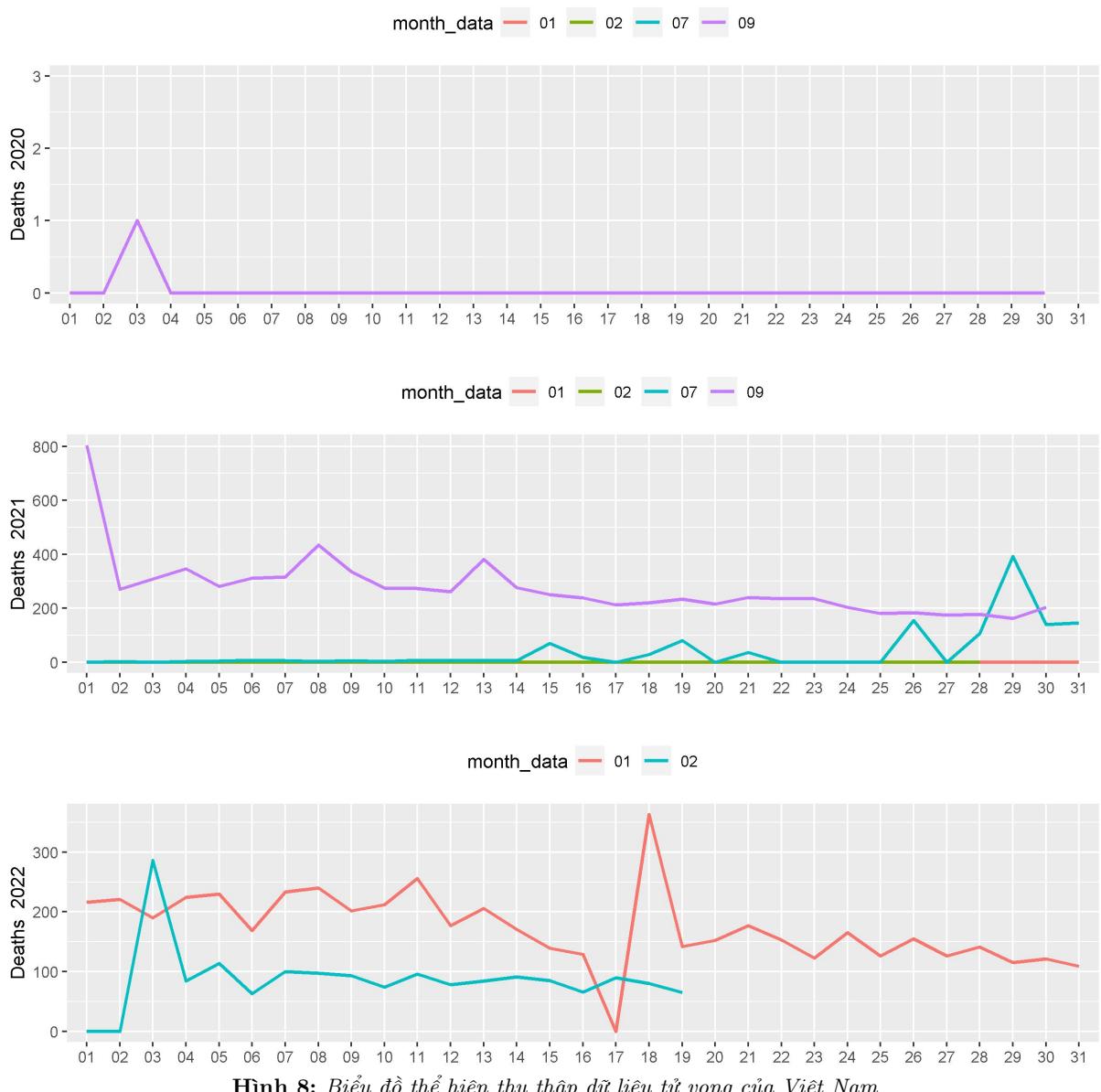
```
#v2
country_chart("Vietnam","line_chart","2_1_7_9","deaths","v2")
country_chart("Japan","line_chart","2_1_7_9","deaths","v2")
country_chart("Indonesia","line_chart","2_1_7_9","deaths","v2")
```



Hình 6: Biểu đồ thể hiện thu thập dữ liệu tử vong của Indonesia



Hình 7: Biểu đồ thể hiện thu thập dữ liệu tử vong của Nhật Bản



Hình 8: Biểu đồ thể hiện thu thập dữ liệu tử vong của Việt Nam

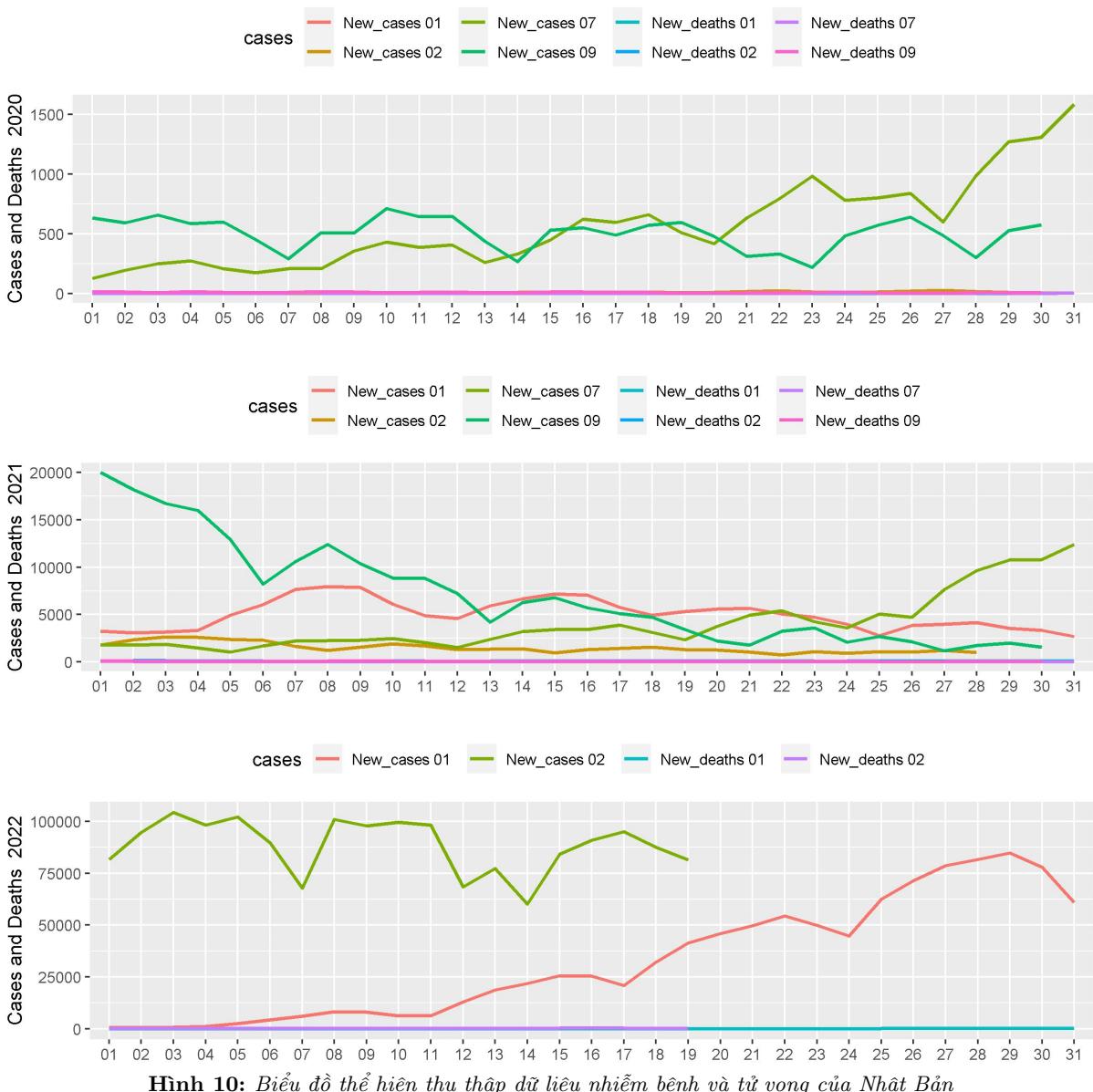
3) Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong cho từng tháng

Source code

```
#v3
country_chart ("Vietnam", "two_line", "2_1_7_9", "", "v3")
country_chart ("Japan", "two_line", "2_1_7_9", "", "v3")
country_chart ("Indonesia", "two_line", "2_1_7_9", "", "v3")
```



Hình 9: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong của Indonesia



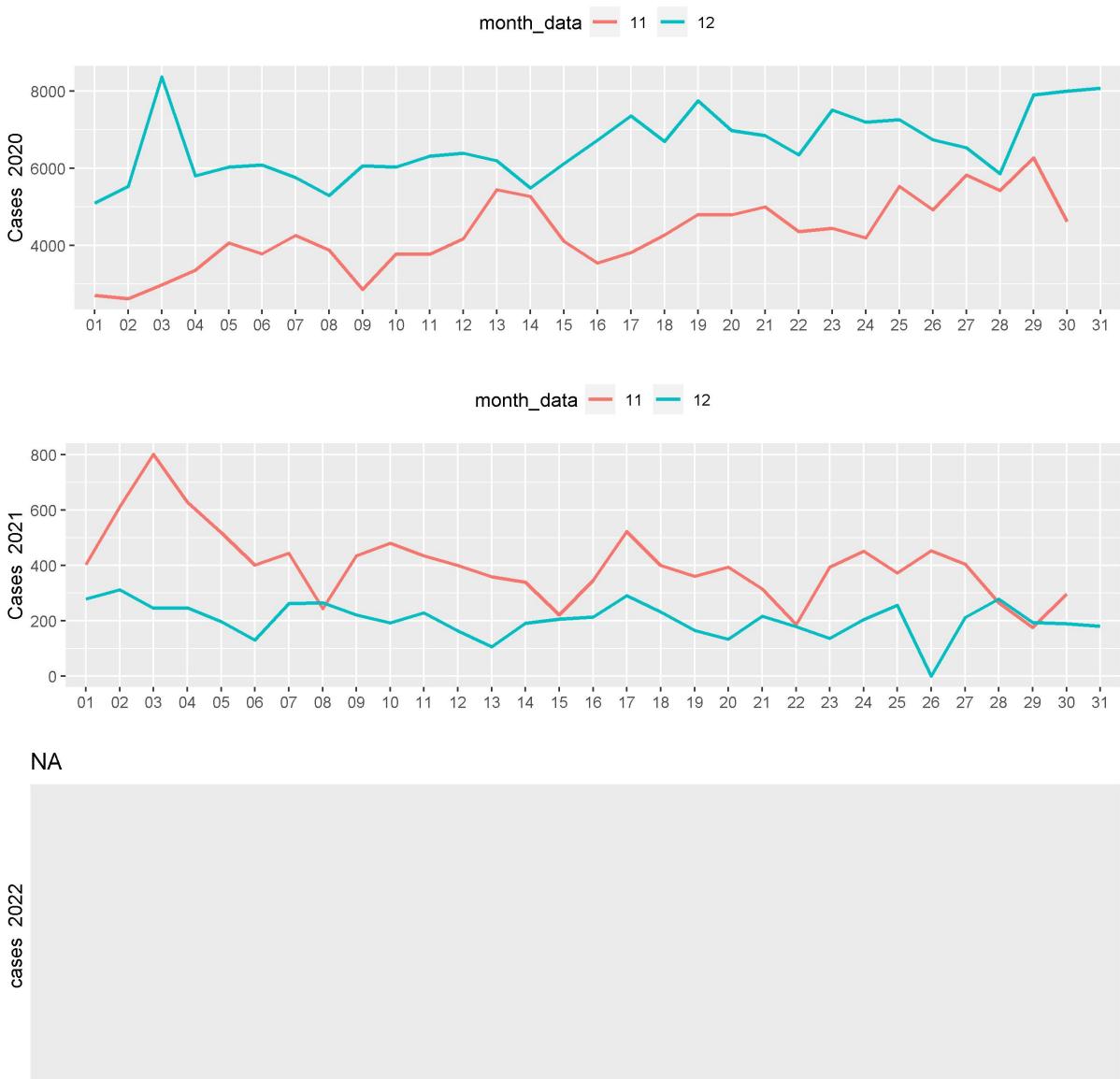


Hình 11: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong của Việt Nam

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

Source code

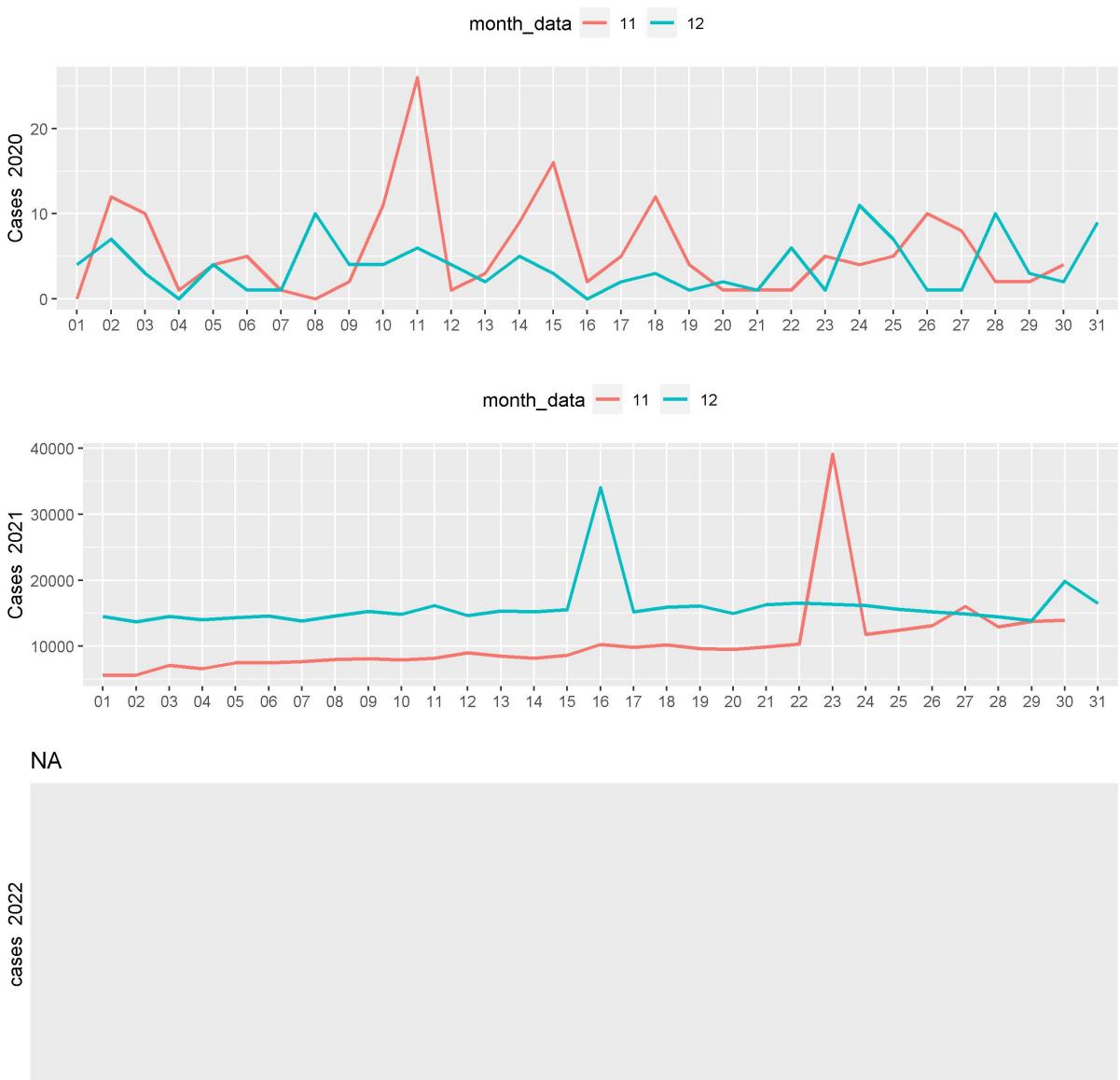
```
#v4
country_chart("Vietnam", "line_chart", "11_12", "cases", "v4")
country_chart("Japan", "line_chart", "11_12", "cases", "v4")
country_chart("Indonesia", "line_chart", "11_12", "cases", "v4")
```



Hình 12: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo 2 tháng cuối của Indonesia



Hình 13: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo 2 tháng cuối của Nhật Bản

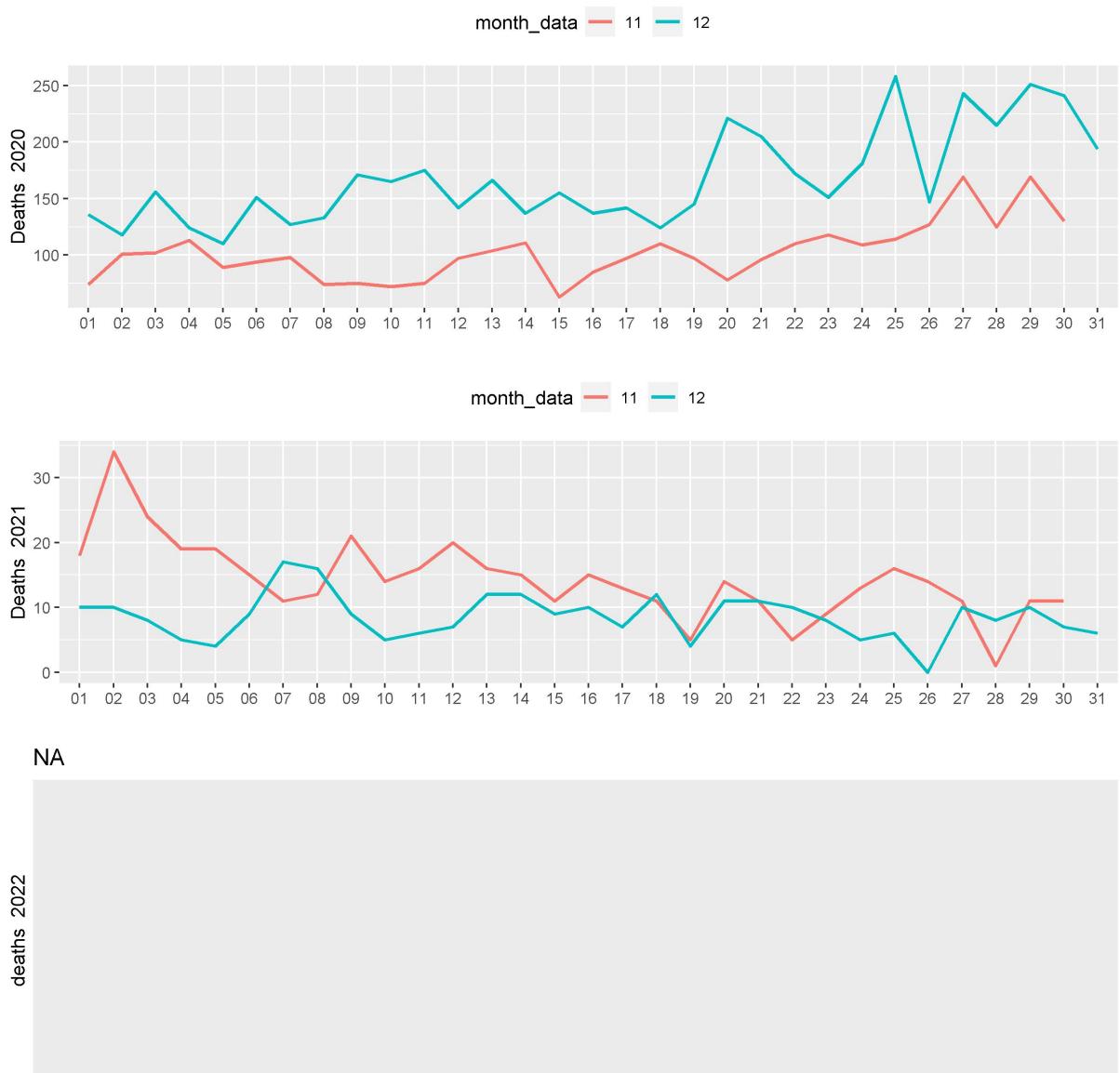


Hình 14: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo 2 tháng cuối của Việt Nam

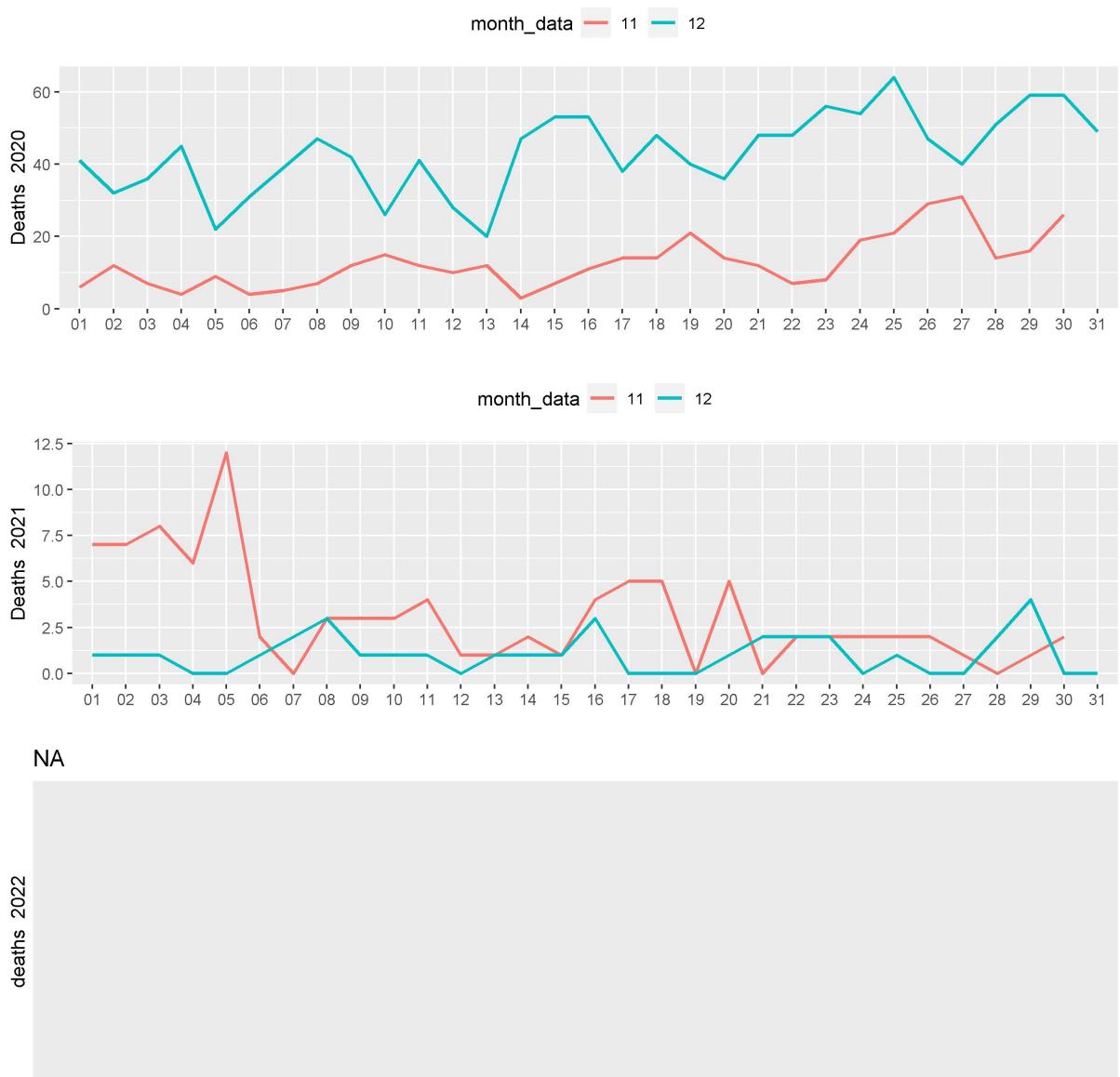
5) Biểu đồ thể hiện thu thập dữ liệu tử vong gồm 2 tháng cuối của năm

Source code

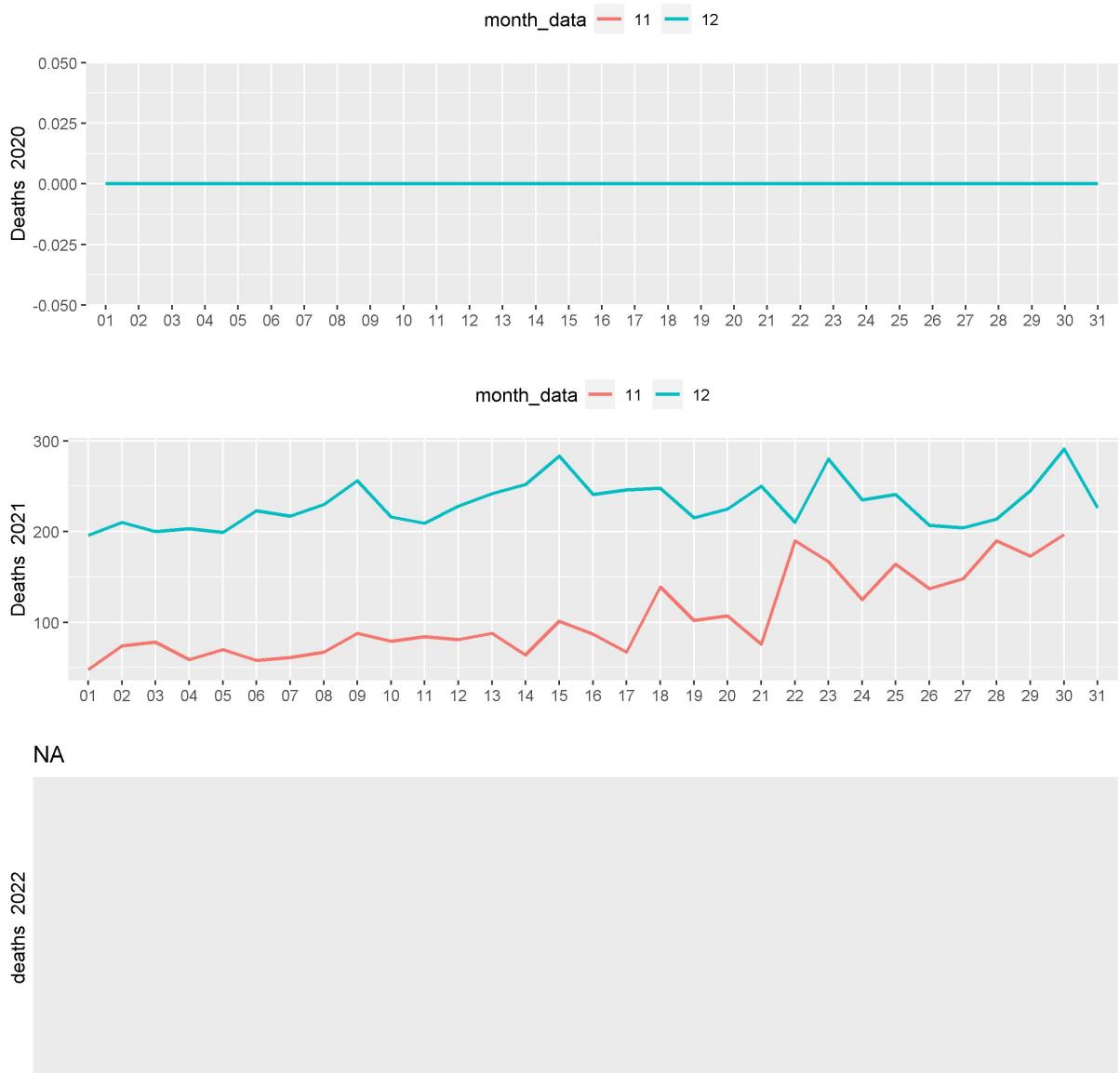
```
#v5
country_chart("Vietnam", "line_chart", "11_12", "deaths", "v5")
country_chart("Japan", "line_chart", "11_12", "deaths", "v5")
country_chart("Indonesia", "line_chart", "11_12", "deaths", "v5")
```



Hình 15: Biểu đồ thể hiện thu thập dữ liệu tử vong theo 2 tháng cuối của Indonesia



Hình 16: Biểu đồ thể hiện thu thập dữ liệu tử vong theo 2 tháng cuối của Nhật Bản

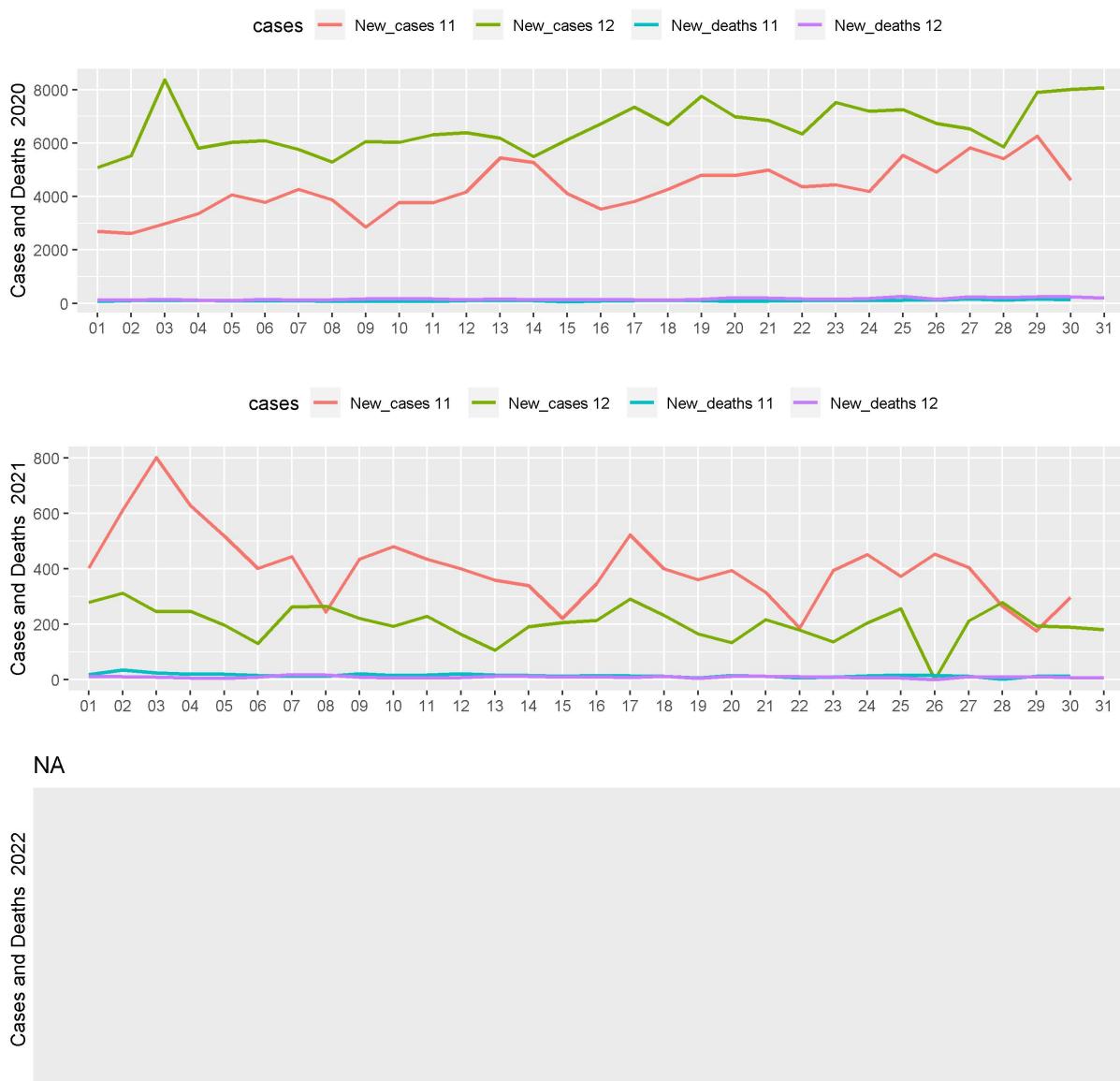


Hình 17: Biểu đồ thể hiện thu thập dữ liệu tử vong theo 2 tháng cuối của Việt Nam

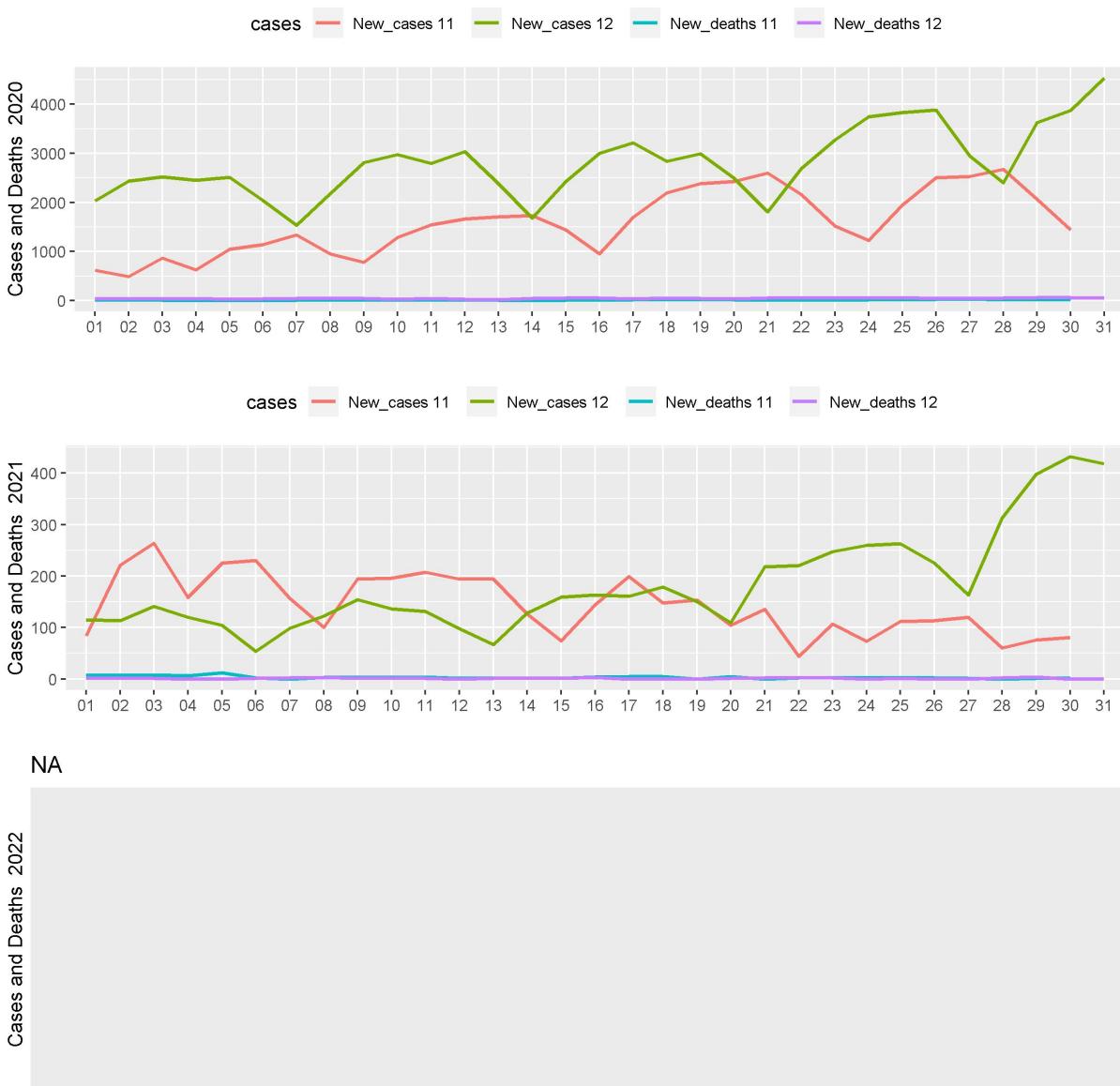
6) Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong gồm 2 tháng cuối của năm

Source code

```
#vv6
country_chart ("Vietnam", "two_line", "11_12", "", "v6")
country_chart ("Japan", "two_line", "11_12", "", "v6")
country_chart ("Indonesia", "two_line", "11_12", "", "v6")
```



Hình 18: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong theo 2 tháng cuối của Indonesia



Hình 19: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong theo 2 tháng cuối của Nhật Bản

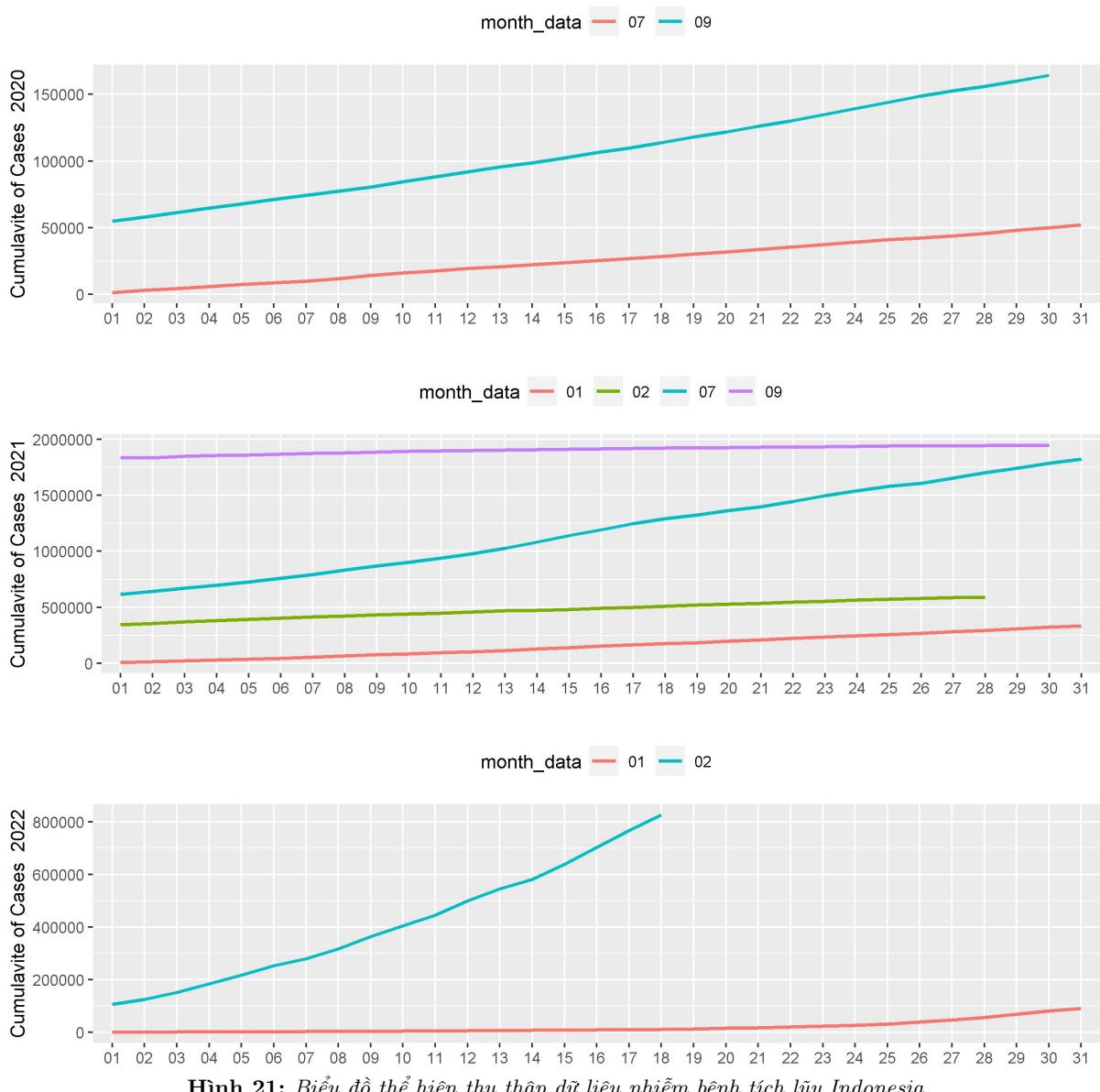


Hình 20: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong theo 2 tháng cuối của Việt Nam

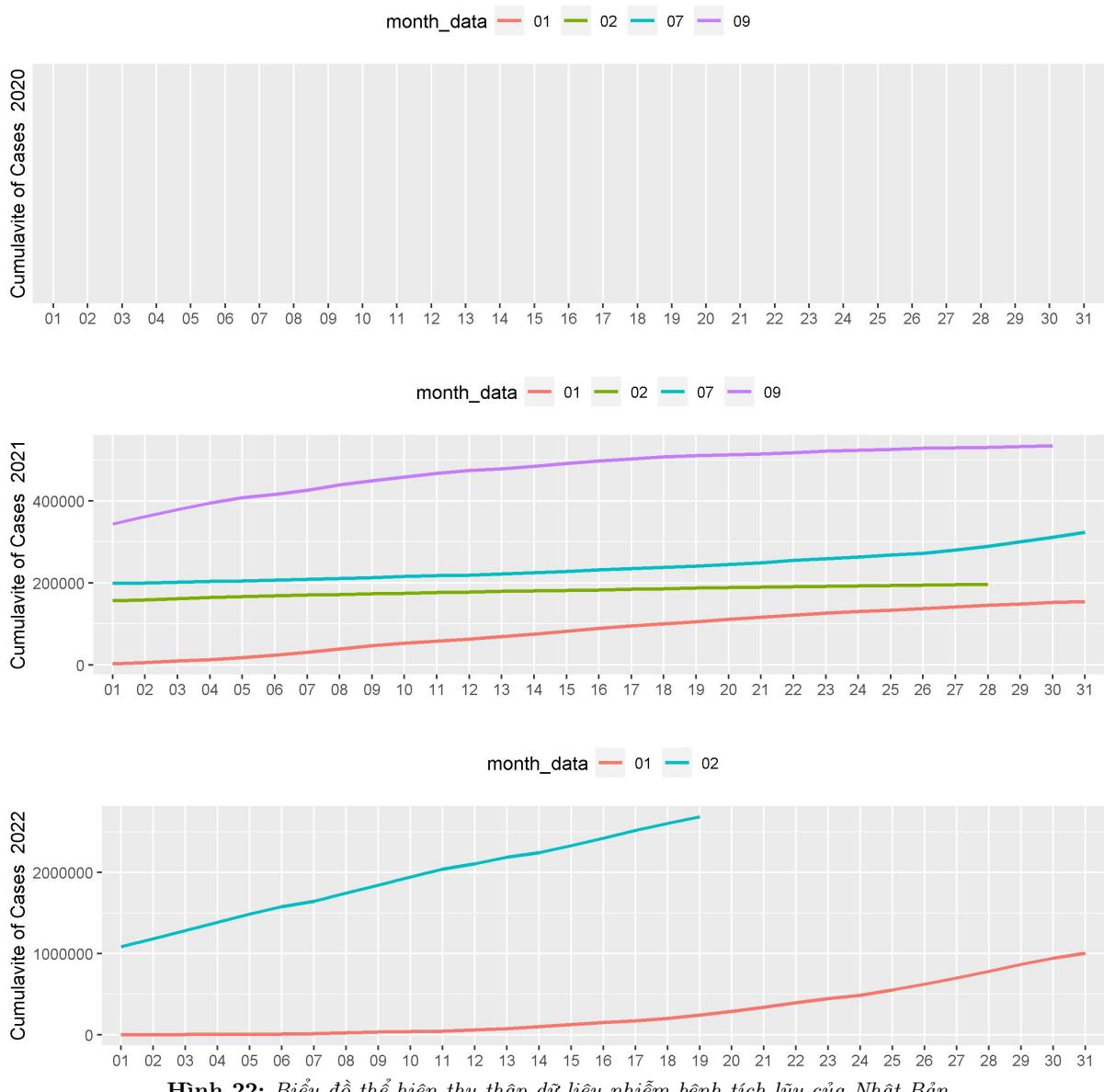
7) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy cho từng tháng

Source code

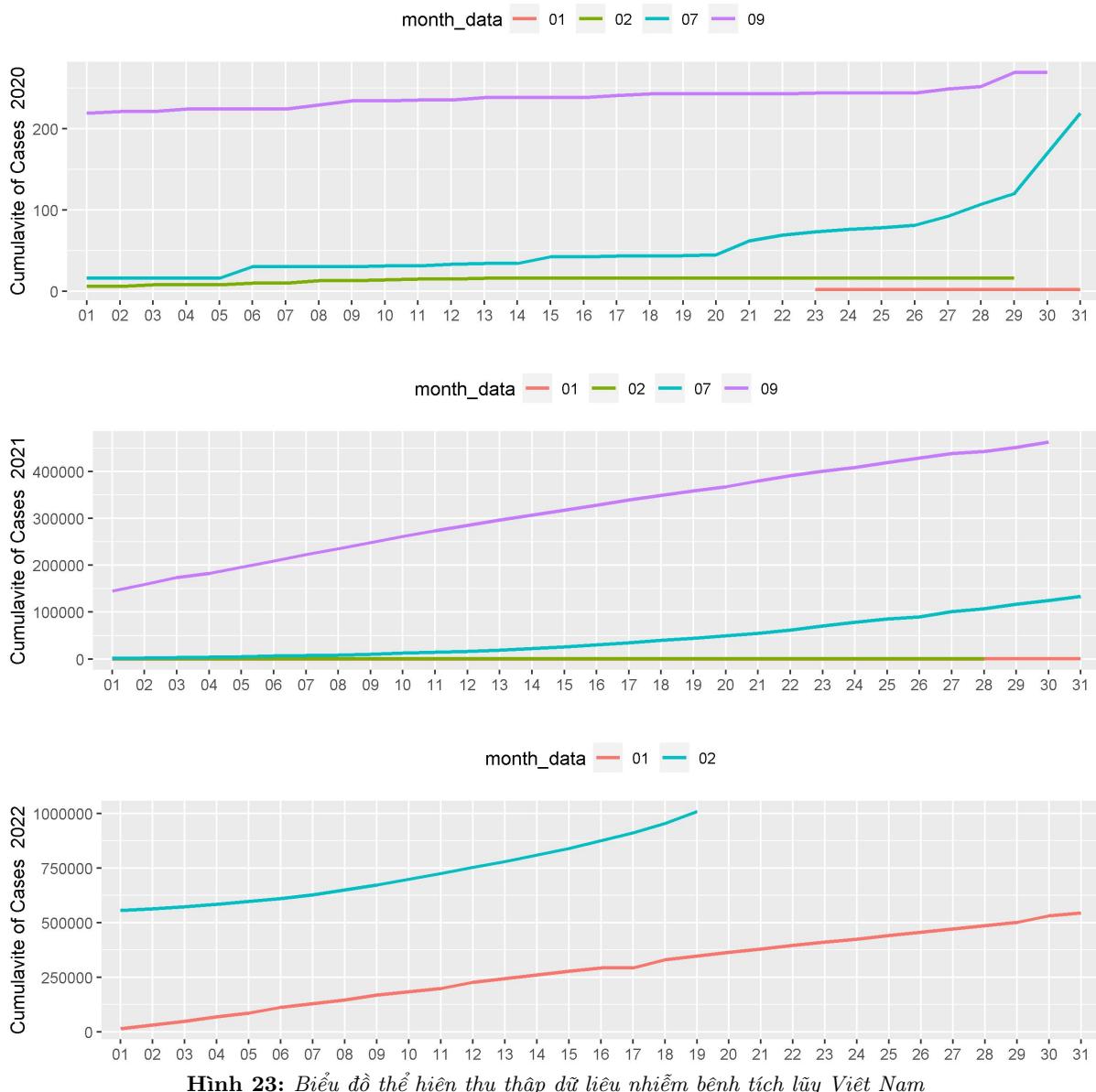
```
#v7
country_chart ("Vietnam", "cum", "2_1_7_9", "cases", "v7")
country_chart ("Japan", "cum", "2_1_7_9", "cases", "v7")
country_chart ("Indonesia", "cum", "2_1_7_9", "cases", "v7")
```



Hình 21: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy Indonesia



Hình 22: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy của Nhật Bản

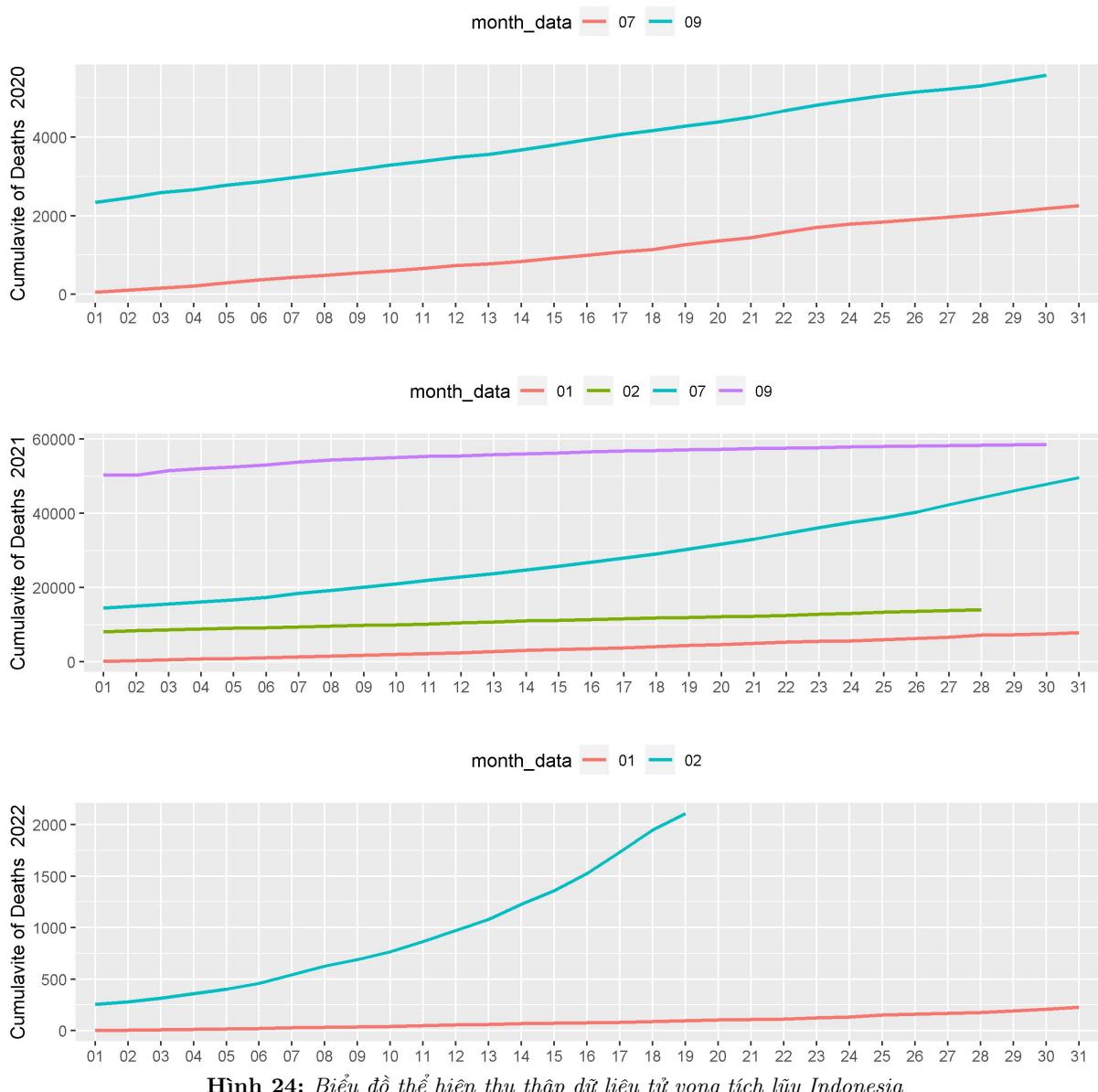


Hình 23: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy Việt Nam

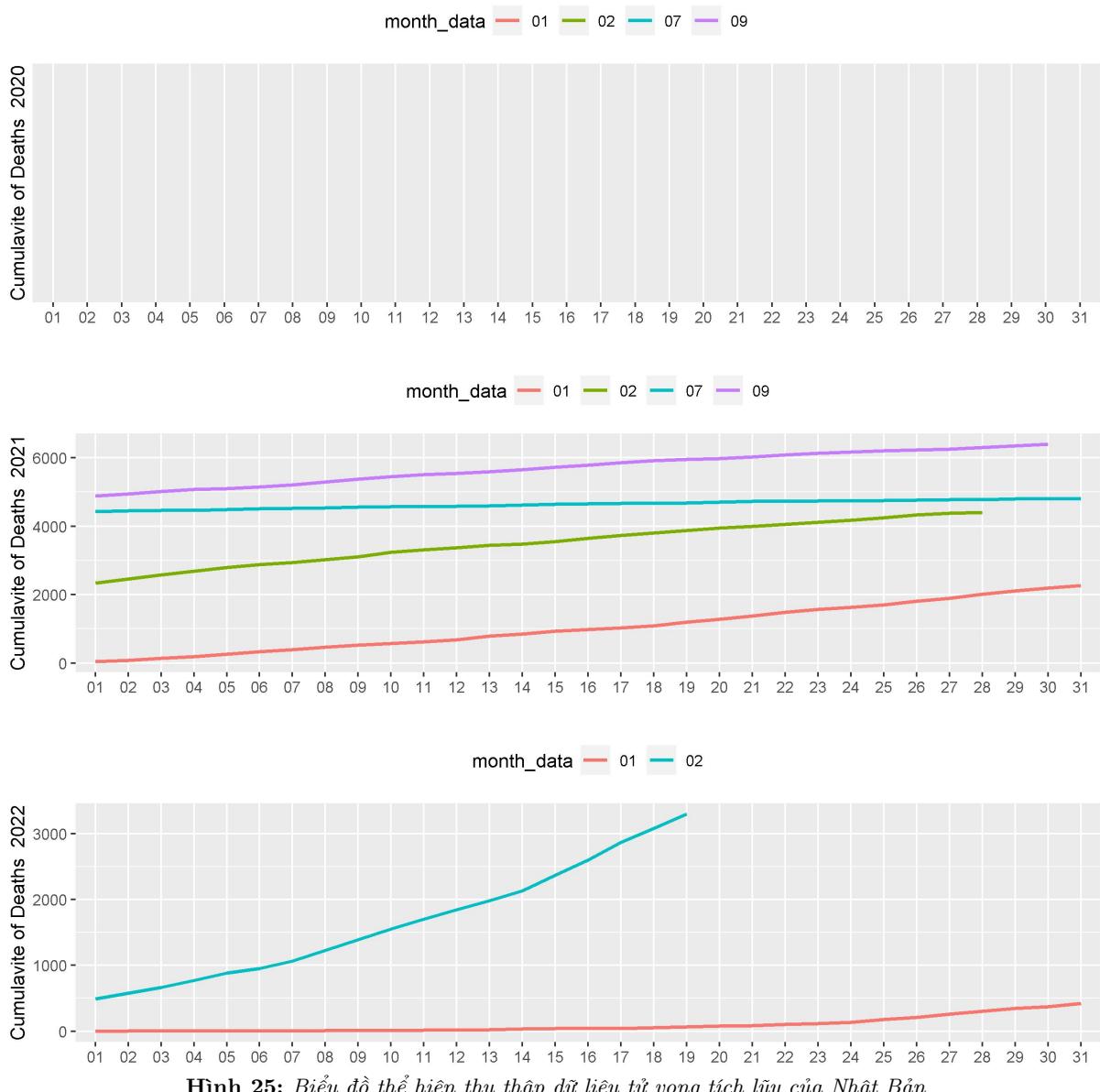
8) Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy cho từng tháng

Source code

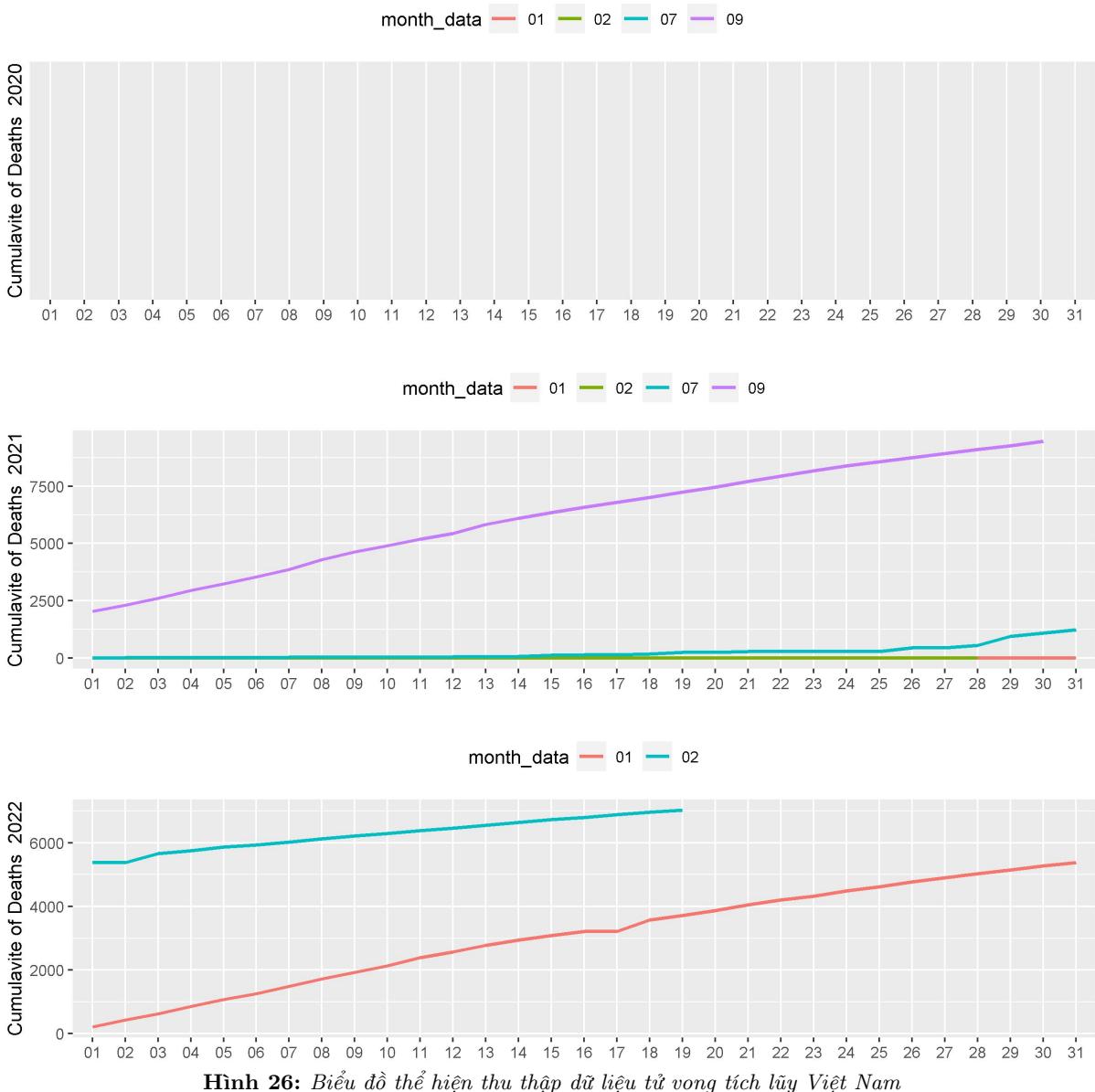
```
#v8
country_chart ("Vietnam", "cum", "2_1_7_9", "deaths", "v8")
country_chart ("Japan", "cum", "2_1_7_9", "deaths", "v8")
country_chart ("Indonesia", "cum", "2_1_7_9", "deaths", "v8")
```



Hình 24: Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy Indonesia



Hình 25: Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy của Nhật Bản



Hình 26: Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy Việt Nam

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

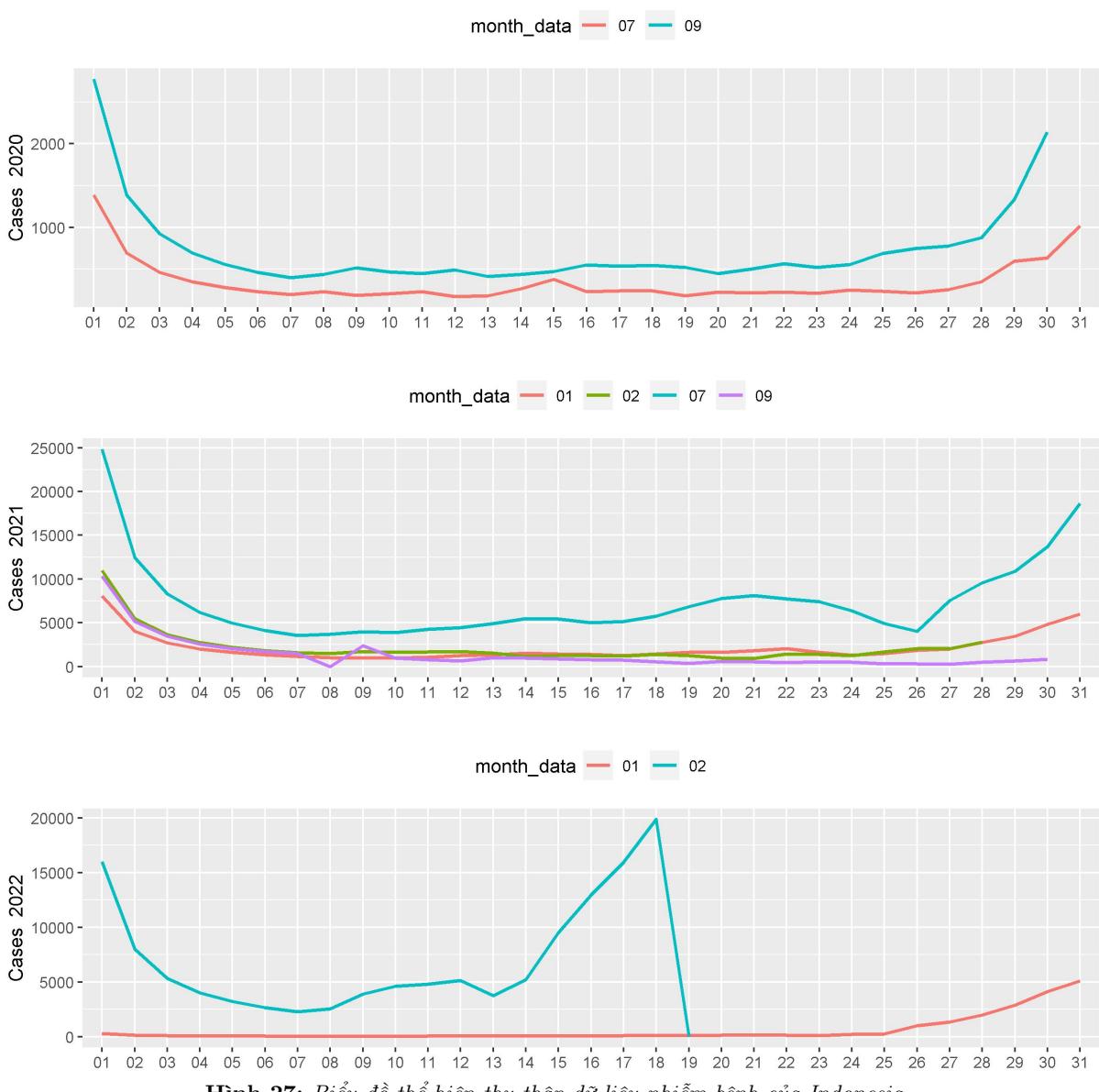
- Với mỗi quốc gia mà thuộc về nhóm, trên từng năm hãy vẽ biểu đồ thể hiện trực Ox là thời gian, trực Oy là nhiễm bệnh/tử vong. Hãy dùng 4 ký số của mã đề để vẽ 4 tháng tương ứng theo ký số đó. Nếu ký số là 0 thì lấy tháng là 10.

- 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.

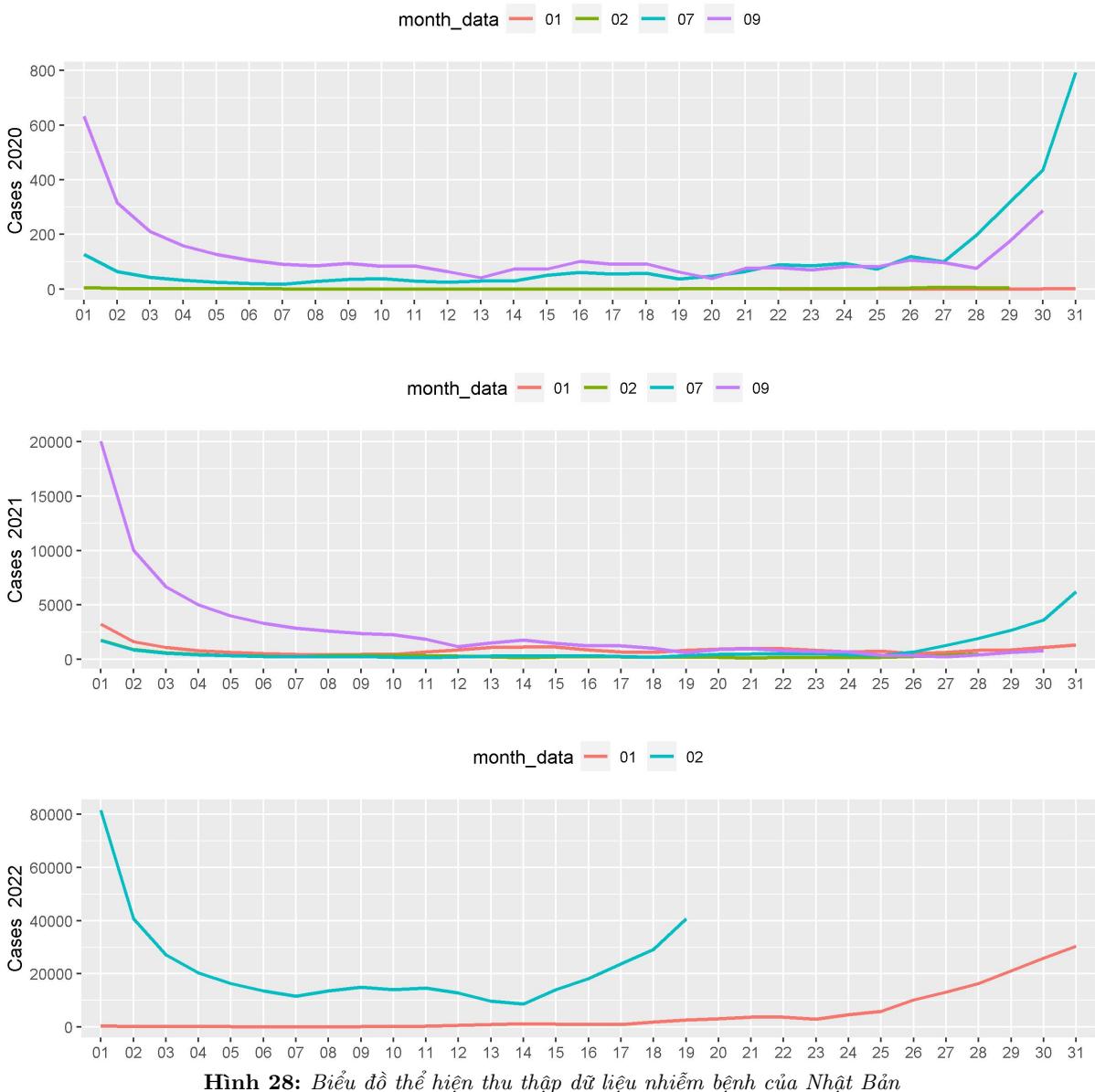
1) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh cho từng tháng

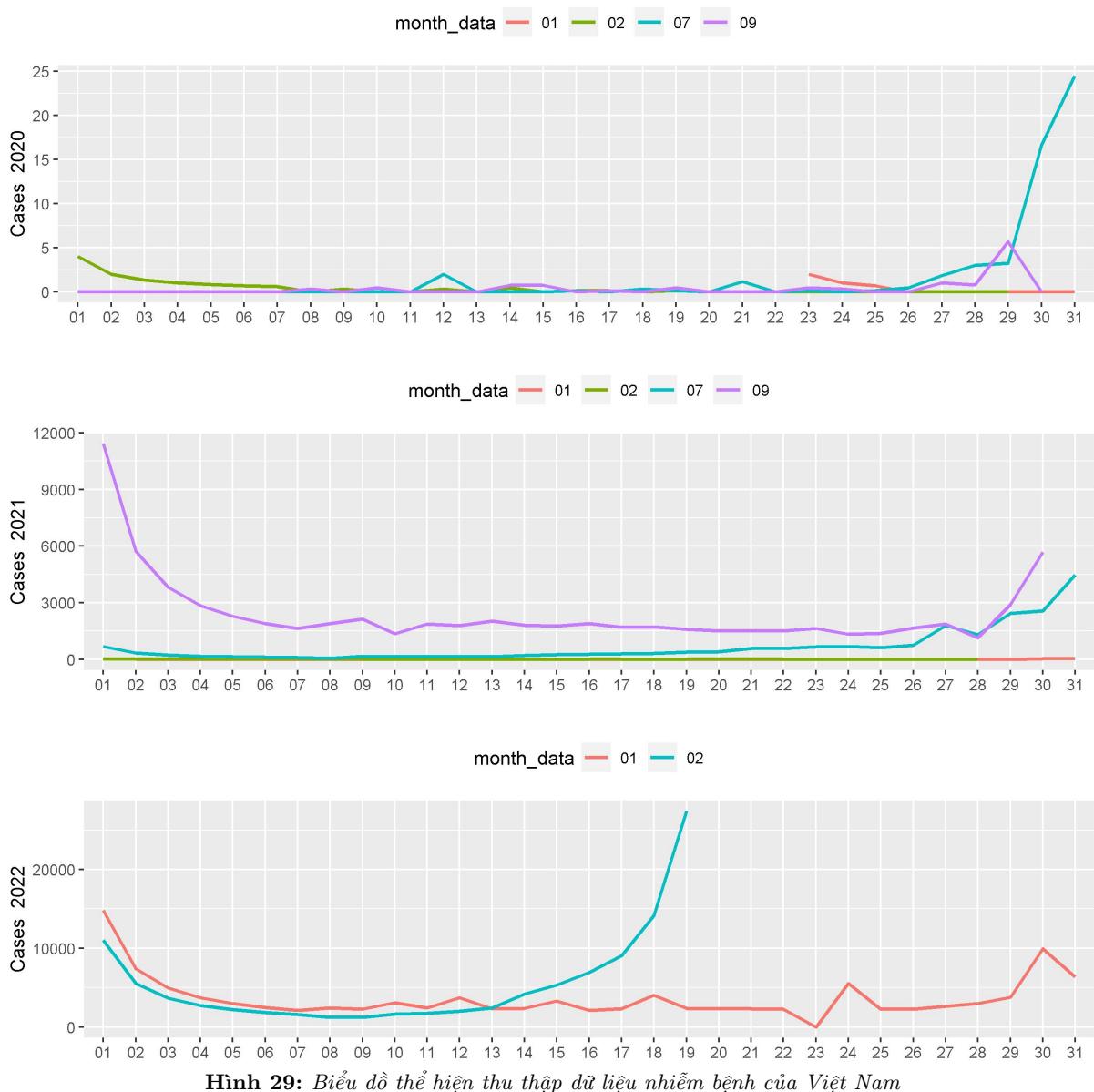
Source code

```
#vi1
country_chart ("Vietnam", "line_chart", "2_1_7_9", "cases", "vi1", "avg")
country_chart ("Japan", "line_chart", "2_1_7_9", "cases", "vi1", "avg")
country_chart ("Indonesia", "line_chart", "2_1_7_9", "cases", "vi1", "avg")
```



Hình 27: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh của Indonesia

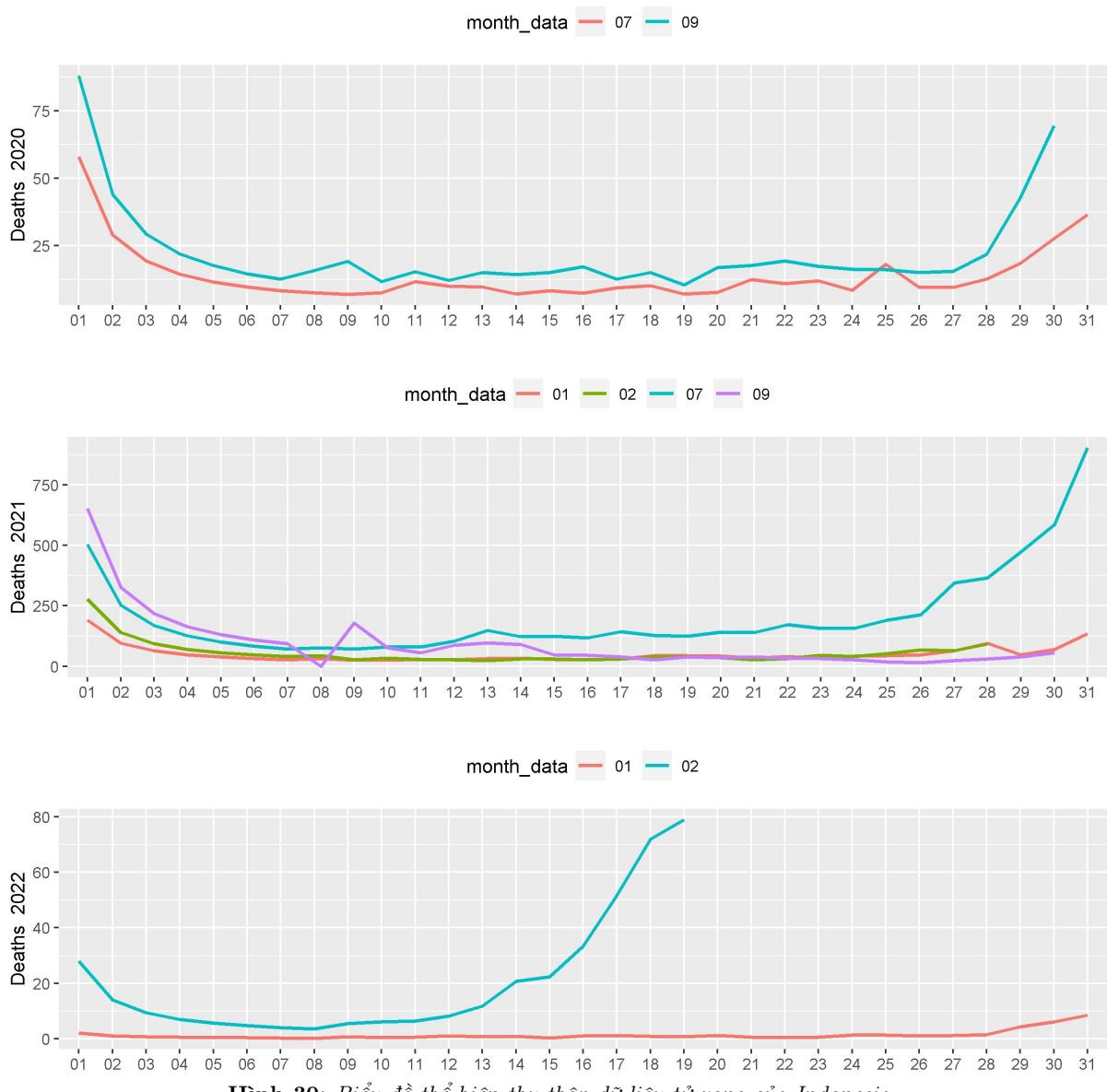




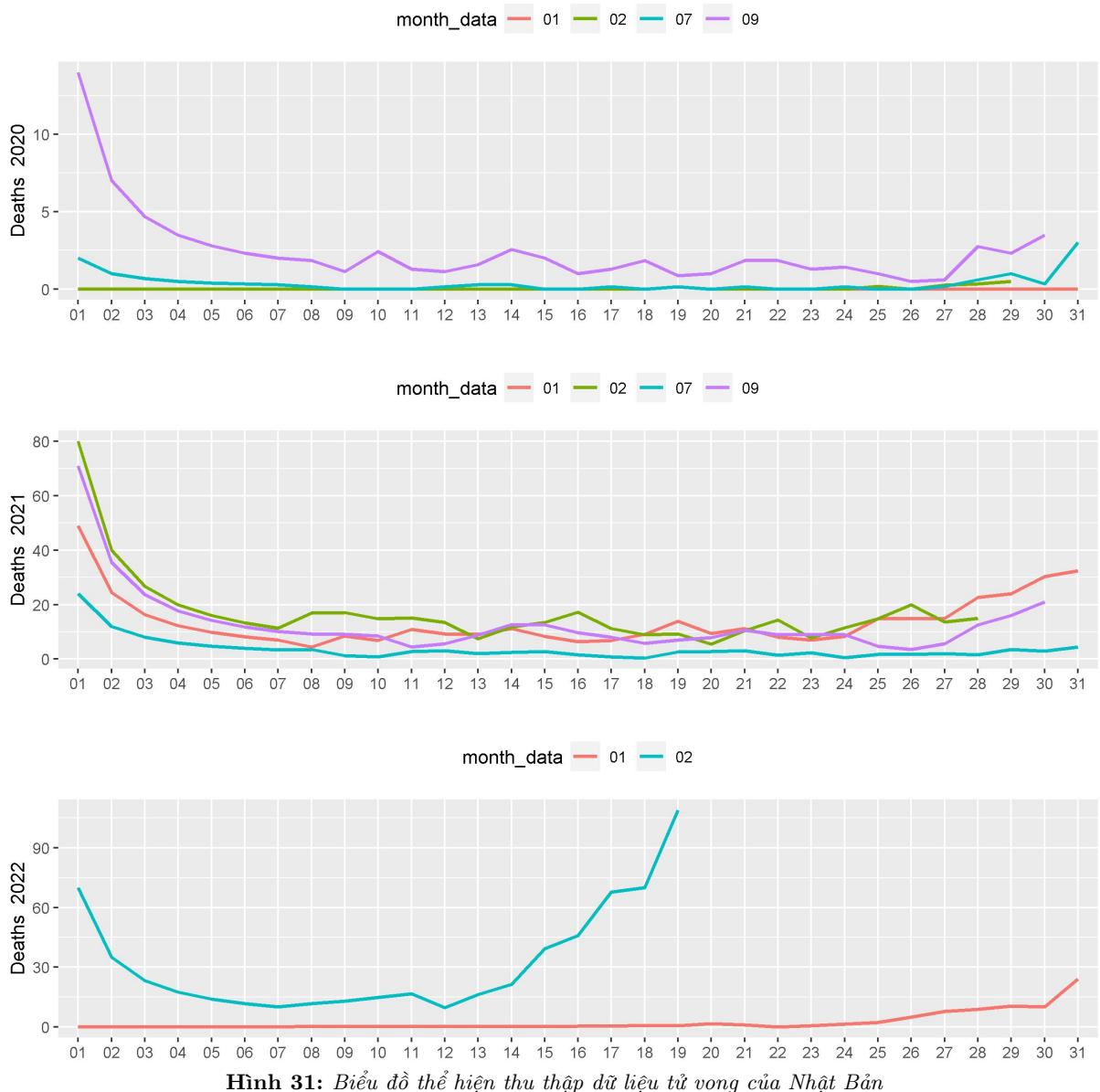
2) Biểu đồ thể hiện thu thập dữ liệu tử vong cho từng tháng

Source code

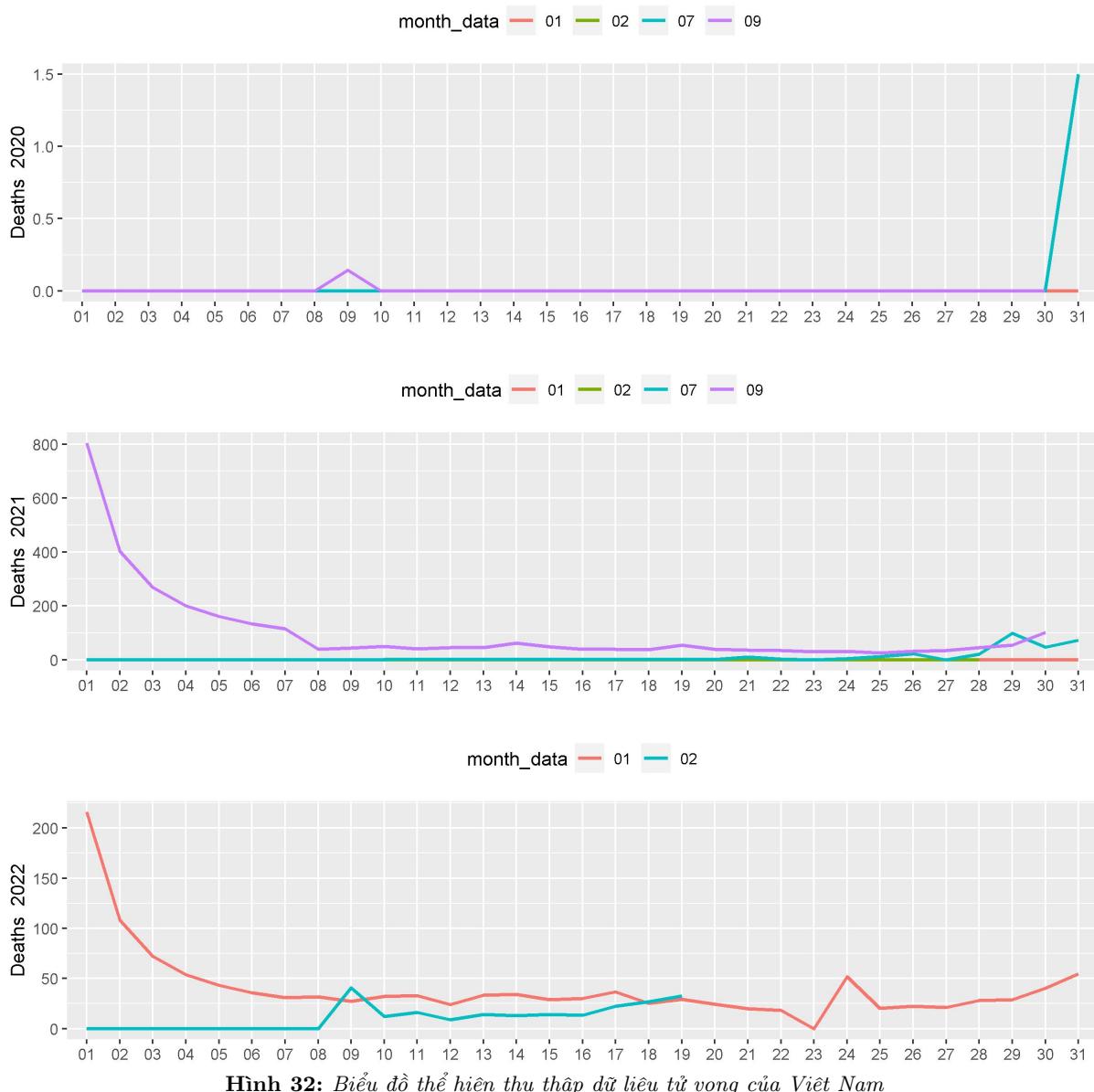
```
#vi2
country_chart("Vietnam","line_chart","2_1_7_9","deaths","vi2","avg")
country_chart("Japan","line_chart","2_1_7_9","deaths","vi2","avg")
country_chart("Indonesia","line_chart","2_1_7_9","deaths","vi2","avg")
```



Hình 30: Biểu đồ thể hiện thu thập dữ liệu tử vong của Indonesia



Hình 31: Biểu đồ thể hiện thu thập dữ liệu tử vong của Nhật Bản

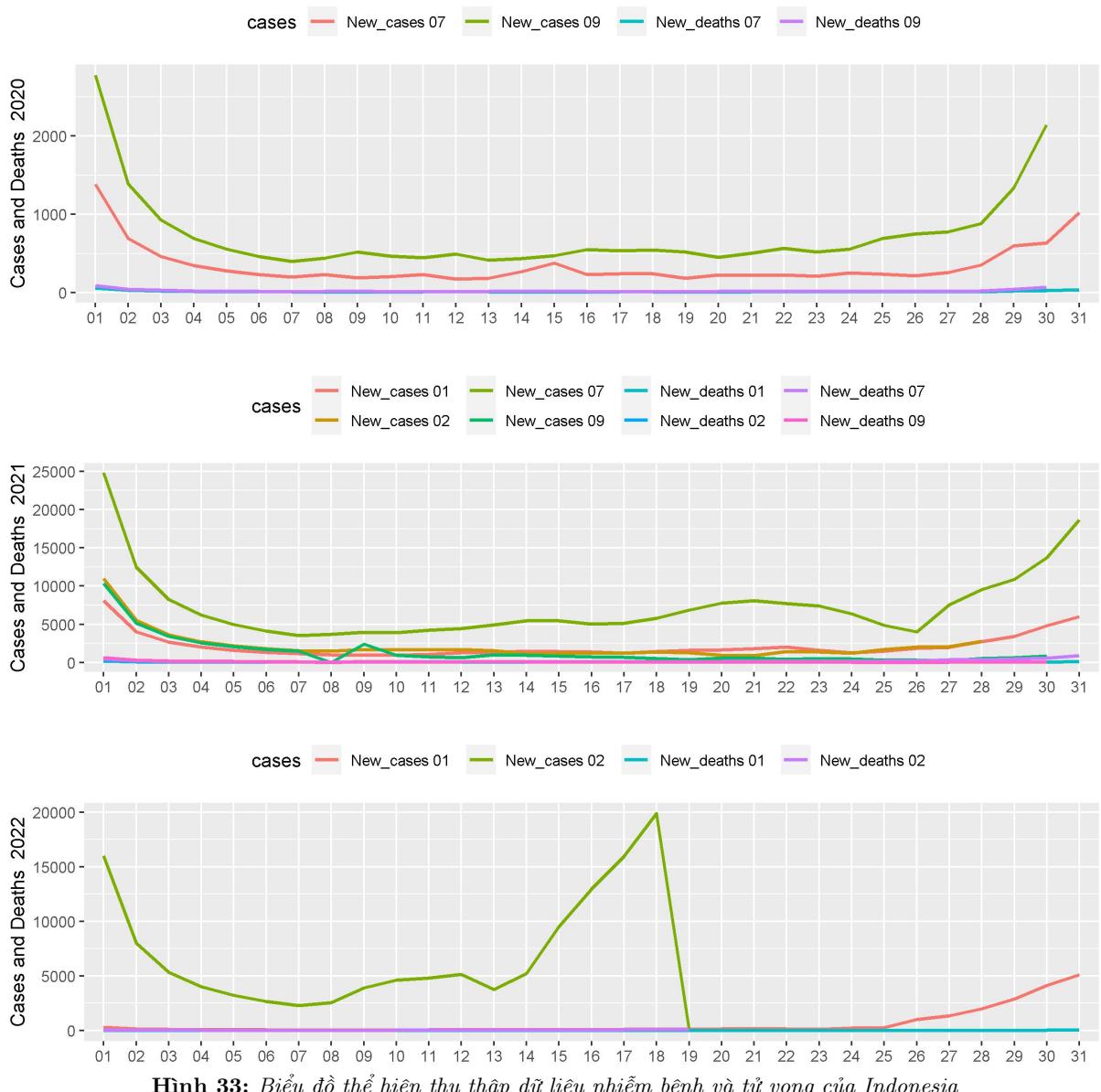


Hình 32: Biểu đồ thể hiện thu thập dữ liệu tử vong của Việt Nam

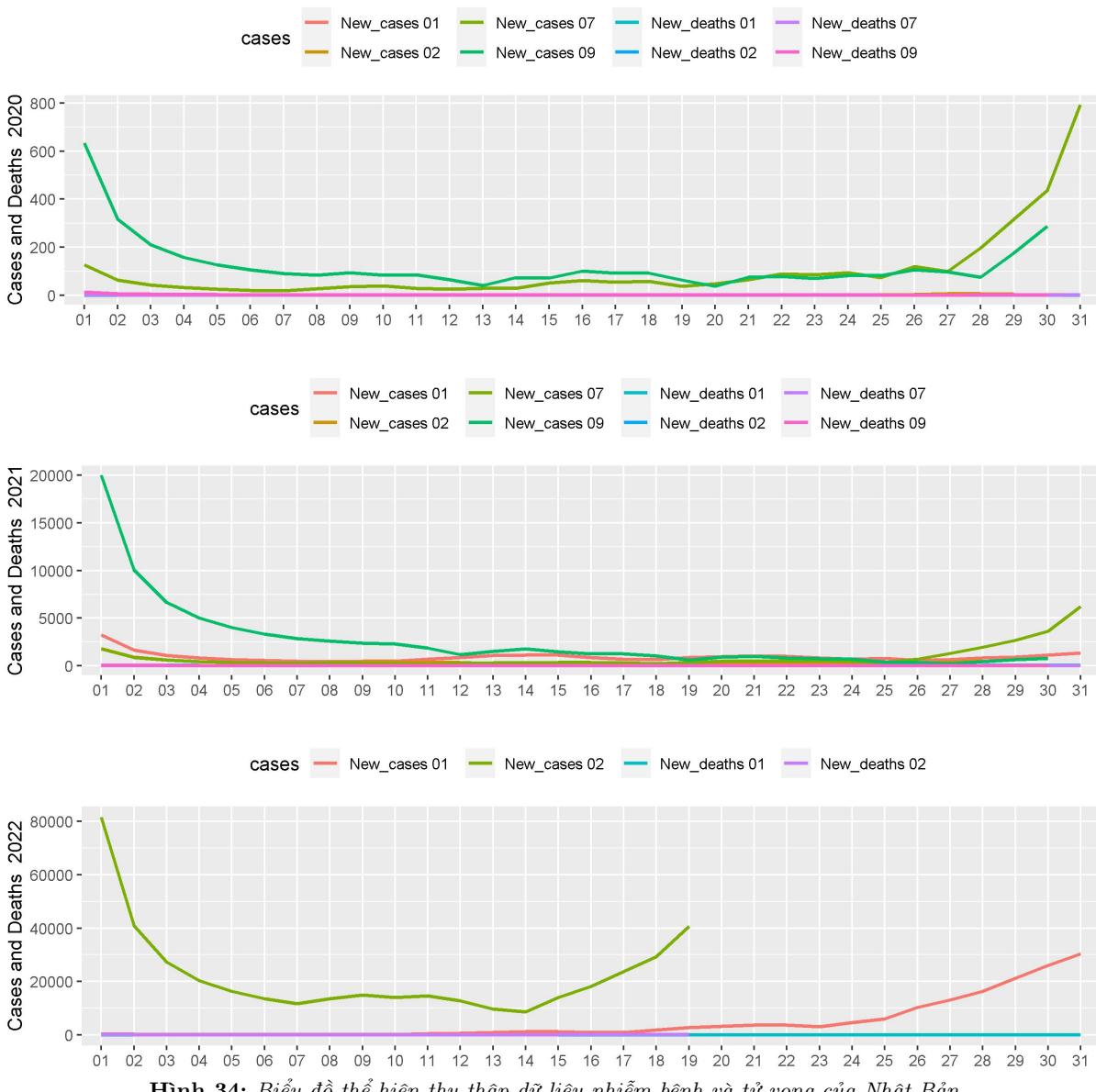
3) Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong cho từng tháng

Source code

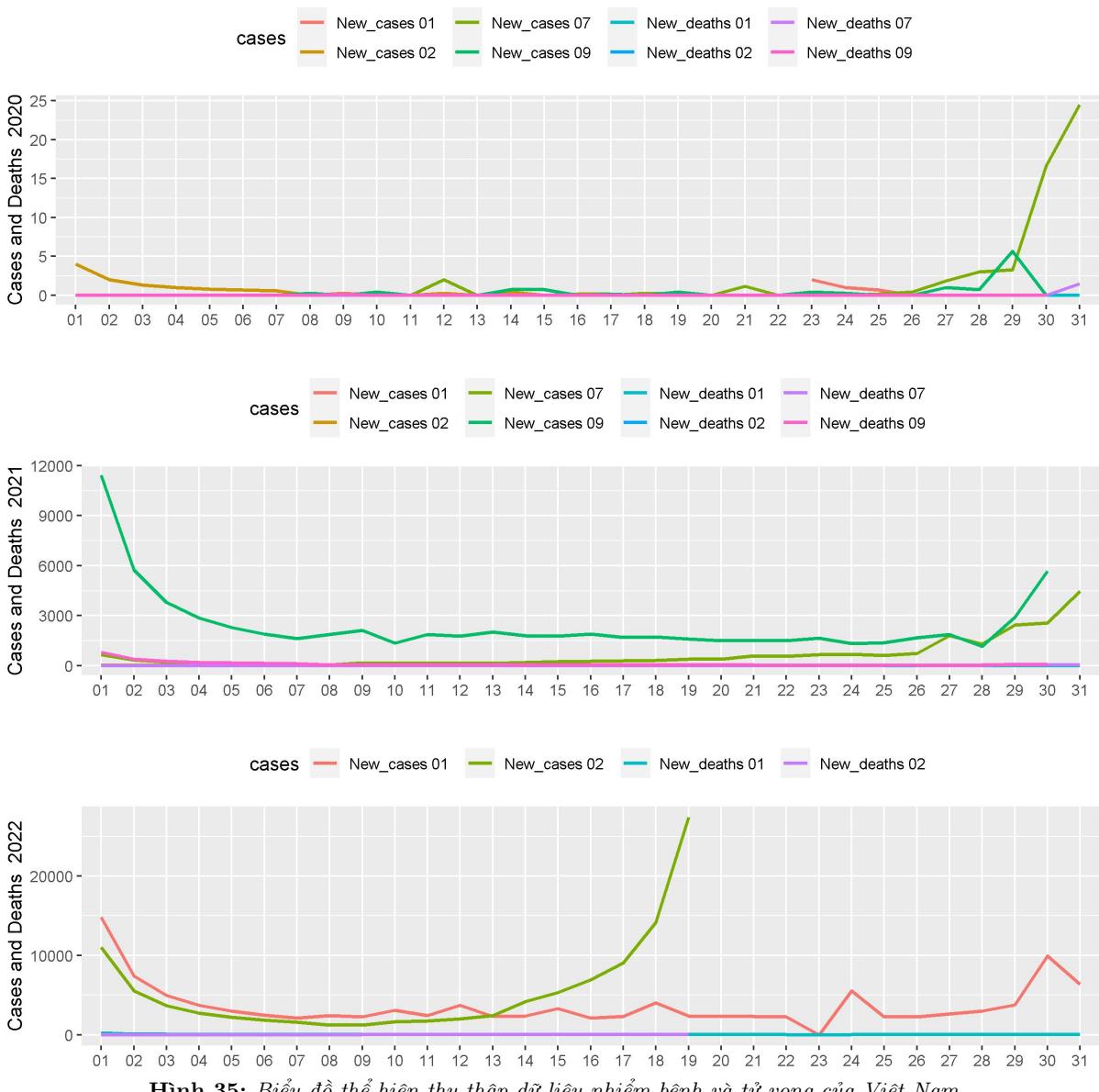
```
#vi3
country_chart("Vietnam", "two_line", "2_1_7_9", "", "vi3", "avg")
country_chart("Japan", "two_line", "2_1_7_9", "", "vi3", "avg")
country_chart("Indonesia", "two_line", "2_1_7_9", "", "vi3", "avg")
```



Hình 33: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong của Indonesia



Hình 34: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong của Nhật Bản

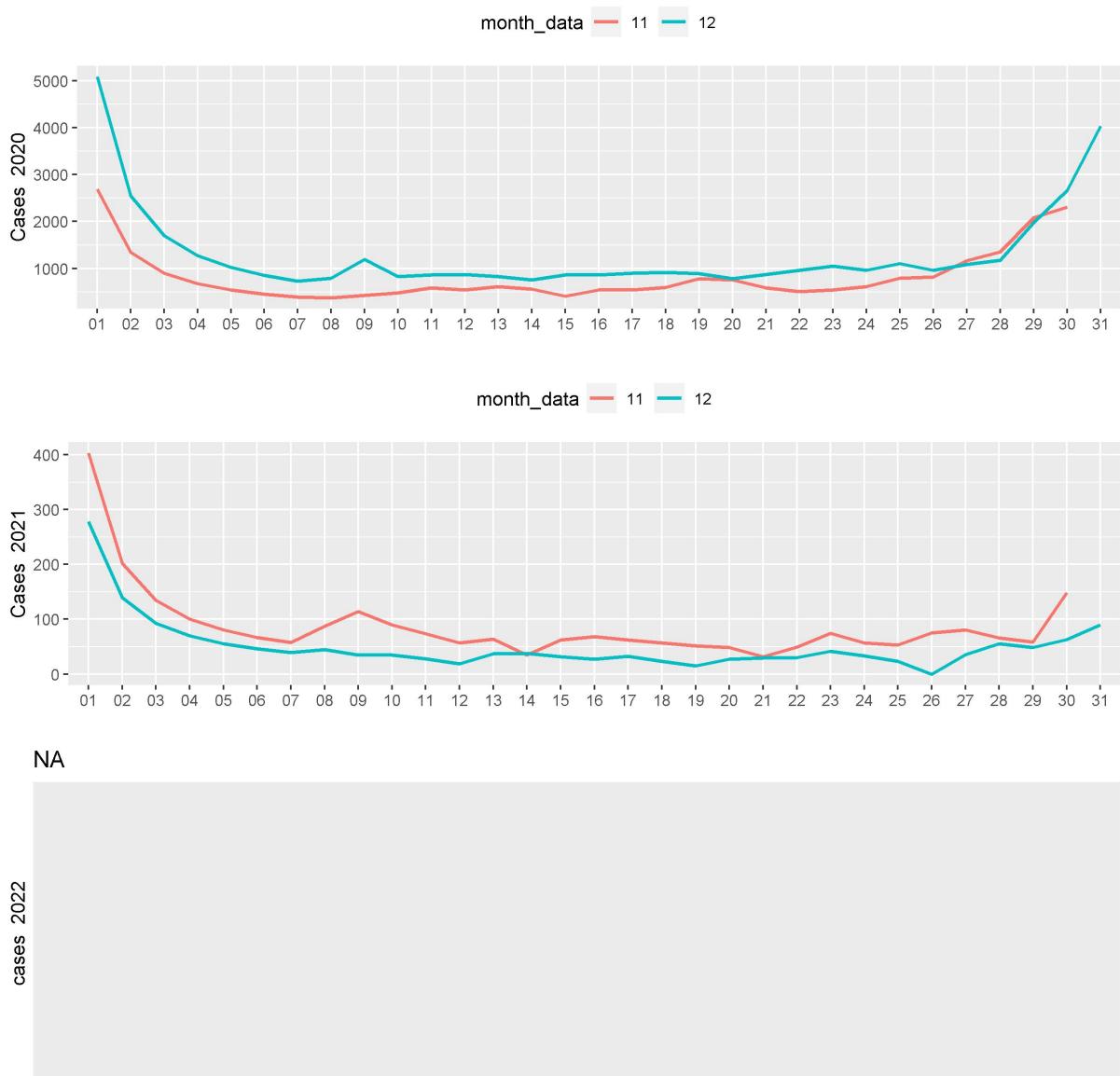


Hình 35: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong của Việt Nam

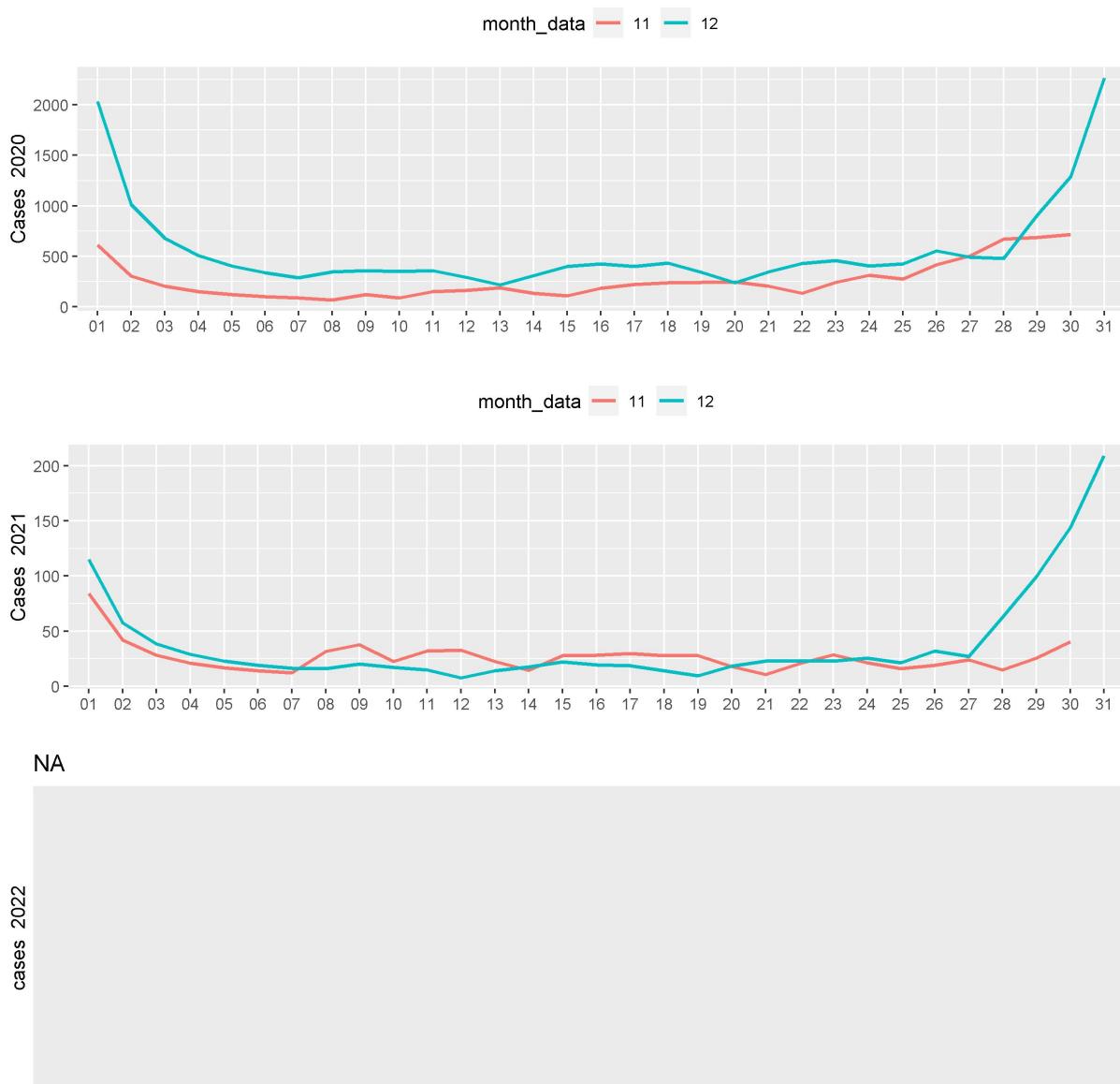
- 4) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh gồm 2 tháng cuối của năm

Source code

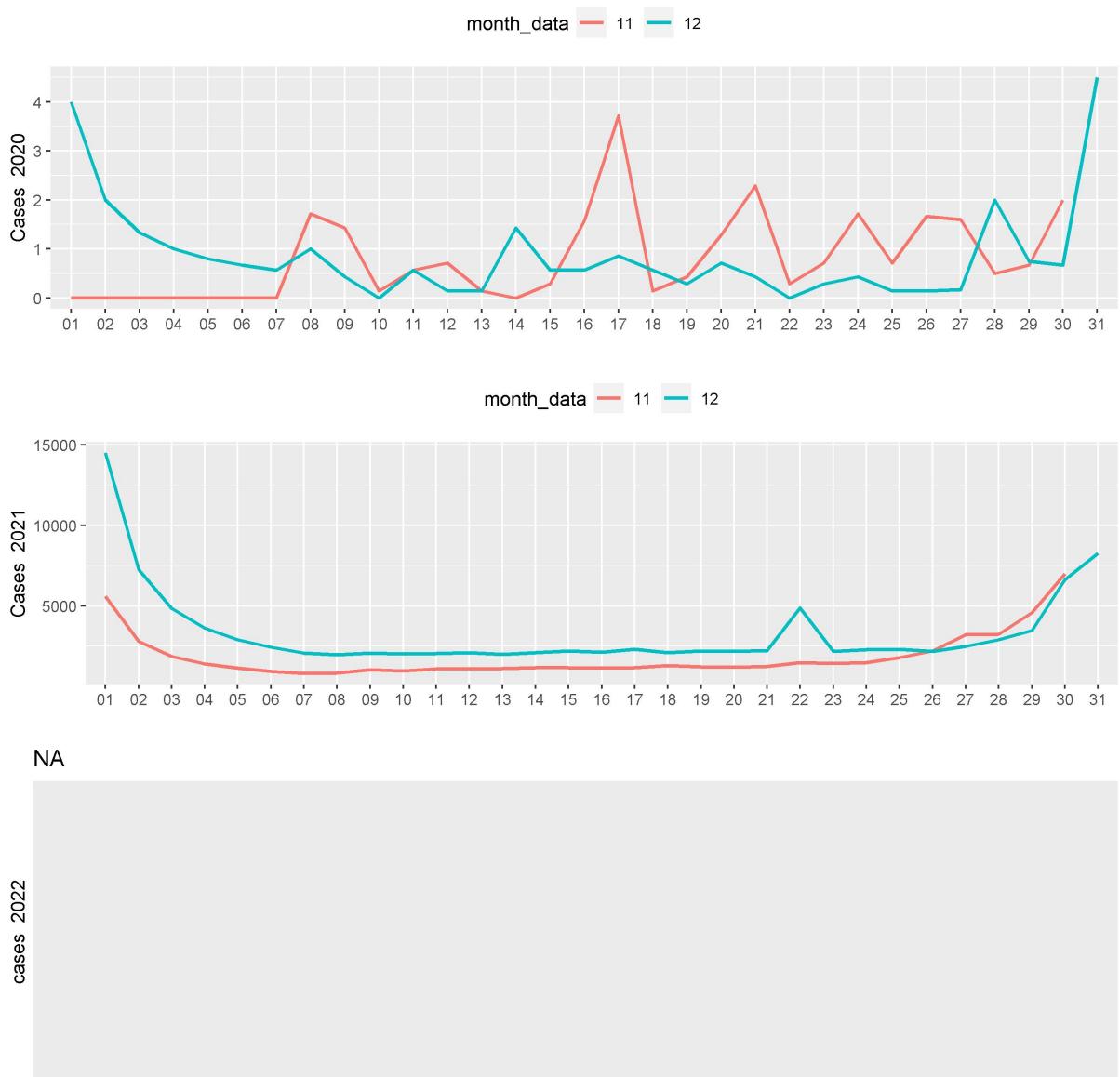
```
#vi4
country_chart("Vietnam","line_chart","11_12","cases","vi4","avg")
country_chart("Japan","line_chart","11_12","cases","vi4","avg")
country_chart("Indonesia","line_chart","11_12","cases","vi4","avg")
```



Hình 36: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo 2 tháng cuối năm của Indonesia



Hình 37: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo 2 tháng cuối năm của Nhật Bản

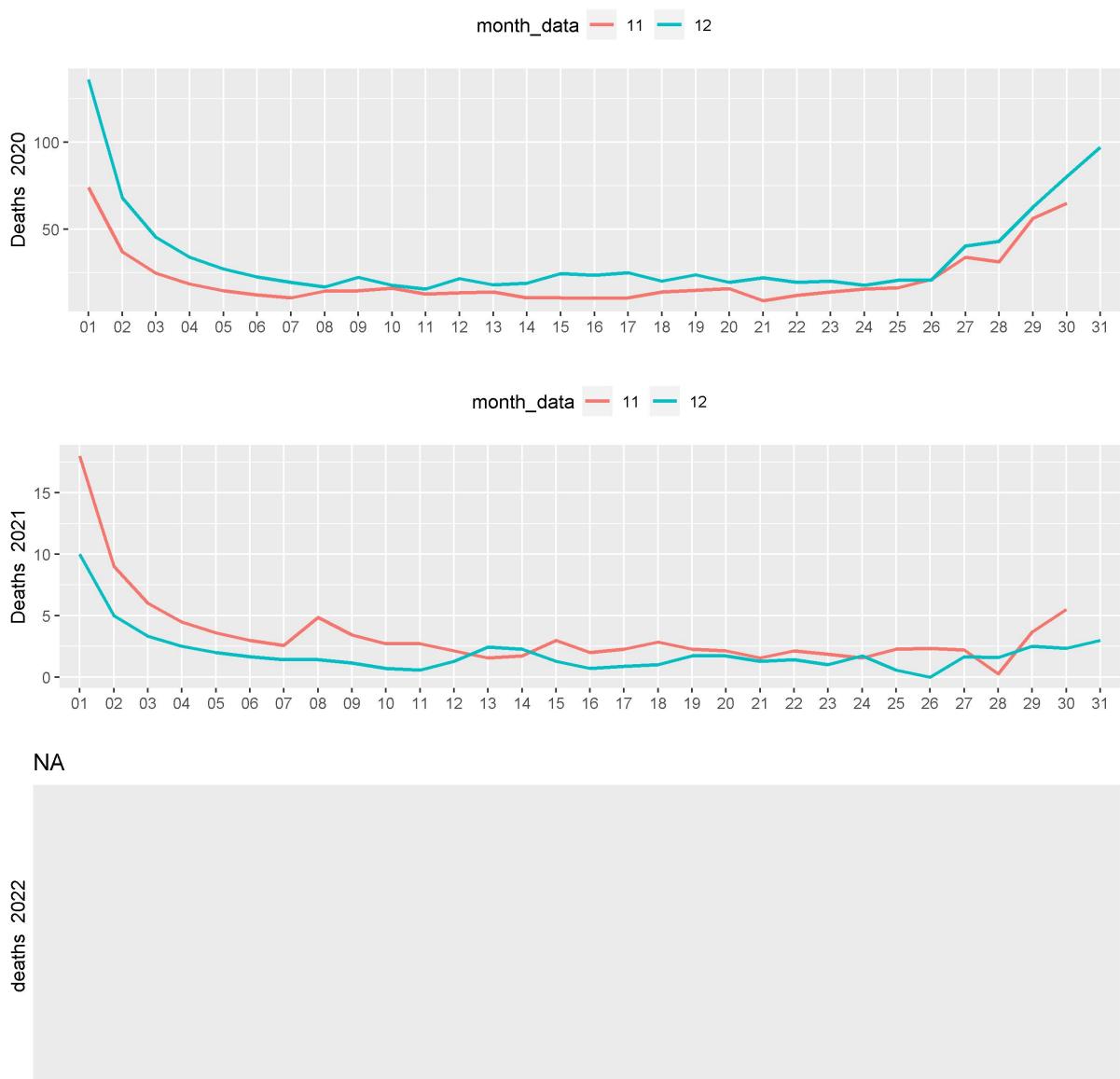


Hình 38: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo 2 tháng cuối năm của Việt Nam

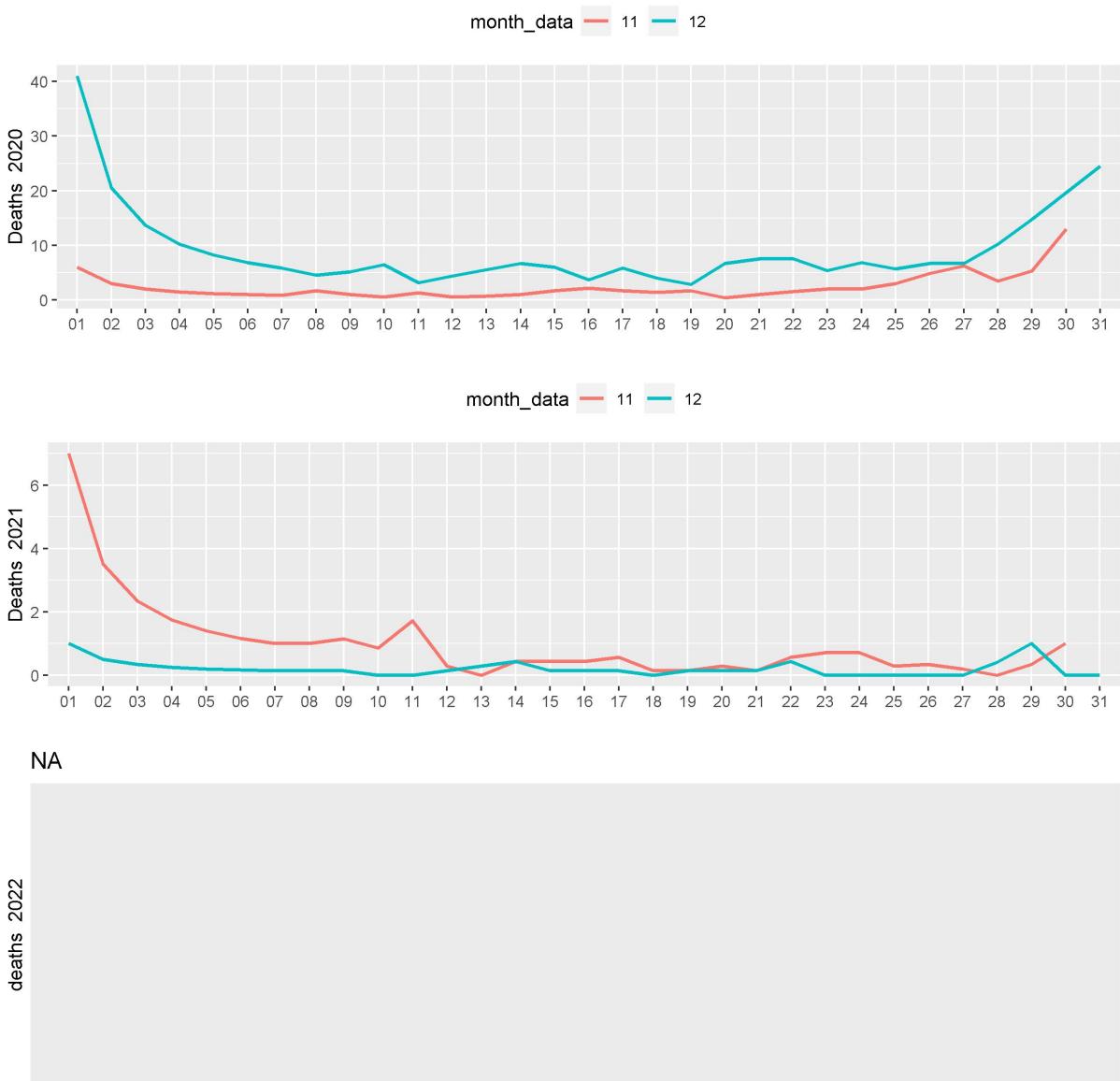
5) Biểu đồ thể hiện thu thập dữ liệu tử vong gồm 2 tháng cuối của năm

Source code

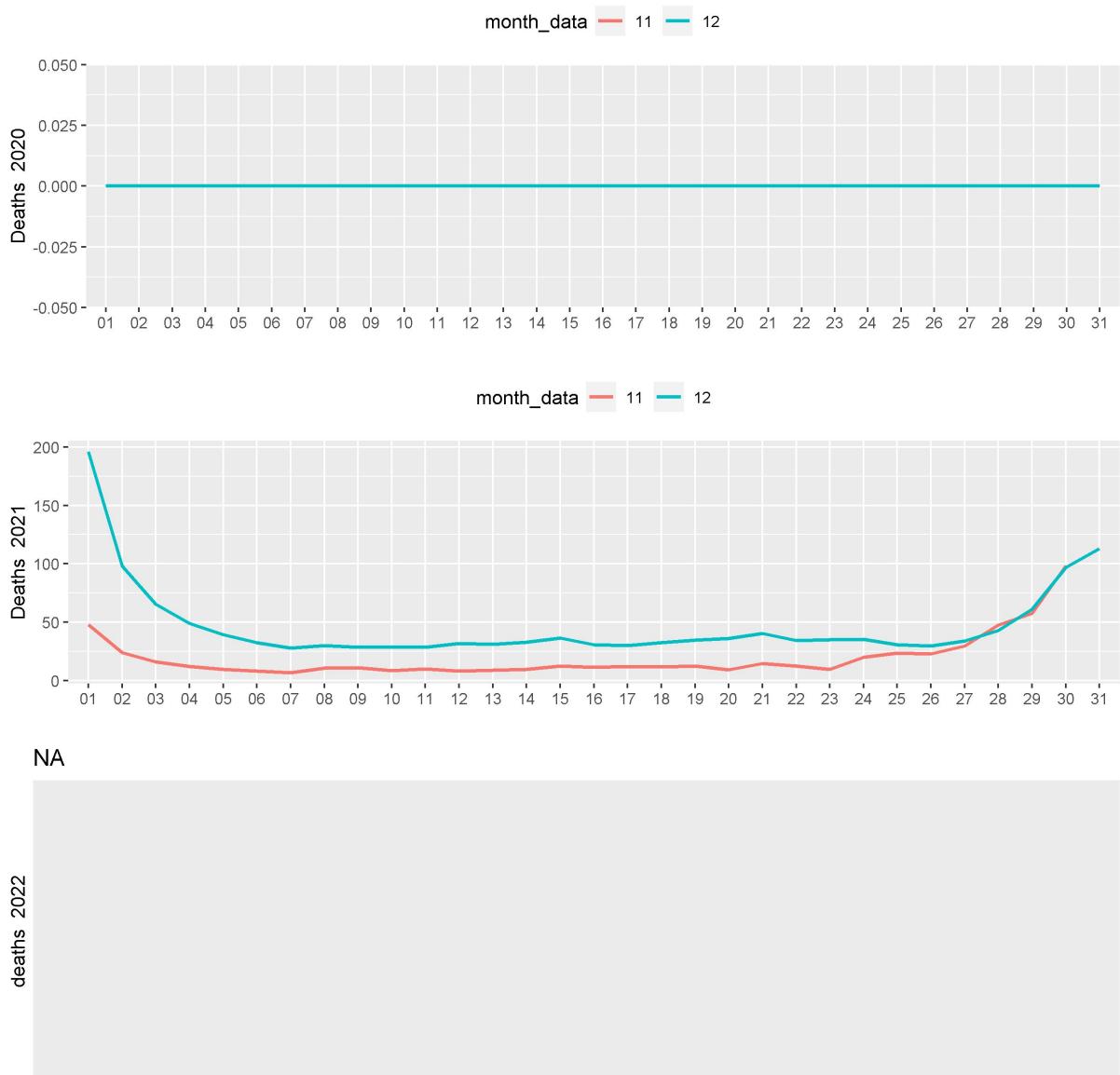
```
#vi5
country_chart("Vietnam","line_chart","11_12","deaths","vi5","avg")
country_chart("Japan","line_chart","11_12","deaths","vi5","avg")
country_chart("Indonesia","line_chart","11_12","deaths","vi5","avg")
```



Hình 39: Biểu đồ thể hiện thu thập dữ liệu tử vong theo 2 tháng cuối năm của Indonesia



Hình 40: Biểu đồ thể hiện thu thập dữ liệu tử vong theo 2 tháng cuối năm của Nhật Bản

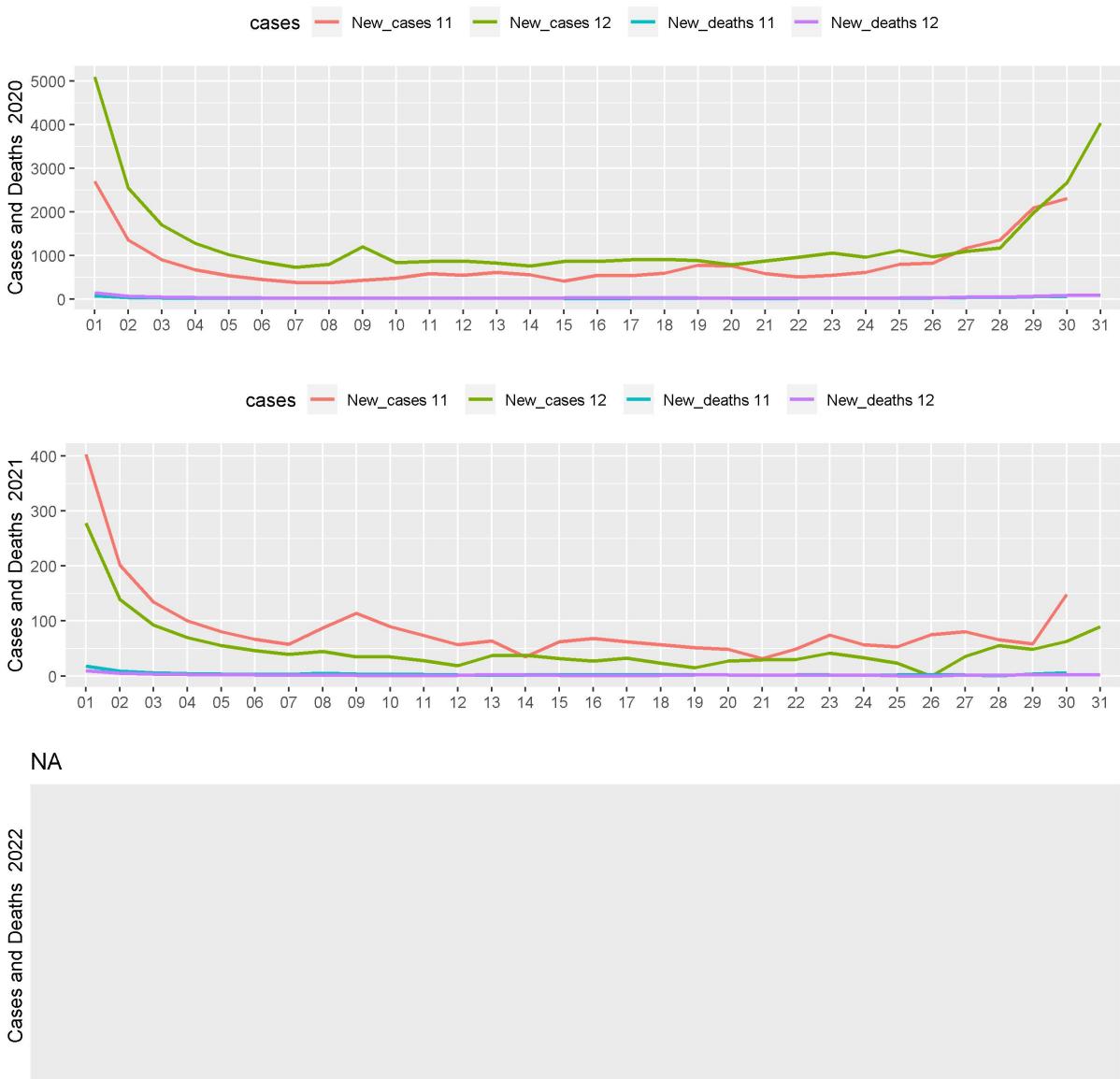


Hình 41: Biểu đồ thể hiện thu thập dữ liệu tử vong theo 2 tháng cuối năm của Việt Nam

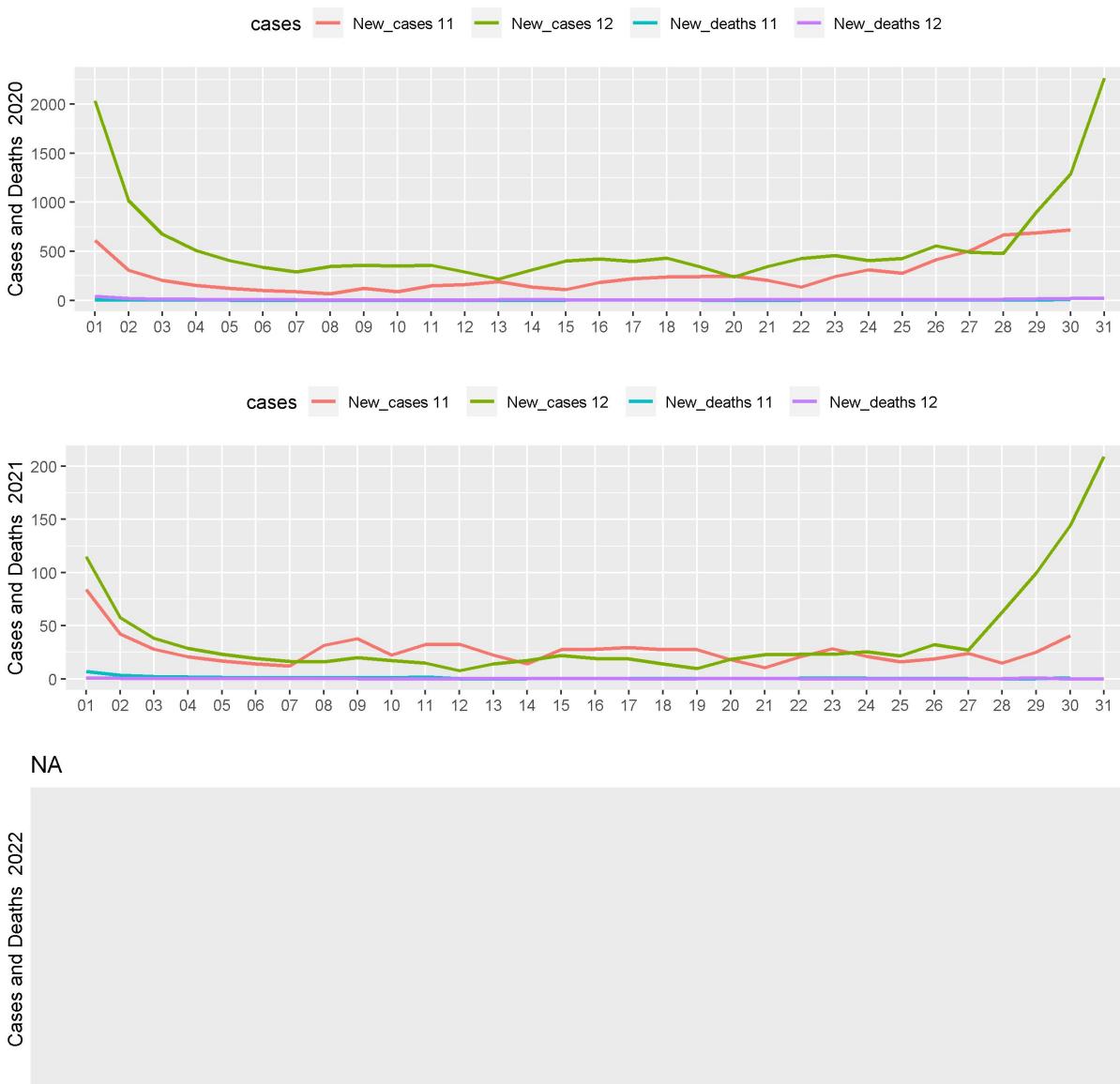
- 6) Biểu đồ thể hiện thu thập dữ liệu gồm nhiễm bệnh và tử vong gồm 2 tháng cuối của năm

Source code

```
#vi6
country_chart("Vietnam","two_line","11_12","","vi6","avg")
country_chart("Japan","two_line","11_12","","vi6","avg")
country_chart("Indonesia","two_line","11_12","","vi6","avg")
```



Hình 42: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong theo 2 tháng cuối năm của Indonesia



Hình 43: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong theo 2 tháng cuối năm của Nhật Bản

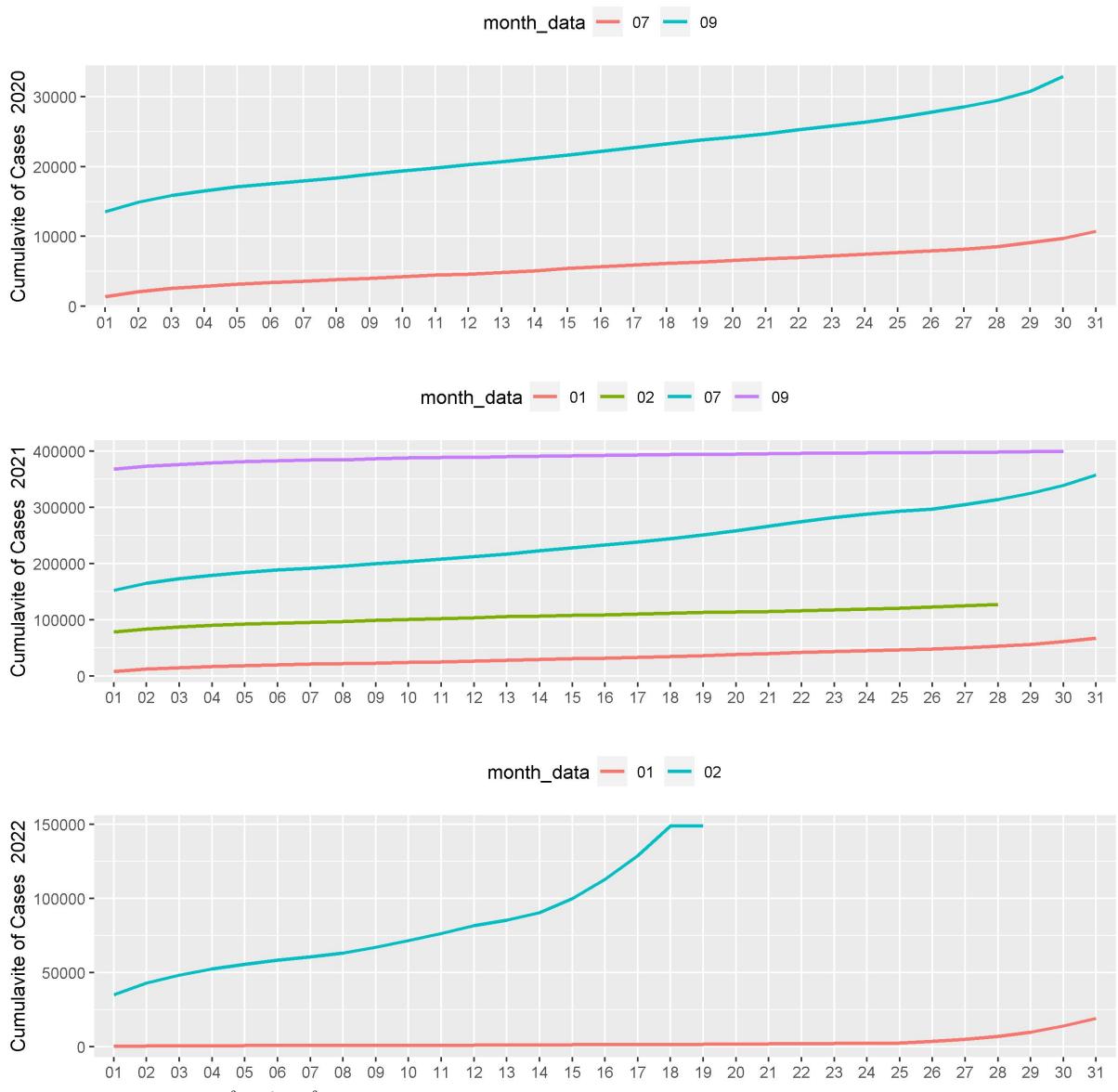


Hình 44: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh và tử vong theo 2 tháng cuối năm của Việt Nam

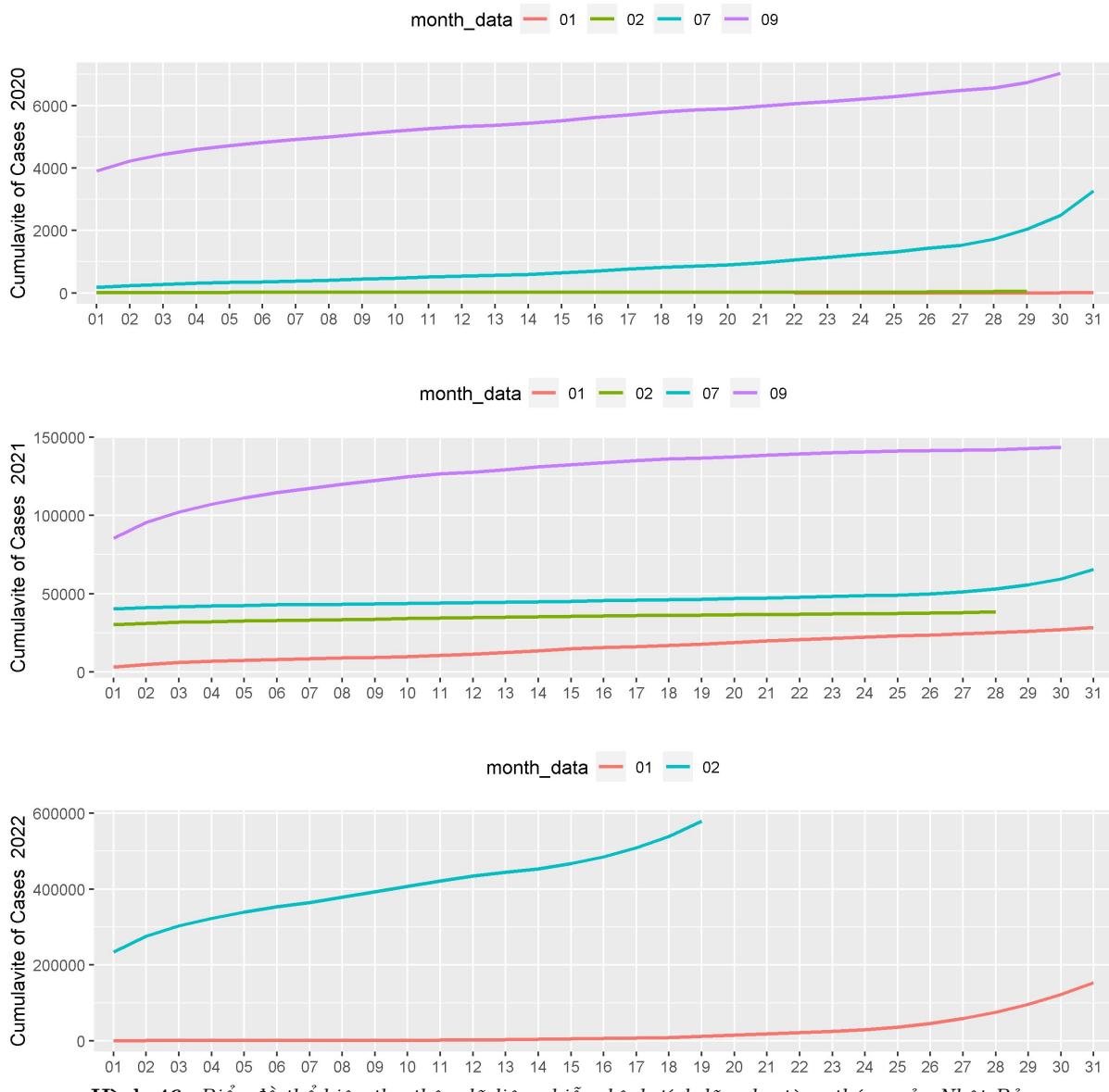
7) Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy cho từng tháng

Source code

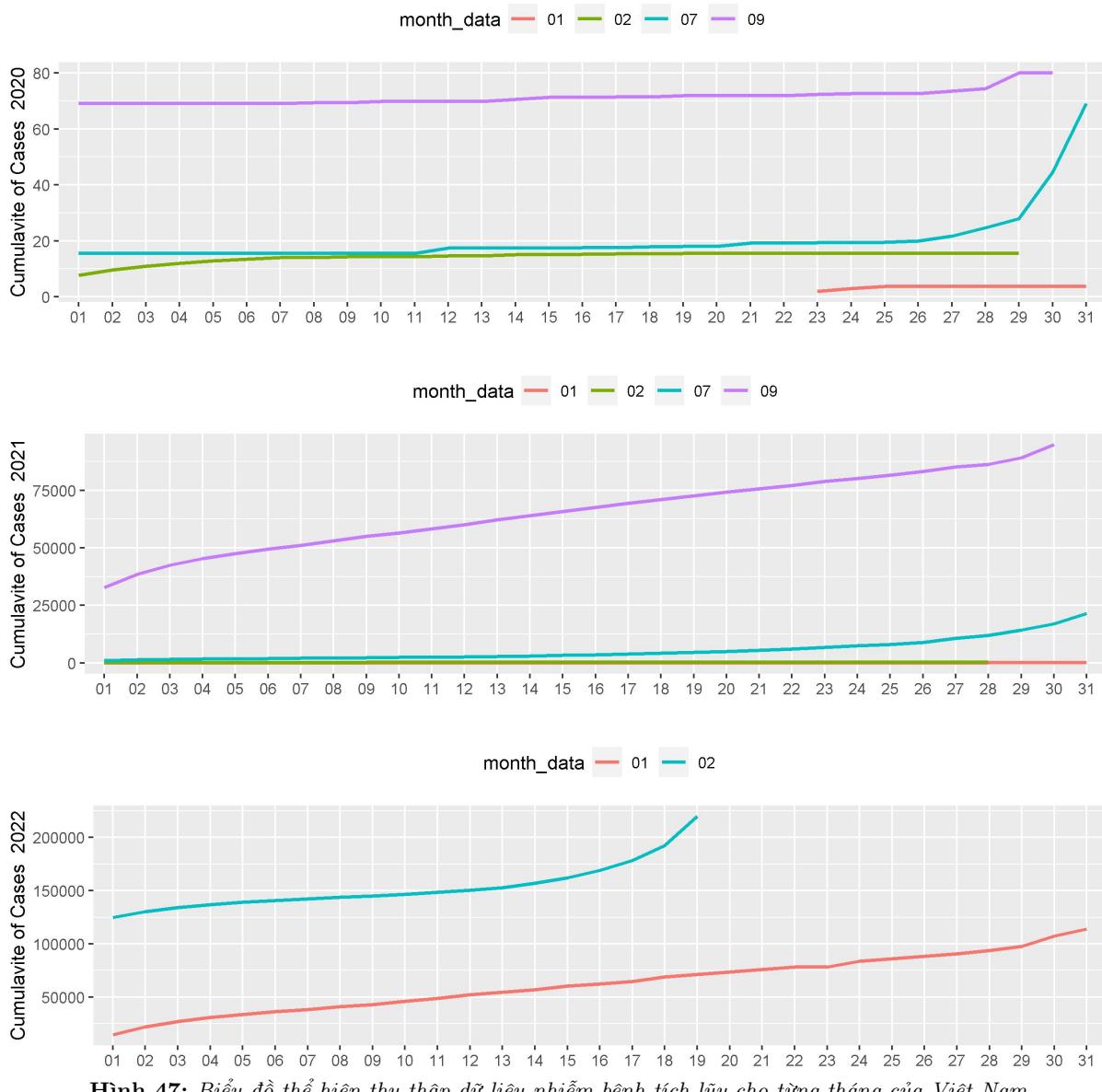
```
#vi7
country_chart("Vietnam", "cum", "2_1_7_9", "cases", "vi7", "avg")
country_chart("Japan", "cum", "2_1_7_9", "cases", "vi7", "avg")
country_chart("Indonesia", "cum", "2_1_7_9", "cases", "vi7", "avg")
```



Hình 45: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy cho từng tháng của Indonesia



Hình 46: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy cho từng tháng của Nhật Bản

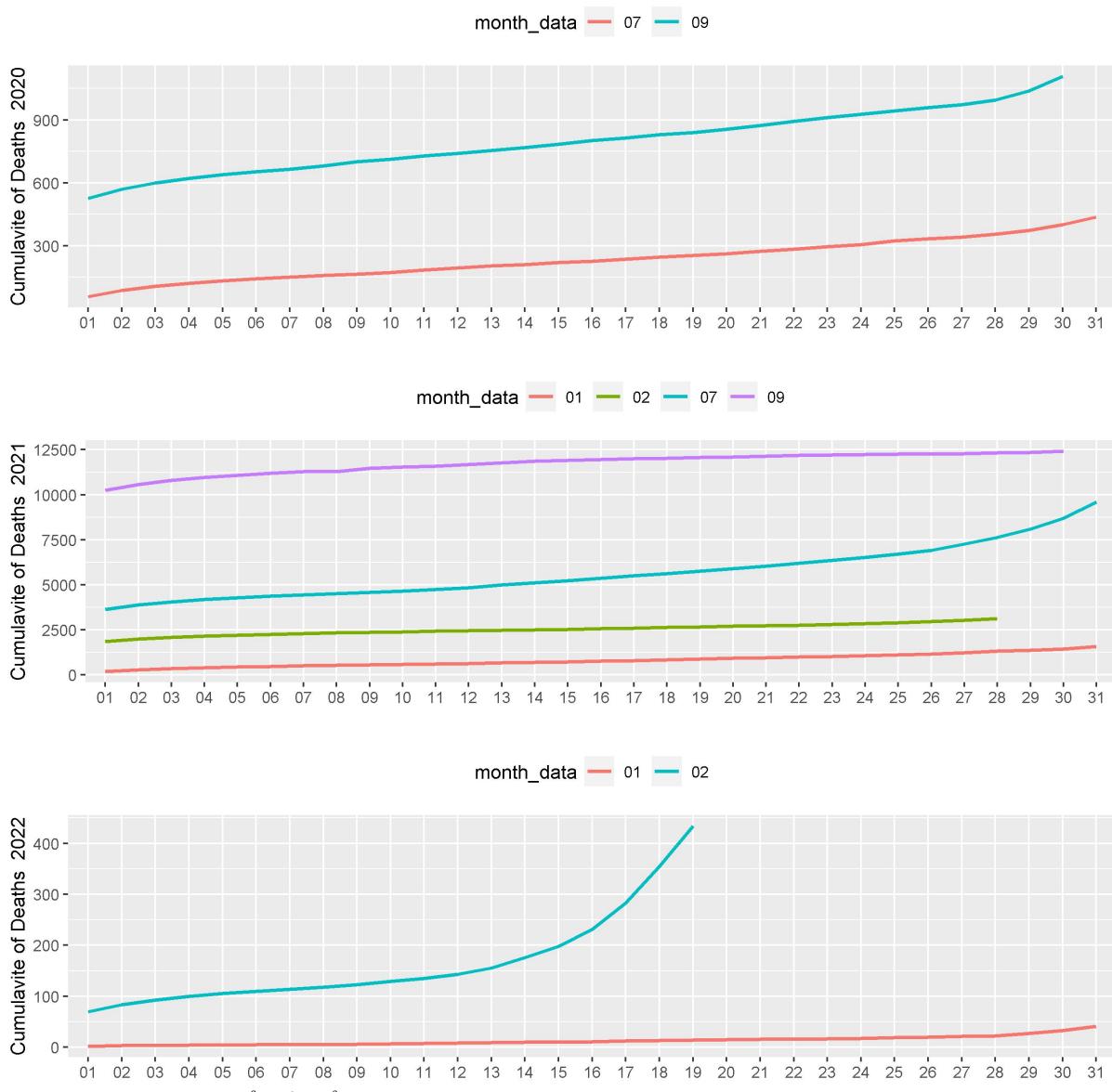


Hình 47: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy cho từng tháng của Việt Nam

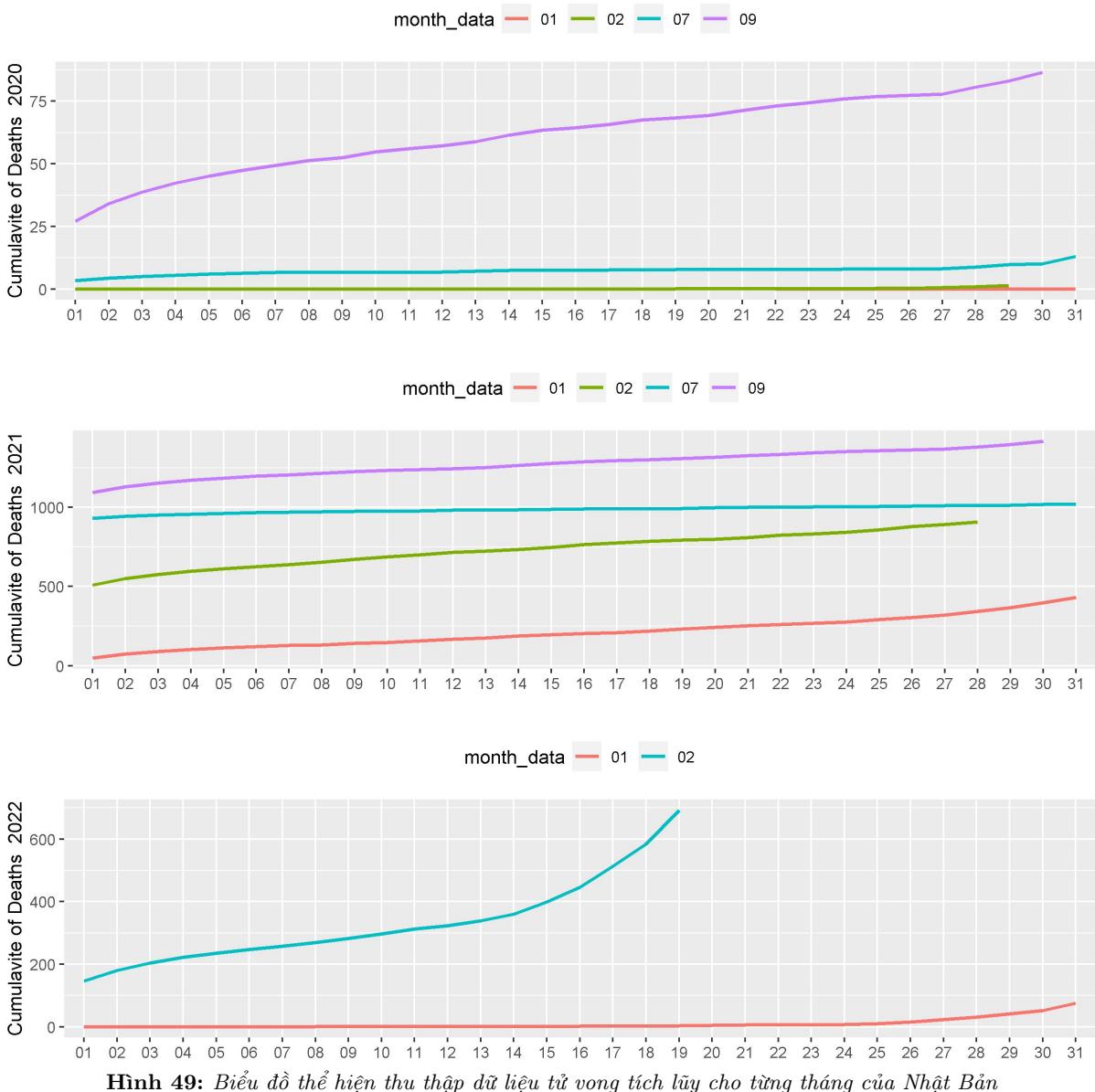
8) Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy cho từng tháng

Source code

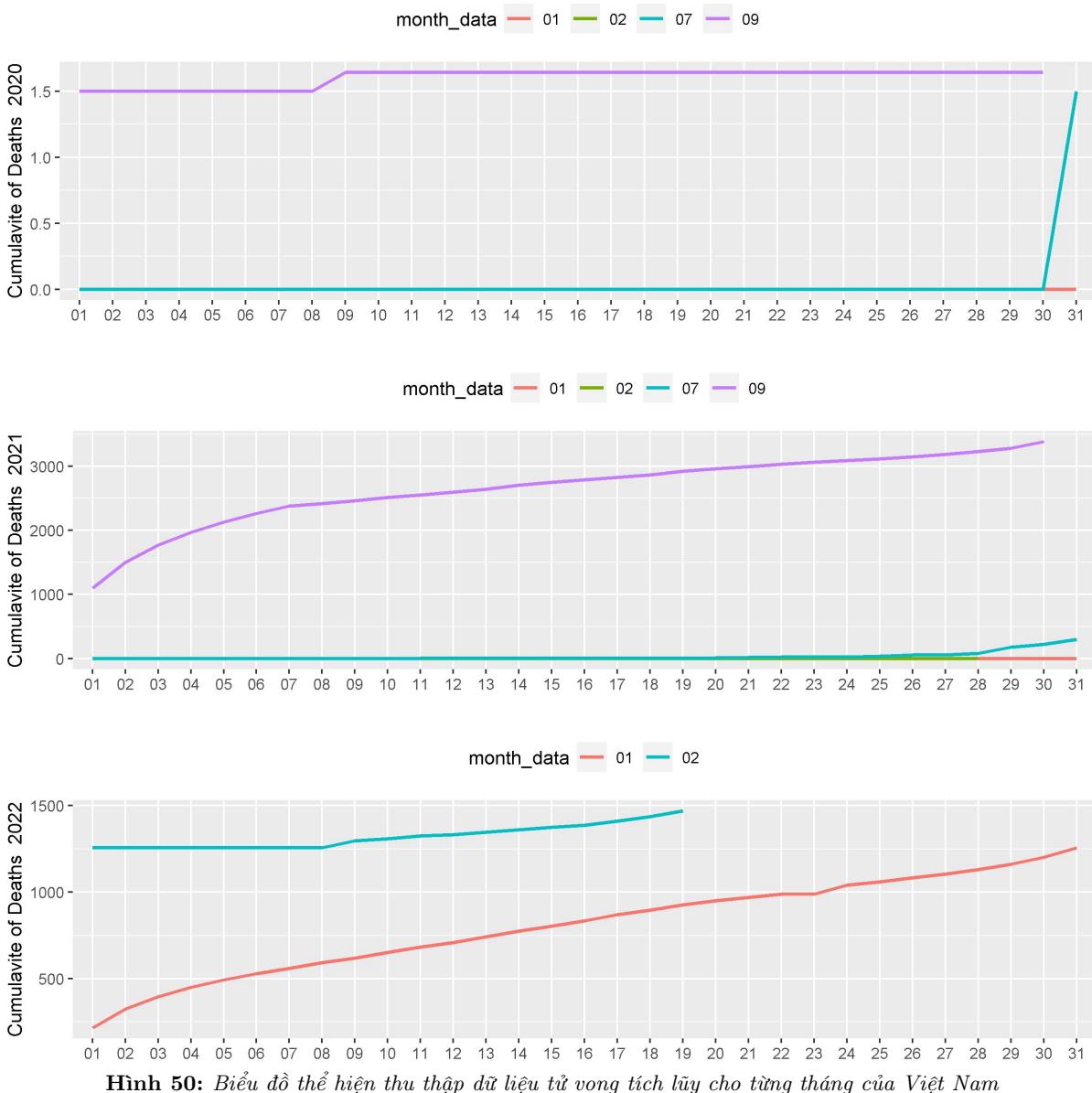
```
#vi8
country_chart ("Vietnam", "cum", "2_1_7_9", "deaths", "vi8", "avg")
country_chart ("Japan", "cum", "2_1_7_9", "deaths", "vi8", "avg")
country_chart ("Indonesia", "cum", "2_1_7_9", "deaths", "vi8", "avg")
```



Hình 48: Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy cho từng tháng của Indonesia



Hình 49: Biểu đồ thể hiện thu thập dữ liệu tử vong tích lũy cho từng tháng của Nhật Bản



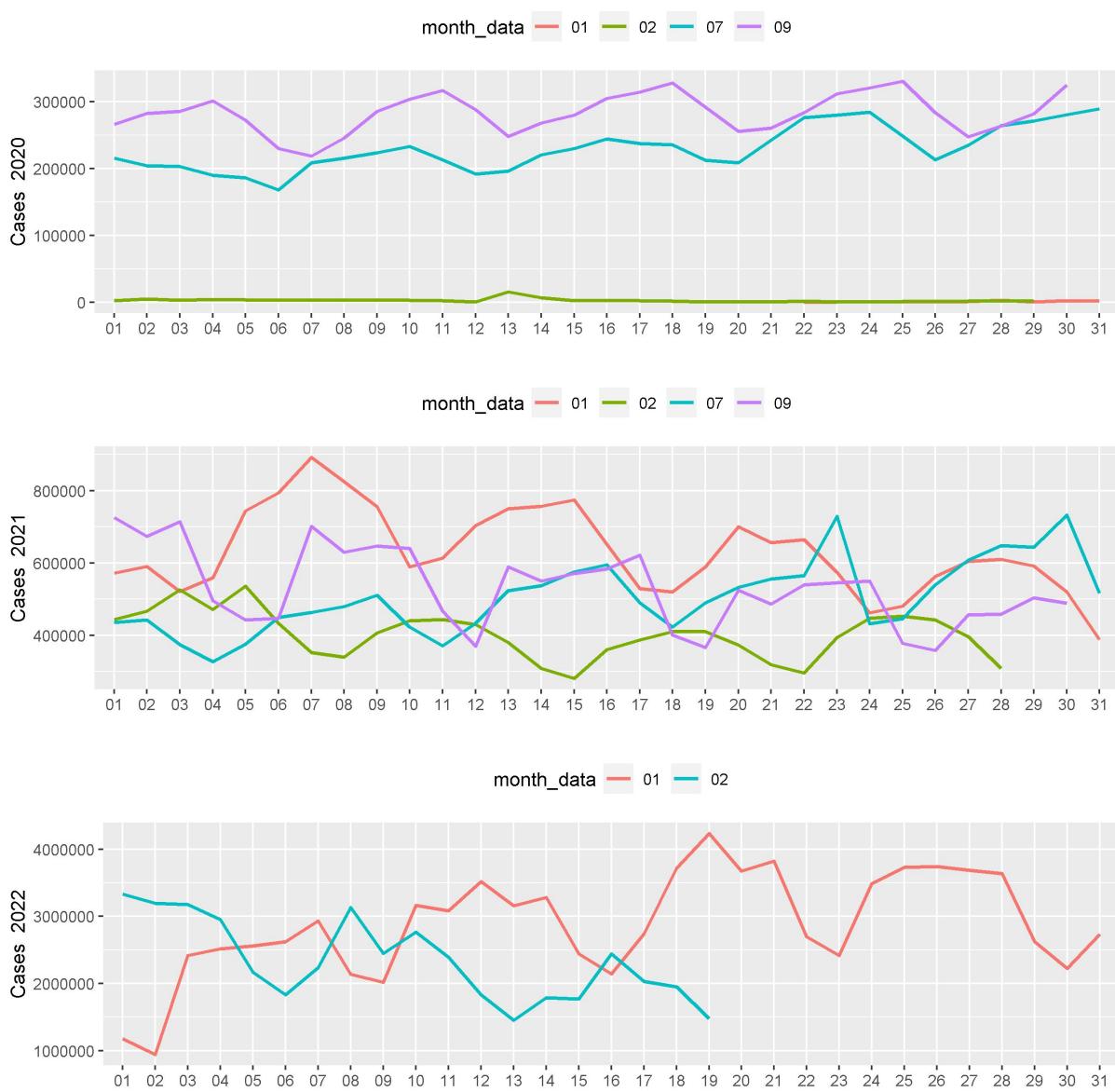
vii) Nhóm câu hỏi liên quan đến tất cả quốc gia theo thời gian là tháng

- Trên từng năm hãy vẽ biểu đồ thể hiện trực Ox là thời gian, trực Oy là nhiễm bệnh/tử vong. Hãy dùng 4 ký số của mã đê 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

Source code

```
#vii1
country_chart("World", "line_chart", "2_1_7_9", "cases", "vii1")
```

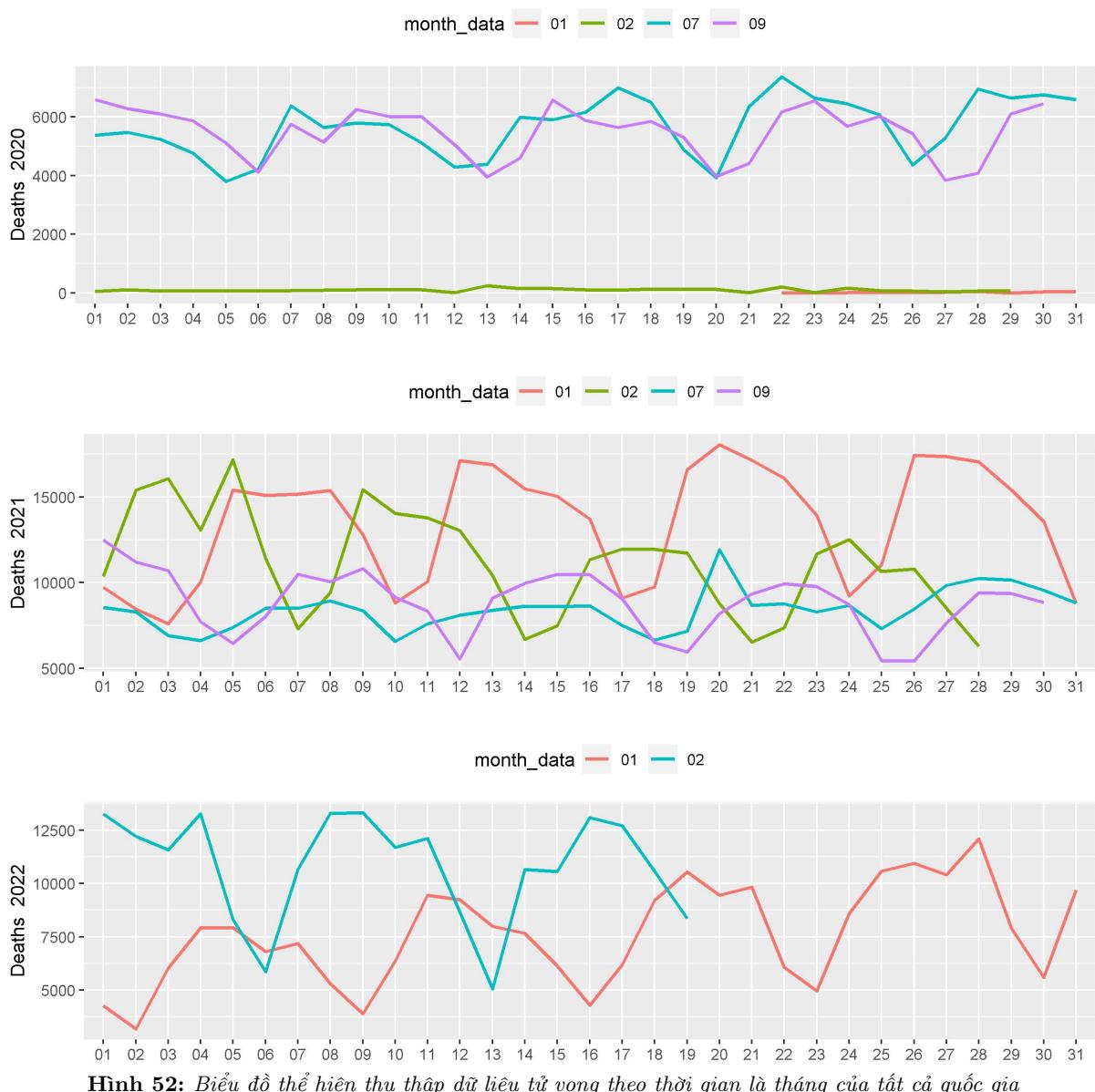


Hình 51: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo thời gian là tháng của tất cả quốc gia

2) Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là tháng của tất cả quốc gia

Source code

```
#vii2
country_chart("World", "line_chart", "2_1_7_9", "deaths", "vii2")
```

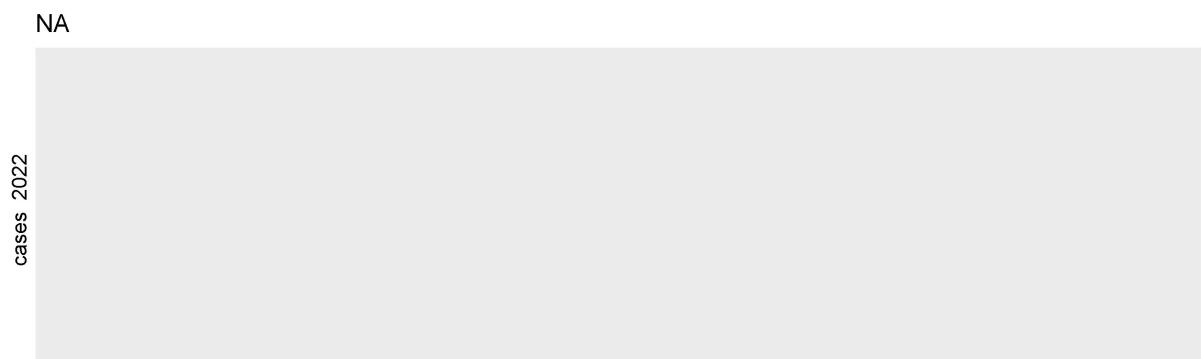
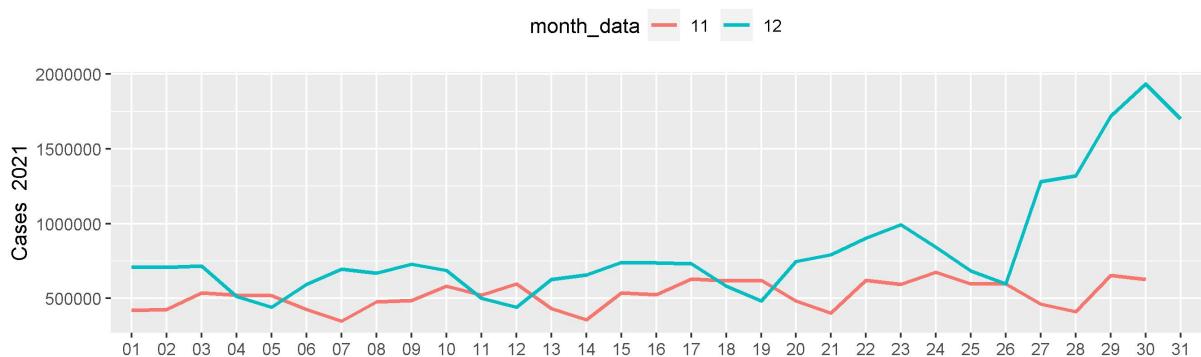
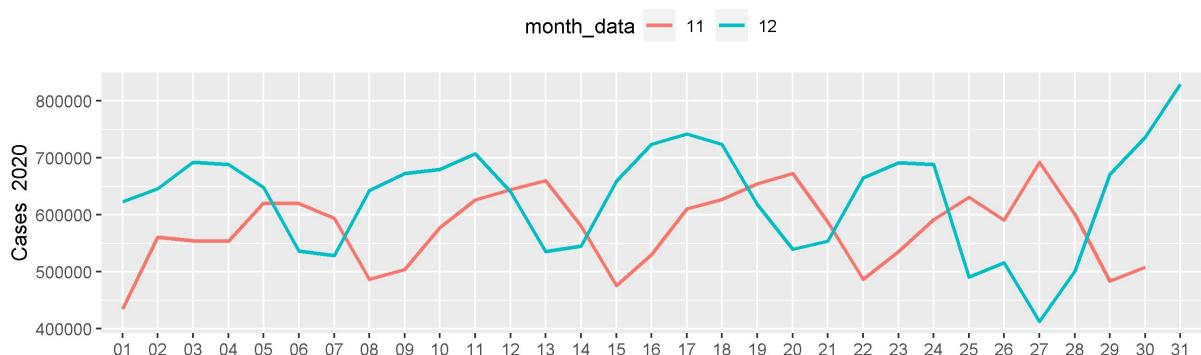


Hình 52: Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là tháng của tất cả quốc gia

- 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

Source code

```
#vii3
country_chart("World", "line_chart", "11_12", "cases", "vii3")
```

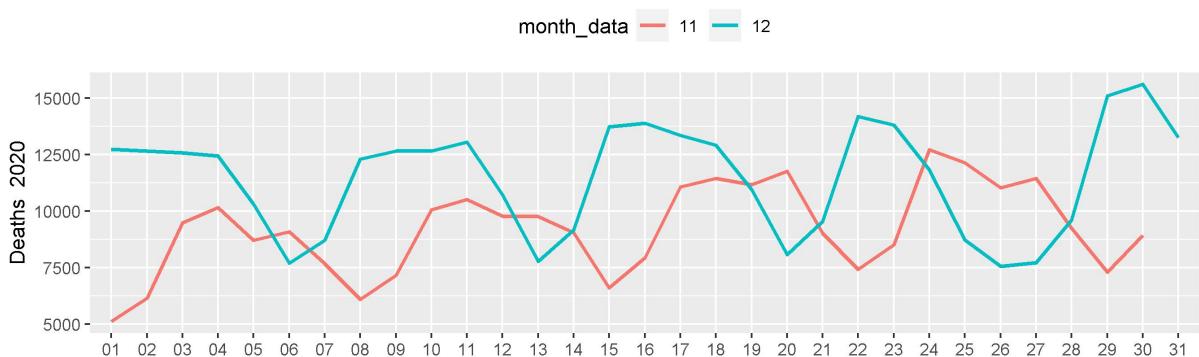


Hình 53: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo thời gian là 2 tháng cuối của năm của tất cả quốc gia

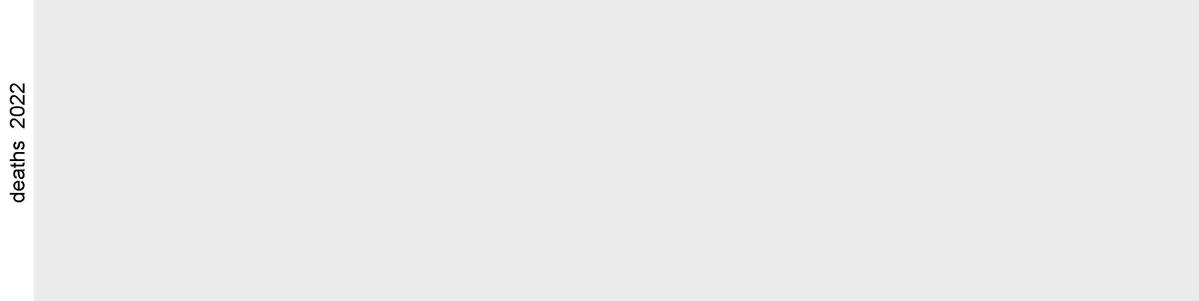
- 4) Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là 2 tháng cuối của năm của tất cả quốc gia

Source code

```
#vii4
country_chart("World", "line_chart", "11_12", "deaths", "vii4")
```



NA

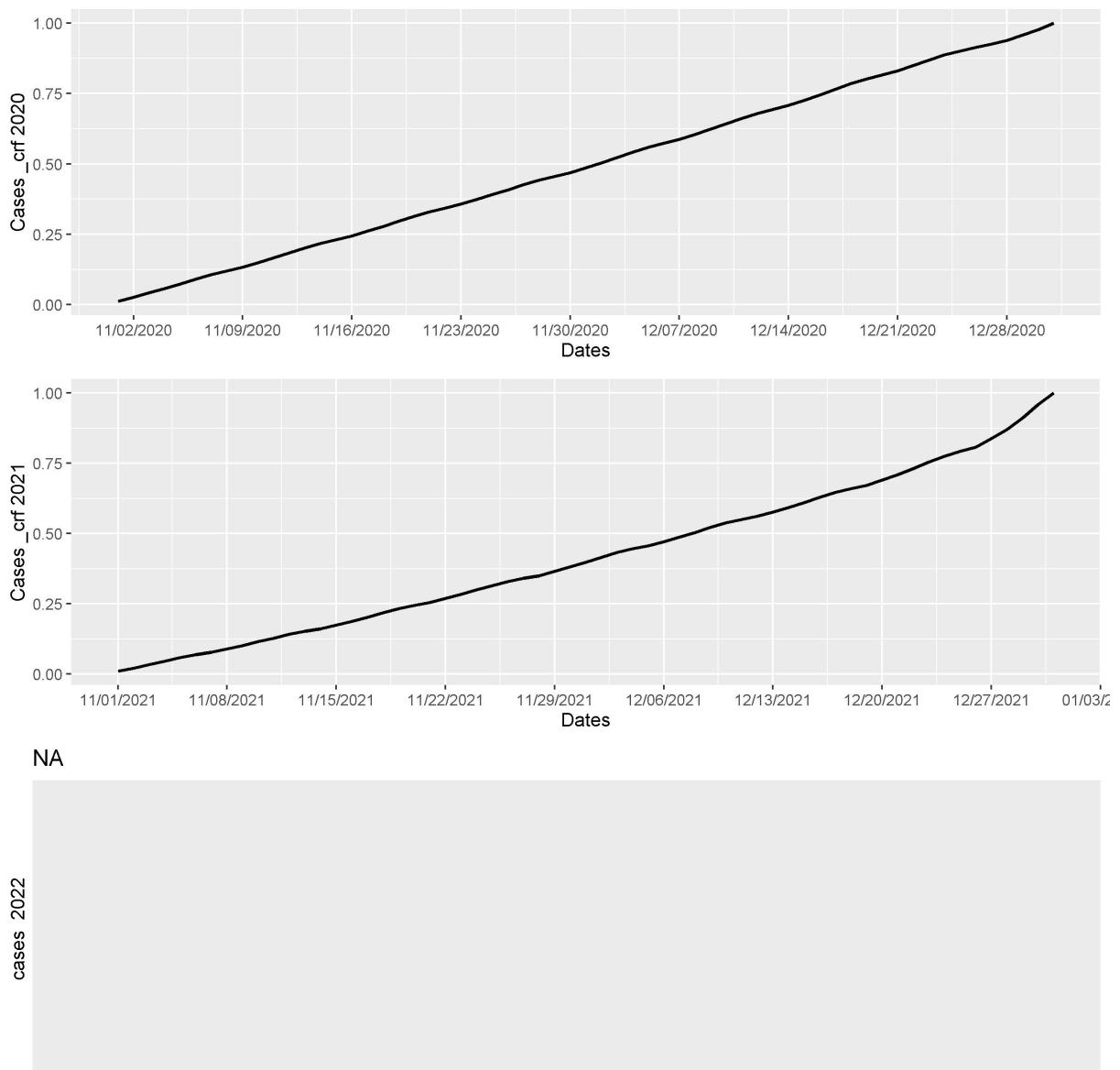


Hình 54: Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là 2 tháng cuối của năm của tất cả quốc gia

- 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

Source code

```
#vii5  
country_chart ("World", "cum_rel", "11_12", "cases", "vii5")
```

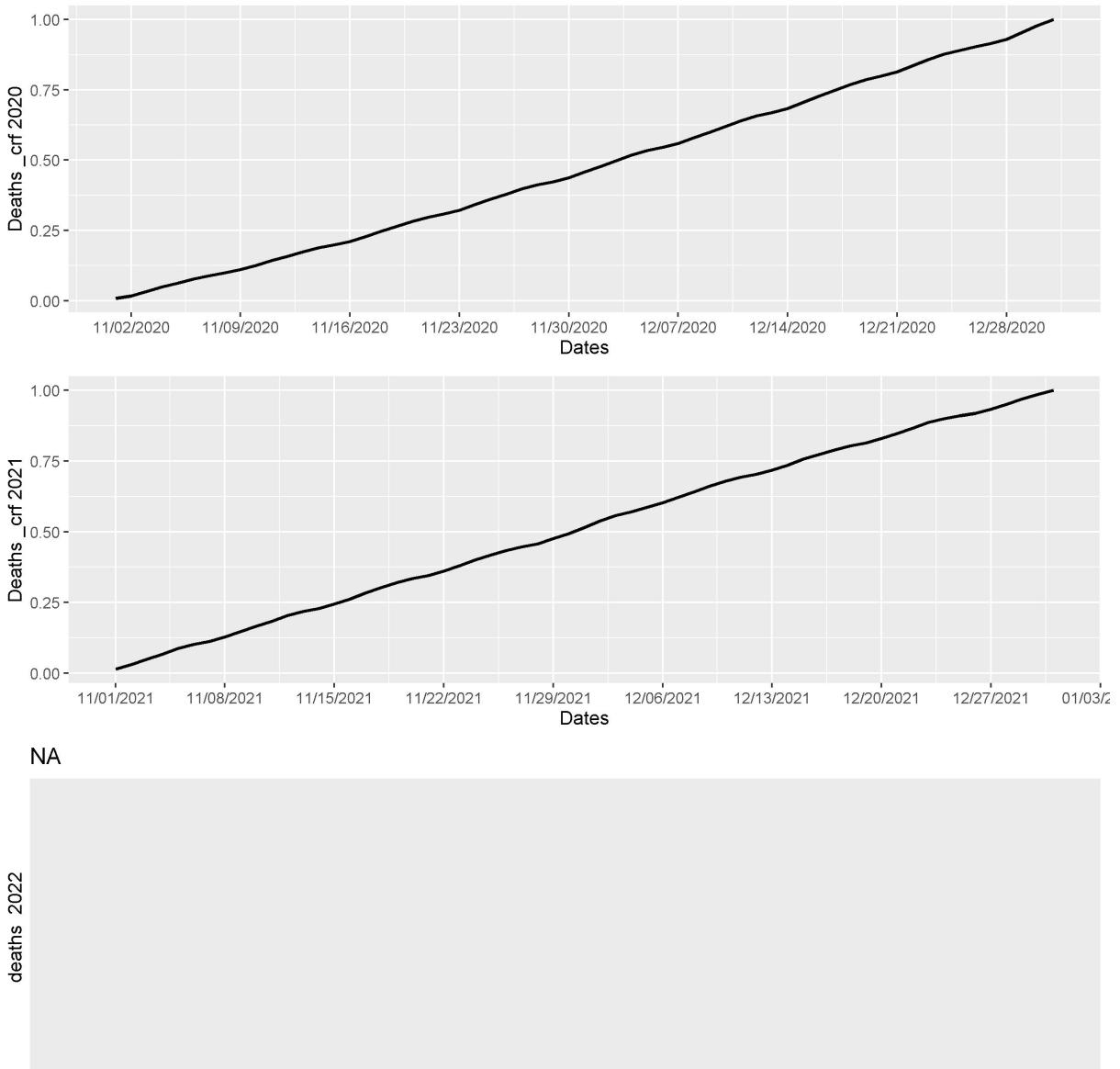


Hình 55: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tương đối tích lũy theo thời gian là 2 tháng cuối của năm của tất cả quốc gia

- 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

Source code

```
#vii6
country_chart ("World", "cum_rel", "11_12", "deaths", "vii6")
```



Hình 56: 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

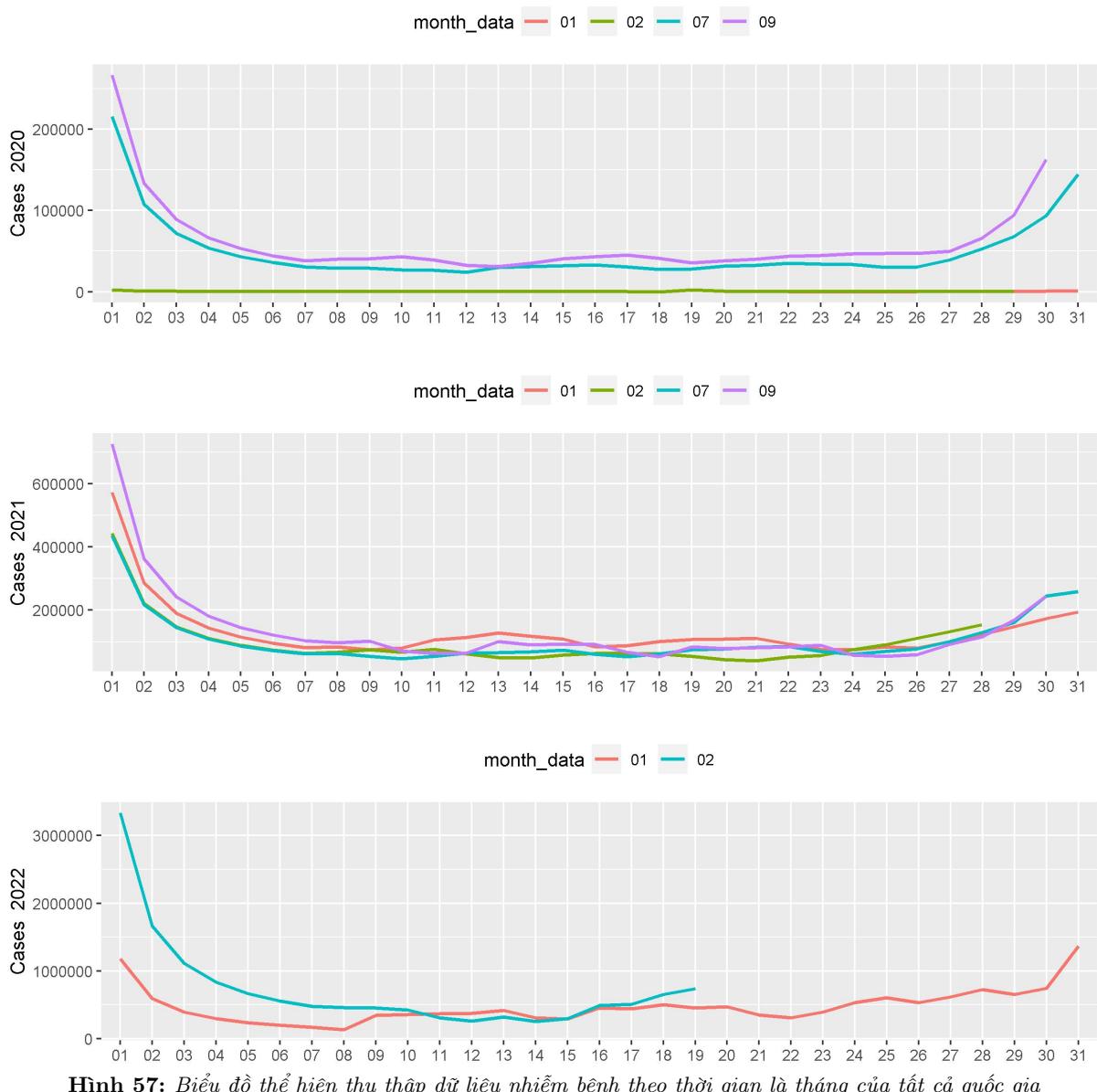
- viii) Nhóm câu hỏi liên quan đến tất cả quốc gia theo trung bình 7 ngày gần nhất

Trên từng năm hãy vẽ biểu đồ thể hiện trực Ox là thời gian, trực 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.

- 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

Source code

```
#viii1
country_chart("World", "line_chart", "2_1_7_9", "cases", "viii1", "avg")
```

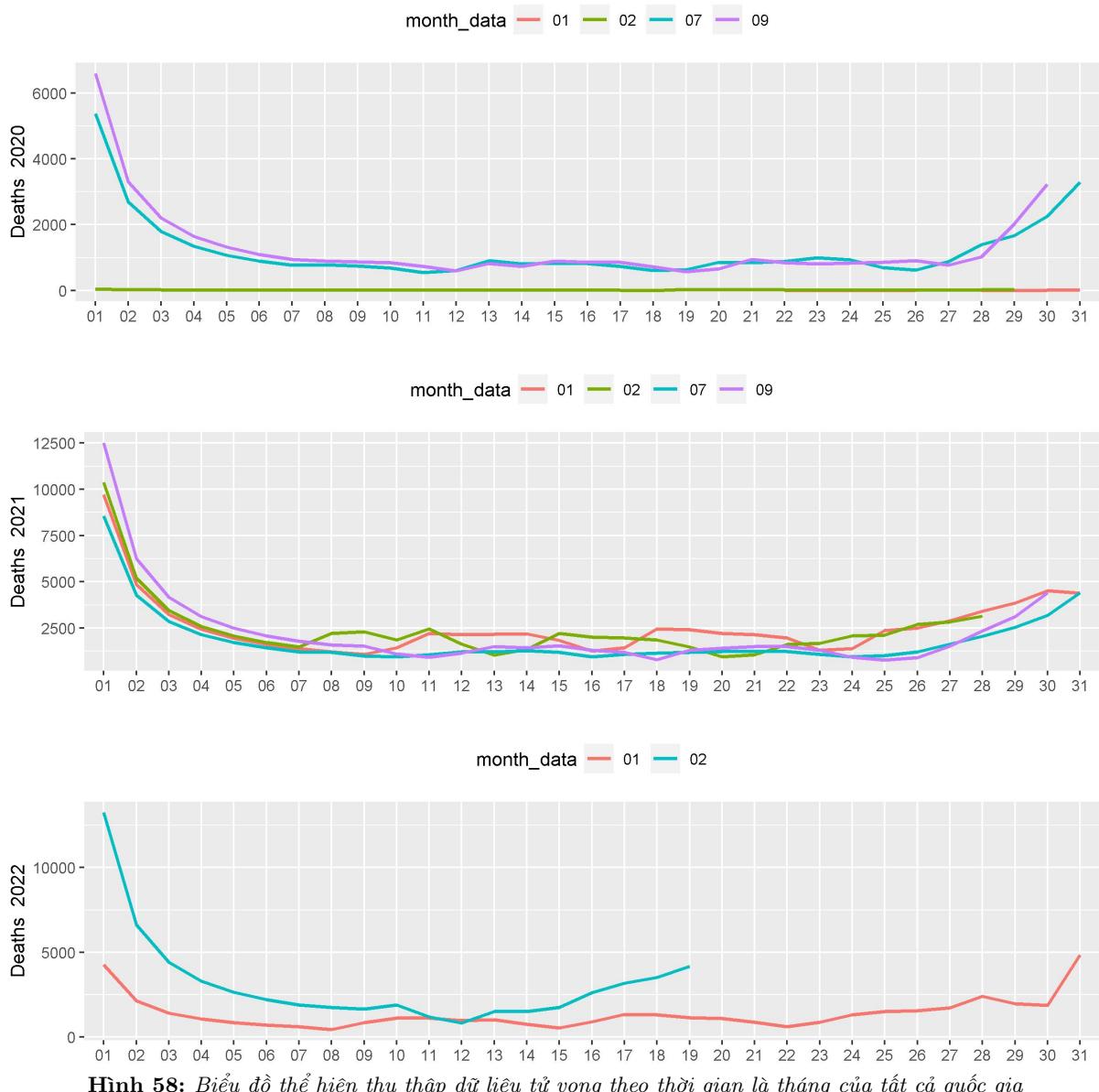


Hình 57: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo thời gian là tháng của tất cả quốc gia

- 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

Source code

```
#viii2
country_chart("World", "line_chart", "2_1_7_9", "deaths", "viii2", "avg")
```

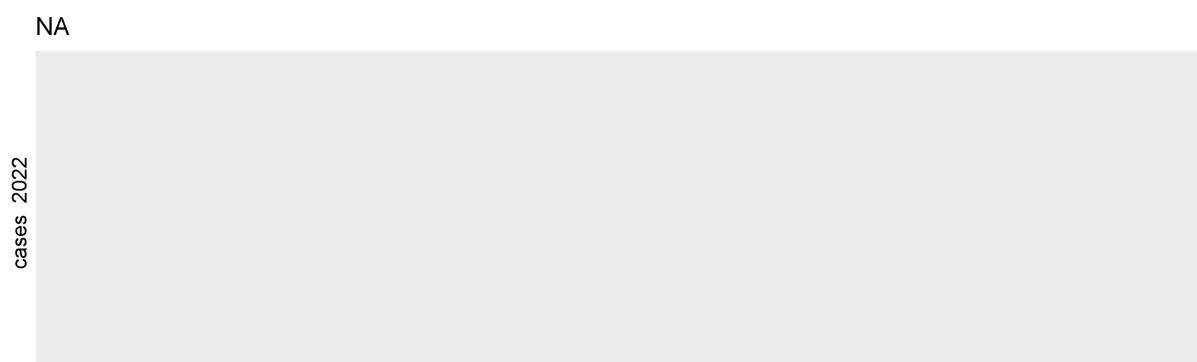
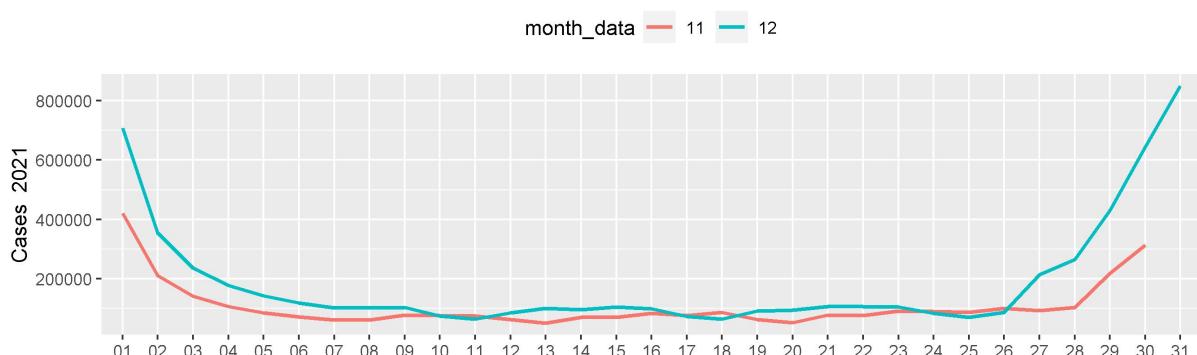
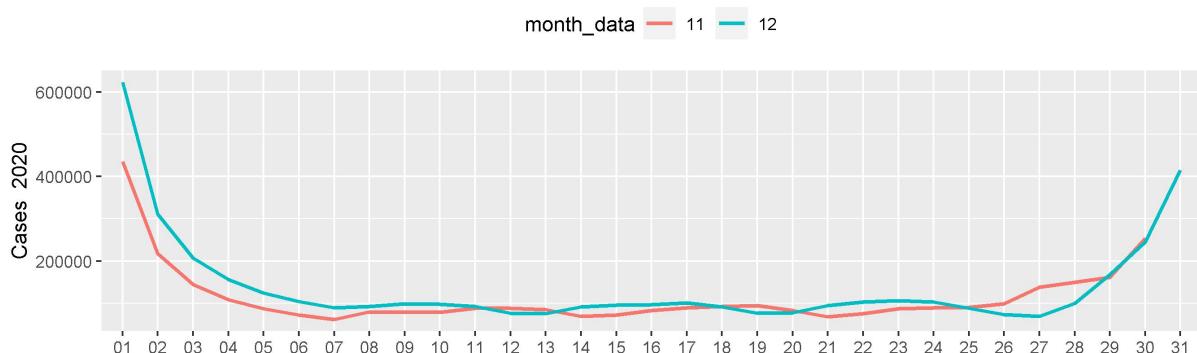


Hình 58: Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là tháng của tất cả quốc gia

- 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

Source code

```
#viii3
country_chart("World", "line_chart", "11_12", "cases", "viii3", "avg")
```

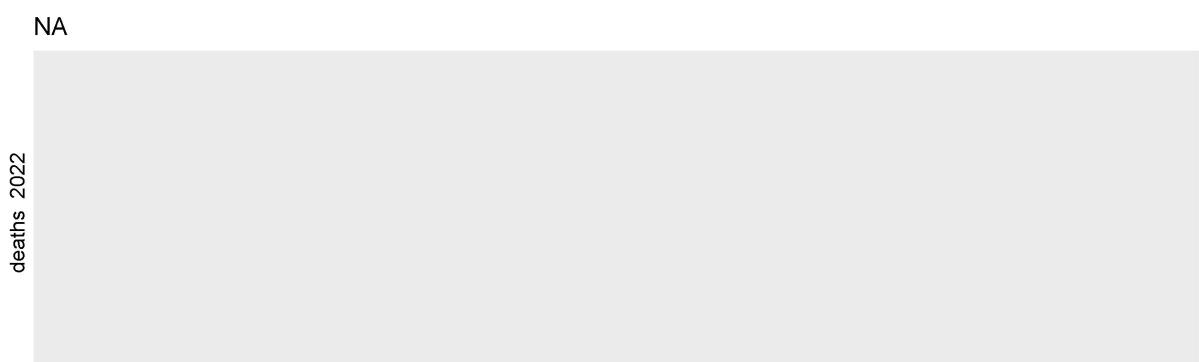
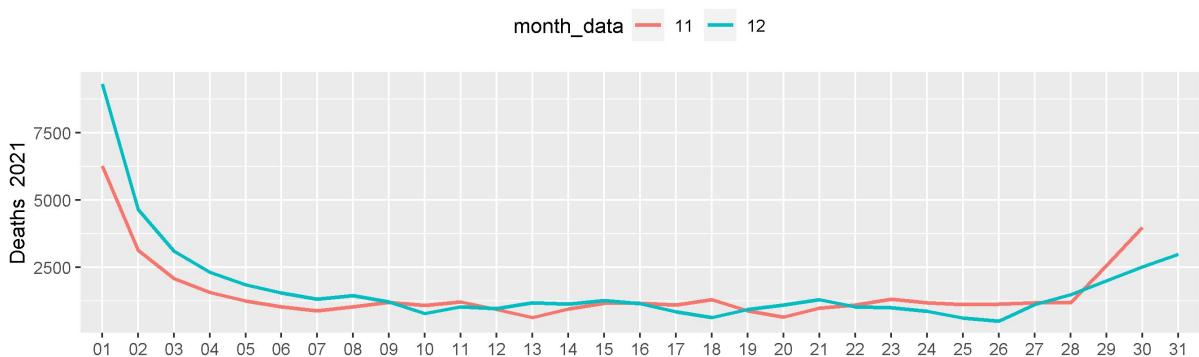
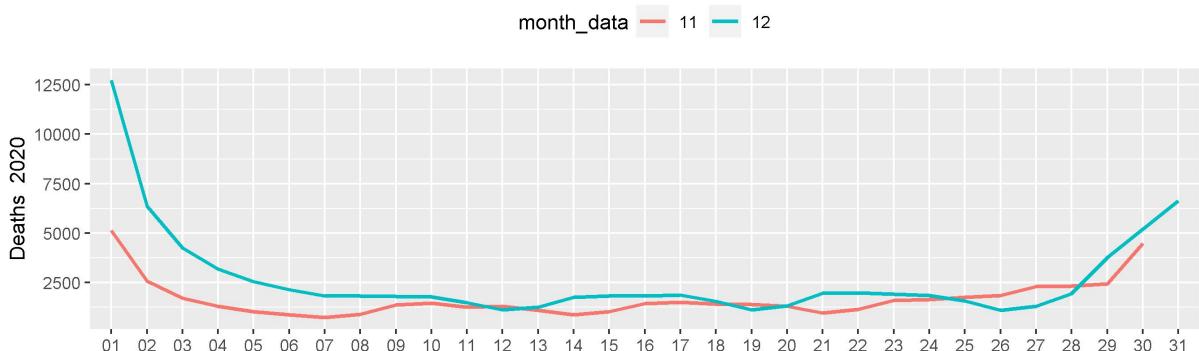


Hình 59: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh theo thời gian là 2 tháng của năm của tất cả quốc gia theo trung bình 7 ngày gần nhất

- 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

Source code

```
#viii4
country_chart("World", "line_chart", "11_12", "deaths", "viii4", "avg")
```

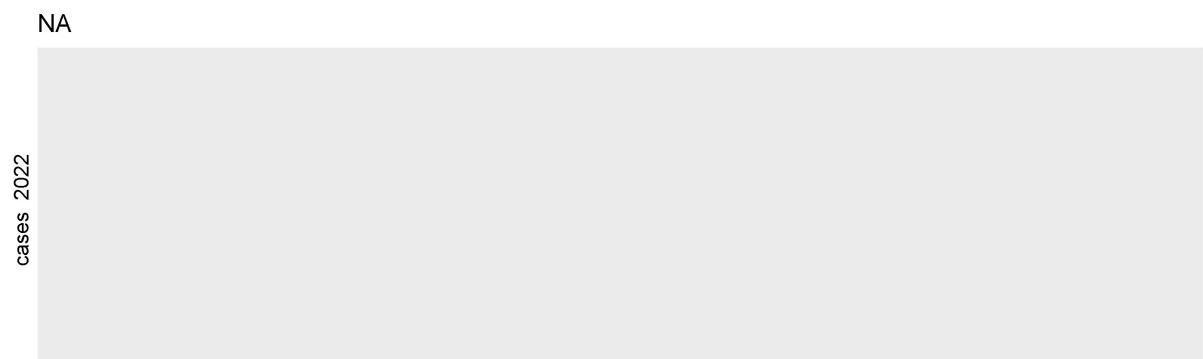
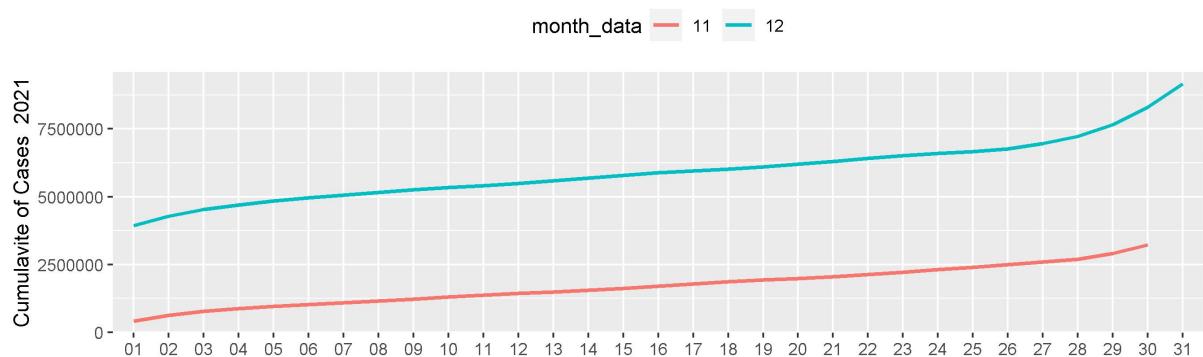
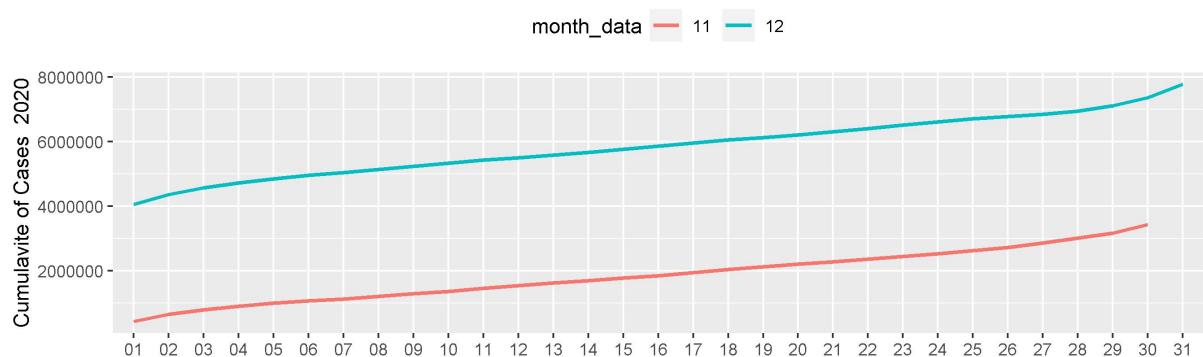


Hình 60: Biểu đồ thể hiện thu thập dữ liệu tử vong theo thời gian là 2 tháng của năm của tất cả quốc gia theo trung bình 7 ngày gần nhất

- 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

Source code

```
#viii5
country_chart("World", "cum", "11_12", "cases", "viii5", "avg")
```

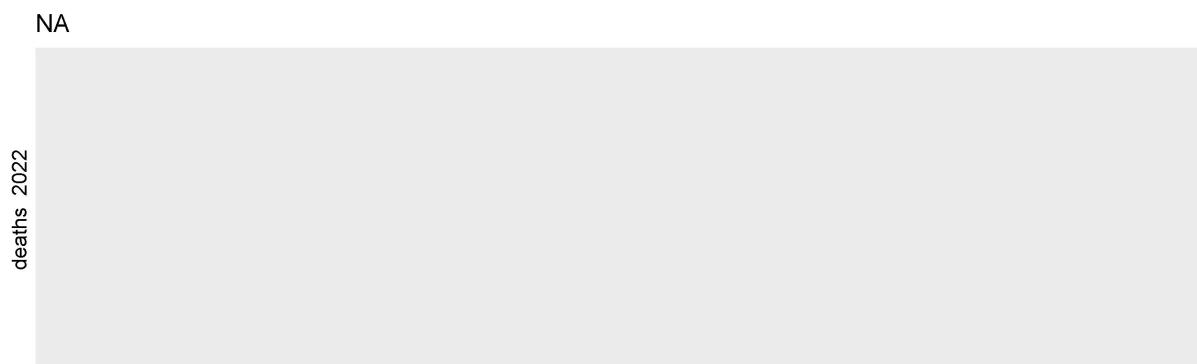
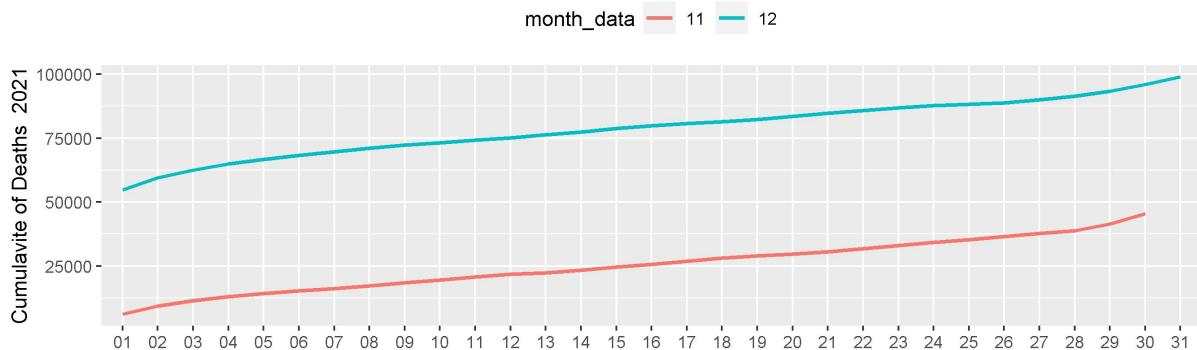
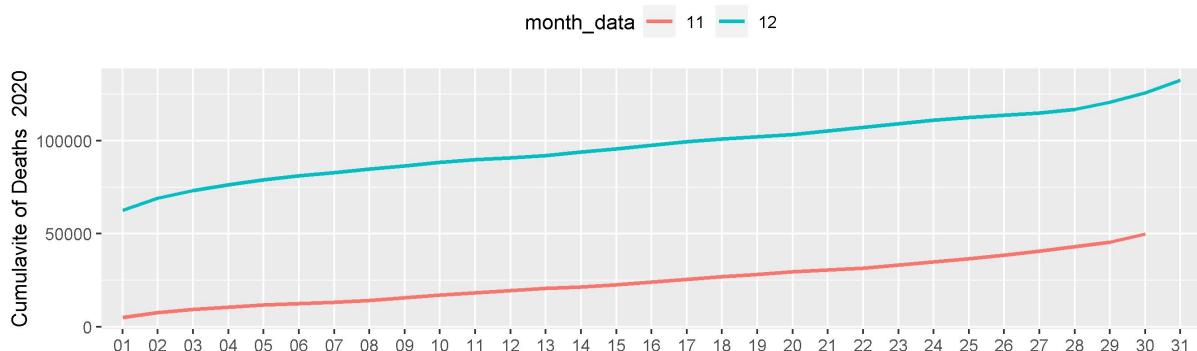


Hình 61: Biểu đồ thể hiện thu thập dữ liệu nhiễm bệnh tích lũy theo thời gian là 2 tháng của năm của tất cả quốc gia theo trung bình 7 ngày gần nhất

- 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

Source code

```
#viii6
country_chart("World", "cum", "11_12", "deaths", "viii6", "avg")
```



Hình 62: 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



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 đê thể hiện. Nếu ký số là 0 thì lấy tháng là 10.

Source code

```
#pre ix1
need_con <- c("Vietnam", "Japan", "Indonesia")
#need_month <- c("02", "01", "07", "09")
three_country <- data %>% filter(location %in% need_con)
three_country$date <- as.POSIXct(three_country$date,
                                 format = "%m/%d/%Y")

row.names(three_country) <- 1:nrow(three_country)

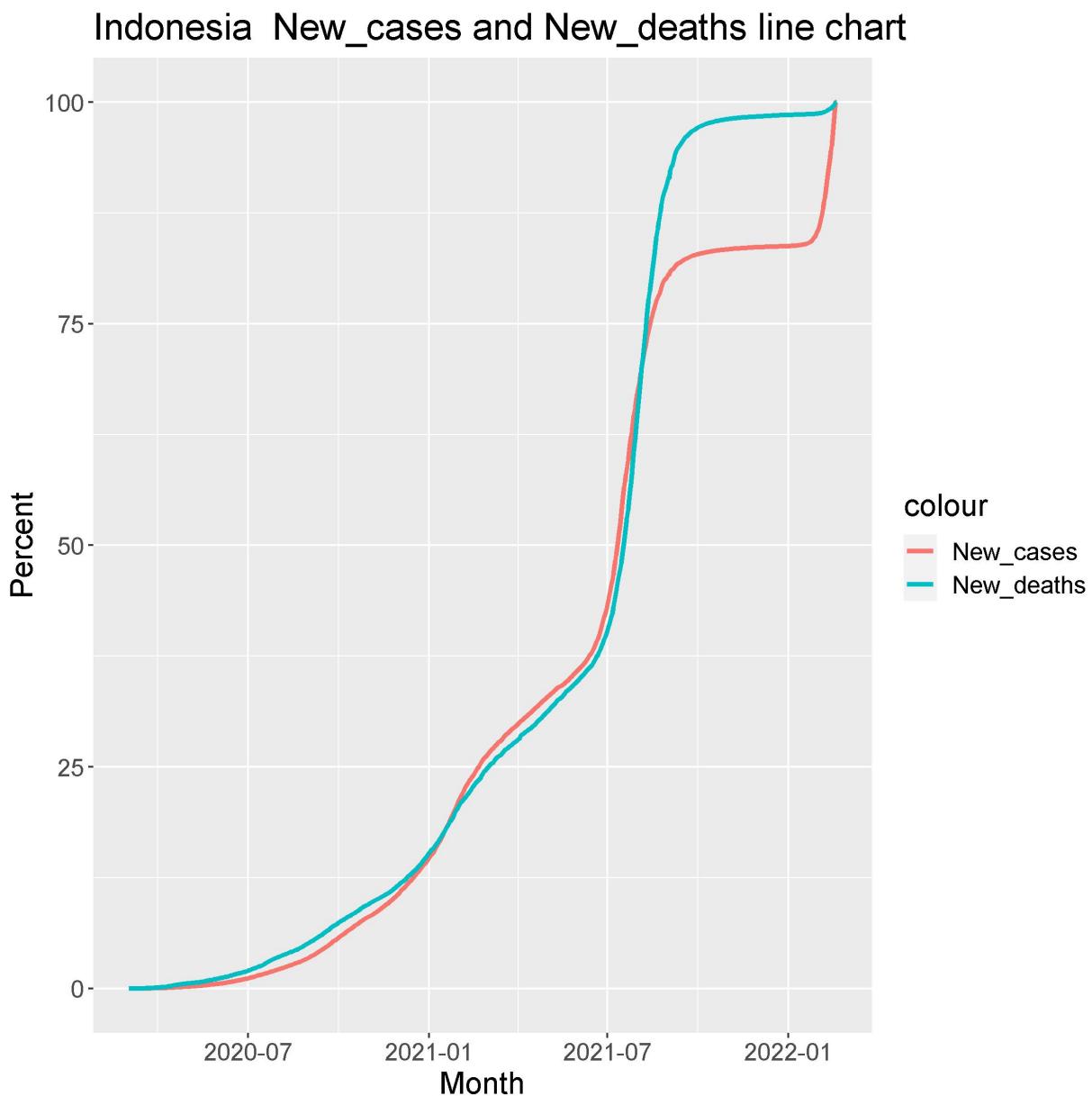
Vie<-three_country%>%filter(three_country$location == "Vietnam")
Jap<-three_country%>%filter(three_country$location == "Japan")
In<-three_country%>%filter(three_country$location == "Indonesia")
options("scipen"=10)
#View(Vie)
#function
draw_chart <- function(country_data, country_name)
{
  new_cases_data <- country_data$new_cases
  new_deaths_data <- country_data$new_deaths
  sum_cases <- sum(new_cases_data, na.rm = TRUE)
  sum_deaths <- sum(new_deaths_data, na.rm = TRUE)
  cases_cumu <- cbind(new_cases_data)
  deaths_cumu <- cbind(new_deaths_data)
  cases_cumu[is.na(cases_cumu)] <- 0
  deaths_cumu[is.na(deaths_cumu)] <- 0
  for(x in 1:(length(new_cases_data) - 1))
  {
    cases_cumu[x + 1] <- cases_cumu[x] + cases_cumu[x + 1]
    deaths_cumu[x + 1] <- deaths_cumu[x] + deaths_cumu[x + 1]
  }
  for(x in 1:length(new_cases_data))
  {
    cases_cumu[x] <- (cases_cumu[x] / sum_cases)*100
    deaths_cumu[x] <- (deaths_cumu[x] / sum_deaths)*100
  }
  name = country_data
  name = name$date
  chart_data <- data.frame(name, cases_cumu, deaths_cumu)
#View(chart_data)
  title = paste(country_name, "New_cases_and_New_deaths_line_chart")
  line_chart <- ggplot(data = chart_data, aes(x = name))+
    geom_line(aes(y=cases_cumu, colour="New_cases"), size=1.2)+
    geom_line(aes(y=deaths_cumu, colour="New_deaths"), size=1.2)+
    ylab("New_cases_and_New_deaths_percent")+
    xlab("Month")+
    ylab("Percent")+
    theme(text = element_text(size = 16))+
    ggtitle(title)
```



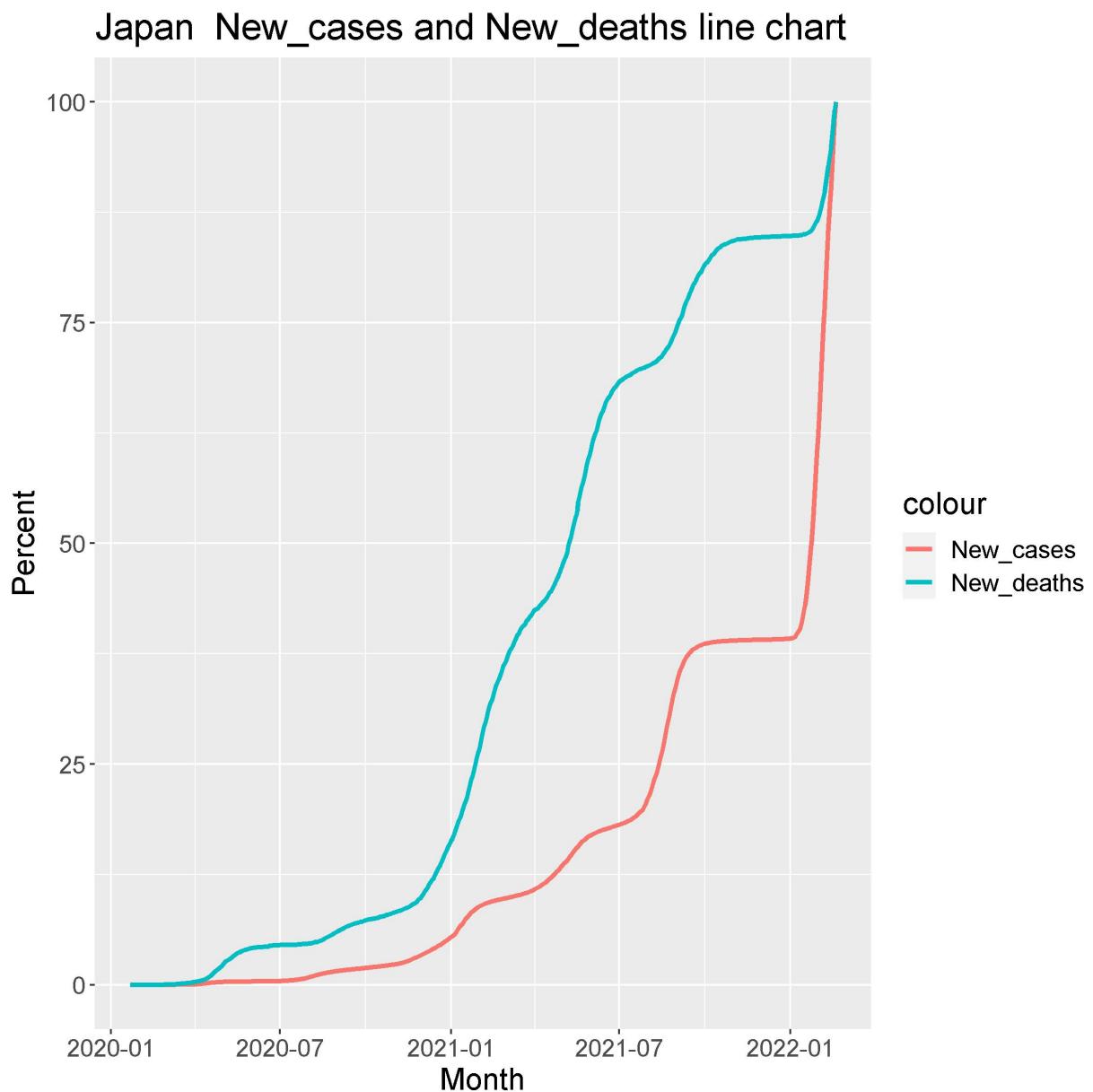
```
        return (line_chart)
    }

#ix1
ix1 <- function()
{
    #View(Vie)
    Vie_chart <- draw_chart(Vie[,3:6], "Vietnam")
    ggsave(filename = "ix1Vietnam.jpeg", plot = Vie_chart,
device = "jpeg", scale = 1, width = 8, height = 8)
    Jap_chart <- draw_chart(Jap[,3:6], "Japan")
    ggsave(filename = "ix1Japan.jpeg", plot = Jap_chart,
device = "jpeg", scale = 1, width = 8, height = 8)
    In_chart <- draw_chart(In[,3:6], "Indonesia")
    ggsave(filename = "ix1Indonesia.jpeg", plot = In_chart,
device = "jpeg", scale = 1, width = 8, height = 8)
}
ix1()
```

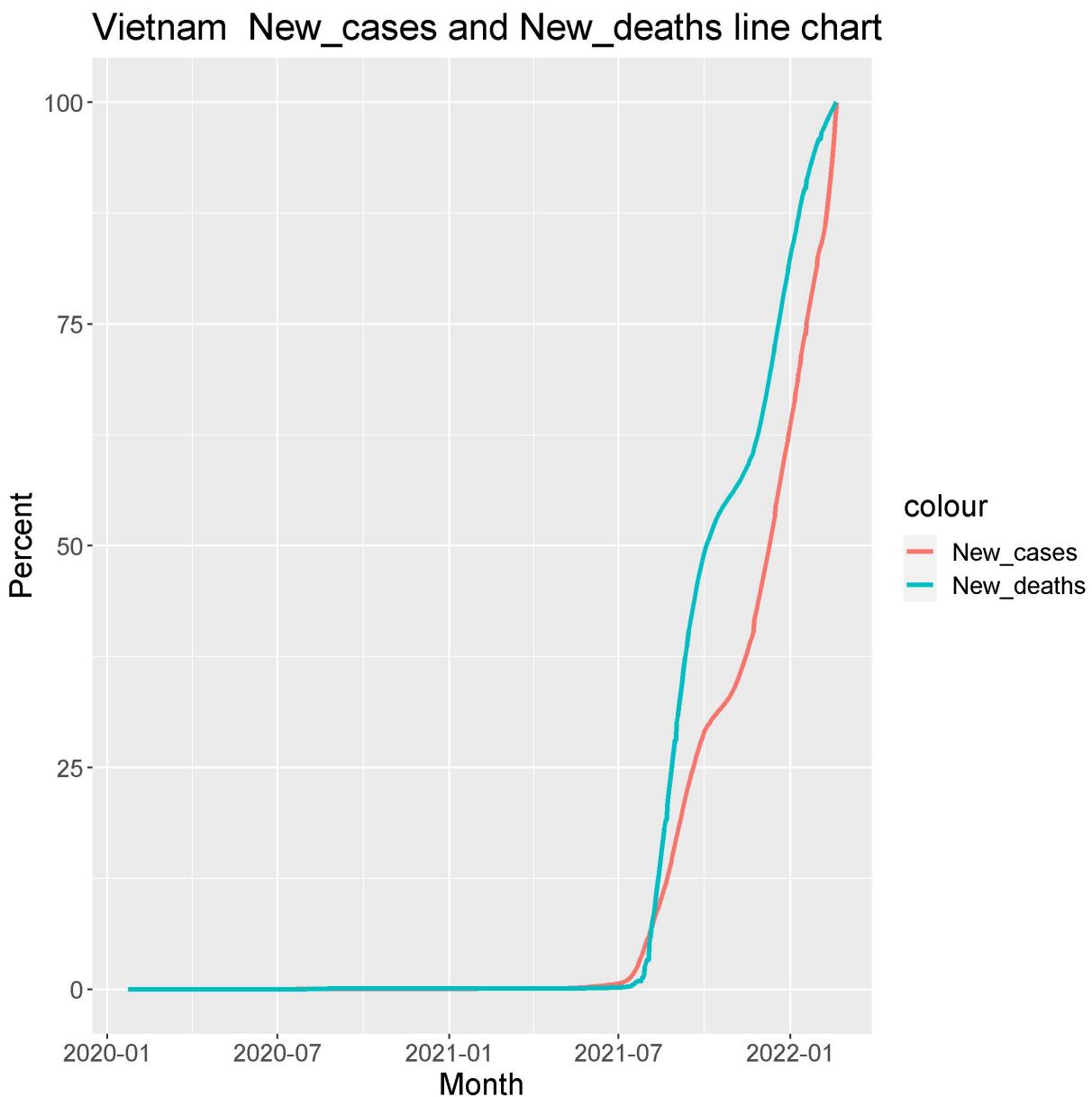
- Biến three_country nhận vào dữ liệu của 3 nước Indonesia, Nhật Bản, Việt Nam. Sử dụng `as.POSIXct` để định dạng lại ngày tháng năm và chuyển kiểu dữ liệu của cột date thành kiểu dữ liệu `Date`.
- Tách dữ liệu ra theo từng nước và gán vào các biến In(Indonesia), Jap(Nhật Bản), Vie(Việt Nam) để hỗ trợ cho việc vẽ biểu đồ theo từng nước.
- Hàm `draw_chart` sẽ trả về biểu đồ cần vẽ theo quốc gia, hàm chứa các tham số:
 - `country_data`: Tham số chứa dữ liệu của quốc gia cần vẽ biểu đồ.
 - `country_name`: Tham số chứa tên của quốc gia cần vẽ biểu đồ.
- Hàm `ggsave()` dùng để xuất biểu đồ theo định dạng ảnh.



Hình 63: 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 của Indonesia



Hình 64: 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 của Nhật Bản



Hình 65: 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 của Việt Nam



Function and Prep for ix 2-3

```
ix_cor_MY <- function(subdata, stt_month)
{
  data_my <- subdata %>% mutate(date = mdy(date),
    Month_Yr = format.ISO8601(date, precision = "ym"))
  data_my <- subset(data_my, !is.na(data_my[,5]))
  data_my <- subset(data_my, !is.na(data_my[,6]))
  subMonth <- subset(data_my, Month_Yr == stt_month)
  if (nrow(subMonth) == 0)
  {
    plot(NULL, xlim=c(0,1), ylim=c(0,1), main = stt_month,
      sub = paste("No_Correlation"), ylab="Deaths", xlab="Cases")
    abline(h = 0.5, col = "red", lwd = 3)
  }
  else
  {
    if (nrow(subMonth) == 1)
    {
      plot(subMonth[,5], subMonth[,6], pch = 10, col = "black",
        main = stt_month, sub = paste("No_Correlation"),
        ylab="Deaths", xlab="Cases")
      abline(h = subMonth[1,6], col = "red", lwd = 3)
    }
    else
    {
      num <- cor(subMonth[,5], subMonth[,6], method = "pearson")
      if (is.na(num)) plot(subMonth[,5], subMonth[,6], pch = 10,
        col = "black", main = stt_month,
        sub = paste("No_Correlation"),
        ylab="Deaths", xlab="Cases")
      else plot(subMonth[,5], subMonth[,6], pch = 10, col = "black",
        main = stt_month,
        sub = paste("Correlation:", round(num,2)),
        ylab="Deaths", xlab="Cases")
      abline(lm(subMonth[,6] ~ subMonth[,5]), col = "red", lwd = 3)
    }
  }
}

id_data <- subset(data, location == "Indonesia")
jp_data <- subset(data, location == "Japan")
vn_data <- subset(data, location == "Vietnam")
my_months <- c("2020-01", "2020-02", "2020-07", "2020-09", "2021-01",
  "2021-02", "2021-07", "2021-09", "2022-01", "2022-02")
```

- Hàm `ix_cor_MY` có tác dụng nhận vào data frame của một đất nước với biến truyền là `subdata` và tên của một tháng được viết theo định dạng "`YYYY-MM`" vào biến `stt_month` và vẽ ra biểu đồ tương quan dựa trên dữ liệu trong tháng đó của quốc gia mong muốn.
- Hàm này sẽ xét cột `date` của `subdata` vừa nhập vào và tiến hành tạo ra một cột mới chỉ chứa tháng và năm dựa trên ngày tương ứng của cột `date` và chuyển data frame mới này vào một biến mới là `data_my`. Sau đó `data_my` này sẽ loại tất cả những hàng không phù hợp với 3 tiêu chí sau: dữ liệu cột `new_case` không phải `NA`, dữ liệu cột `new_death` không phải `NA`, tháng ở cột mới vừa tạo phải khớp với tên của tháng trong biến `stt_month`. Sau đó là tiếp tục đưa data frame mới này vào biến `subMonth`. Với dữ liệu đã được xử lý xong, ta tiến hành vẽ biểu đồ.
- Nếu data frame của biến `subMonth` có số hàng là 0 thì nghĩa là trong tháng đó không có dữ



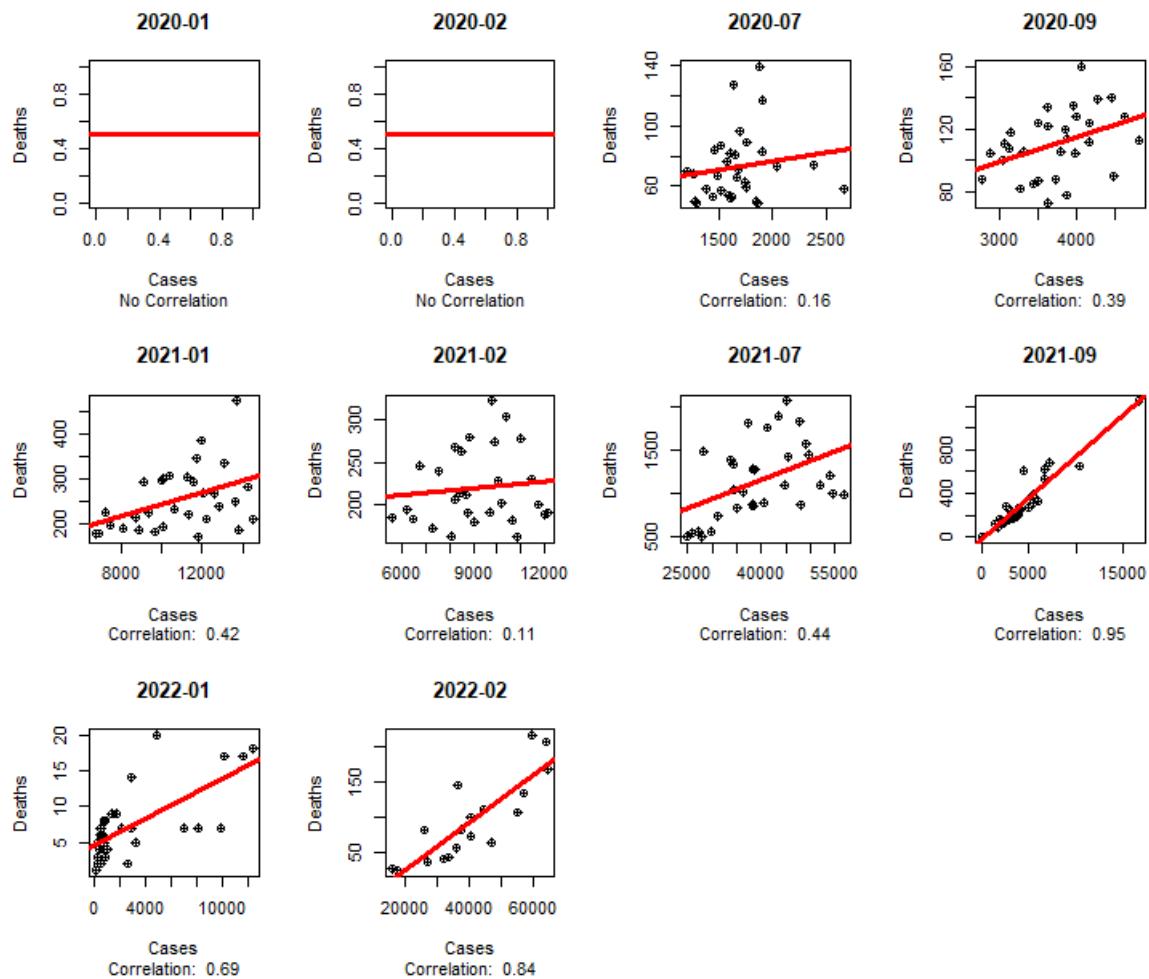
liệu để có thể xét tương quan được. Vậy ta sẽ vẽ một biểu đồ rỗng với một đường nằm ngang và in ra dòng chữ "*NoCorrelation*" dưới biểu đồ.

- Nếu data frame của biến *subMonth* có số hàng là 1 thì nghĩa là trong tháng đó chỉ có duy nhất 1 dữ liệu và nhiều đó là không đủ để có thể xét tương quan được. Vậy ta cũng sẽ vẽ một biểu đồ với chỉ duy nhất một điểm tượng trưng cho dữ liệu duy nhất đó và một đường nằm ngang tại vị trí của điểm đó và in ra dòng chữ "*NoCorrelation*" dưới biểu đồ.
- Nếu 2 điều trên không xảy ra nghĩa là dữ liệu của ta có thể tính được tương quan. Ta tính hệ số tương quan này bằng cách dùng hàm *cor(ctnew_case, ct_newdeath, method = "pearson")* để tính hệ số tương quan pearson cho 2 cột nhiễm và tử vong của dữ liệu. Ta đưa hệ số tương quan này vào biến *num*.
- Nếu biến *num* có giá trị là *NA*, nghĩa là một hoặc hai cột trong dữ liệu mang các giá trị giống hệt nhau và điều này không thể cho ta thấy được sự tương quan giữa 2 dữ liệu nhiễm và tử vong. Ta tiến hành vẽ một biểu đồ với các điểm dựa trên dữ liệu nhiễm và tử vong theo từng cặp sau đó dùng thêm hàm *lm()* để vẽ một đường tuyến tính dựa trên xu hướng của các điểm trên biểu đồ. Vì các giá trị không có tương quan nên đường này sẽ là một đường ngang. Ở dưới biểu đồ sẽ in ra dòng chữ "*NoCorrelation*".
- Nếu biến *num* không phải *NA* thì nghĩa là dữ liệu của ta có sự tương quan. Ta cũng tiến hành vẽ biểu đồ như khi *num* là *NA* tuy nhiên ở dưới biểu đồ sẽ là hệ số tương quan của dữ liệu mà ta vừa tính được làm tròn lên 2 chữ số thập phân.
- Đoạn code ở dưới hàm là các câu lệnh để lấy ra dữ liệu của 3 nước thuộc về nhóm cần tính cũng như tạo ra 1 vector chứa tên các tháng cần tính tương quan. Với điều này, ở mỗi quốc gia ta sẽ chỉ cần chạy 1 vòng lặp *for* cho từng giá trị tháng trong vector chứa tháng và xài hàm để vẽ biểu đồ tương quan cho dữ liệu của quốc gia đó với tất cả các tháng chứa trong vector. Đồng thời ta sẽ xài câu lệnh *par(mfrow = c(3, 4))* điều chỉnh layout của bảng plot để hiện được cùng lúc nhiều biểu đồ.

2) Xét tương quan trong mỗi tháng

Source code for Indonesia

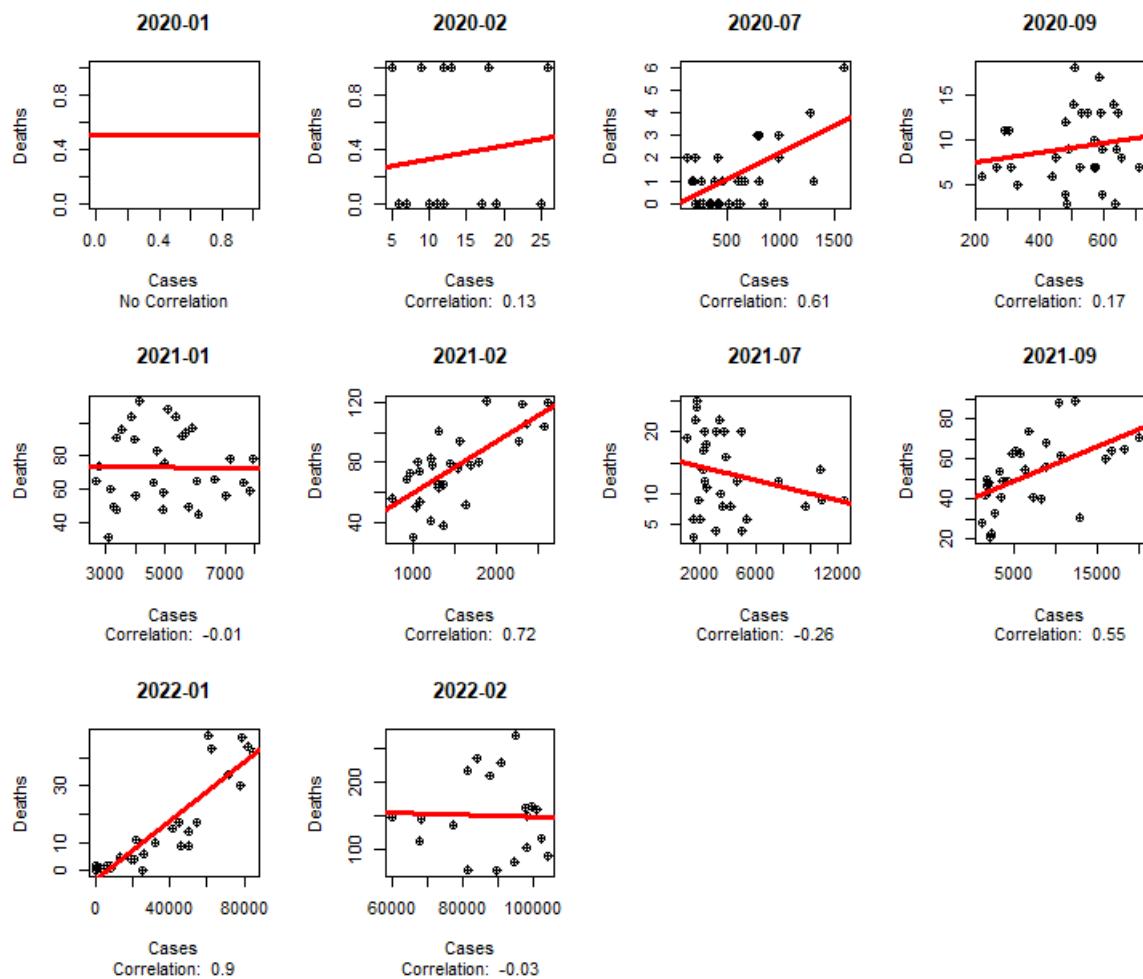
```
par(mfrow=c(3,4))
for(i in 1:length(my_months))
{
  ix_cor_MY(id_data,my_months[i])
}
```



Hình 66: Biểu đồ thể hiện tương quan trên từng tháng của Indonesia

Source code for Japan

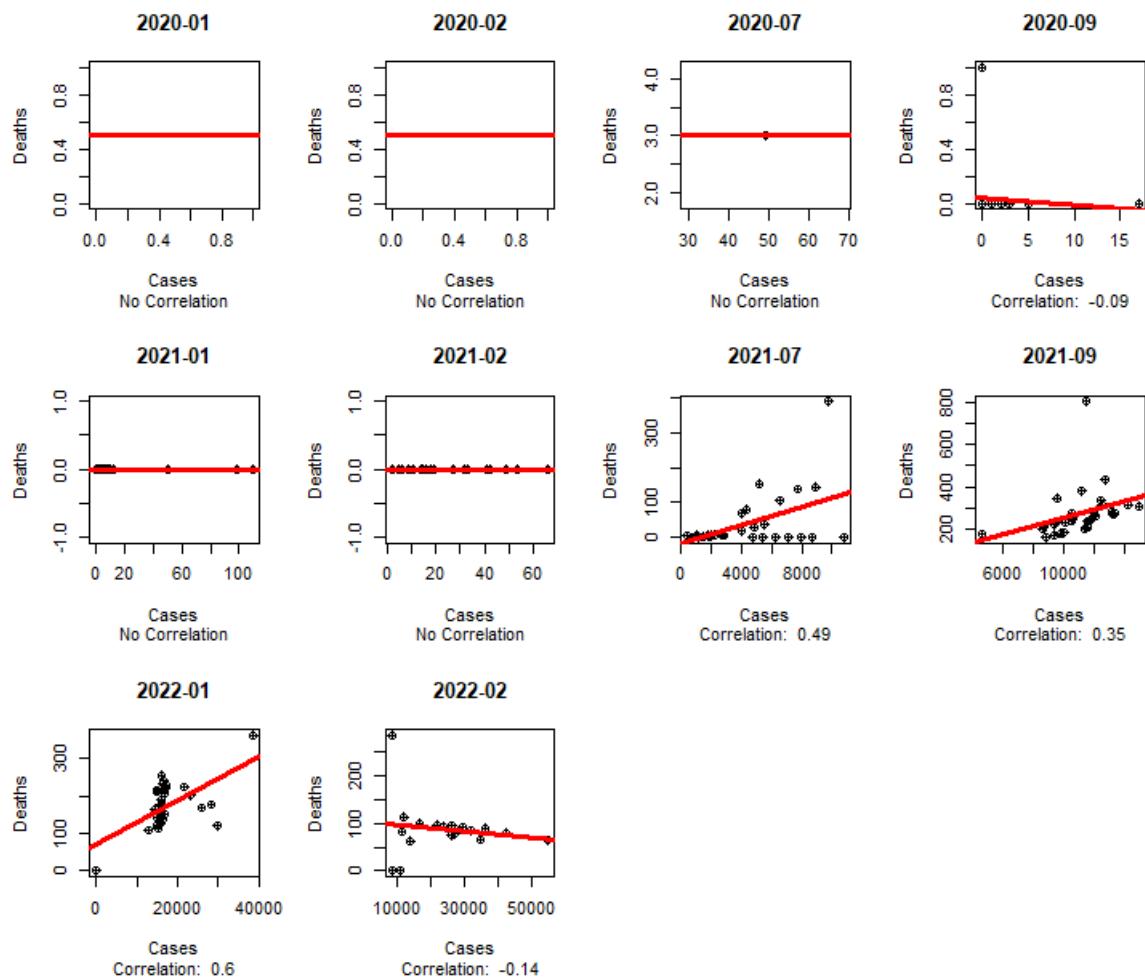
```
par(mfrow=c(3 ,4))
for (i in 1:length(my_months))
{
  ix_cor_MY(jp_data,my_months[ i ])
}
```



Hình 67: Biểu đồ thể hiện tương quan trên từng tháng của Nhật Bản

Source code for Vietnam

```
par(mfrow=c(3 ,4))
for (i in 1:length(my_months))
{
  ix_cor_MY(vn_data ,my_months [ i ])
}
```



Hình 68: Biểu đồ thể hiện tương quan trên từng tháng của Việt Nam



3) Xét tương quan trong mỗi tháng theo trung bình 7 ngày gần nhất

Avg 7 days function and Prep for ix3

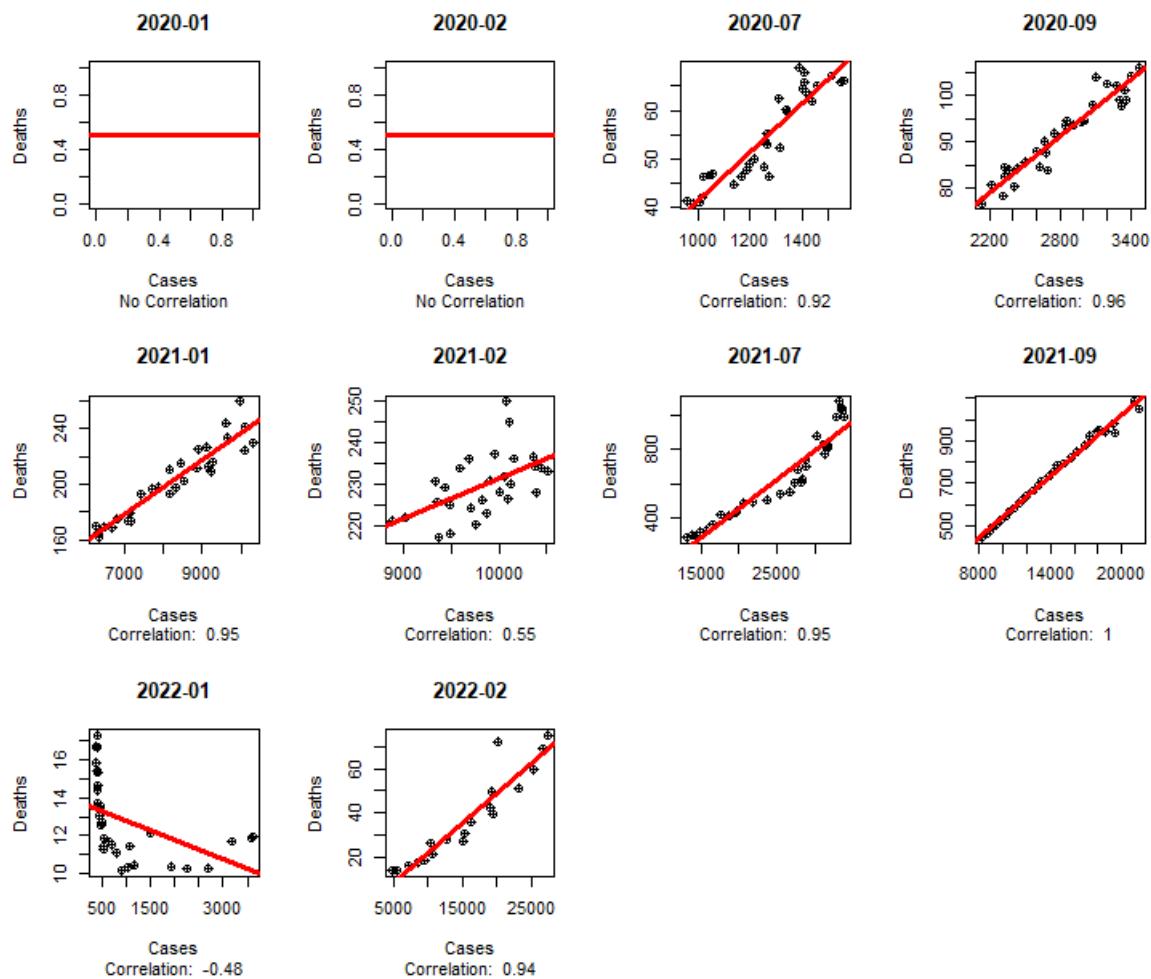
```
avg7 <- function (subdata , col)
{
  new_avg7 <- subdata [ , col ]
  j <- 1
  for ( i  in  1 : length ( new_avg7 ))
  {
    new_avg7 [ i ] <- ( sum ( new_avg7 [ j : i ] , na . rm = TRUE ) / ( i - j + 1 ))
    if ( i >= 7 ) j <- j + 1
  }
  return ( new_avg7 )
}

id_data [ , 5 ] <- avg7 ( id_data , 5 )
id_data [ , 6 ] <- avg7 ( id_data , 6 )
jp_data [ , 5 ] <- avg7 ( jp_data , 5 )
jp_data [ , 6 ] <- avg7 ( jp_data , 6 )
vn_data [ , 5 ] <- avg7 ( vn_data , 5 )
vn_data [ , 6 ] <- avg7 ( vn_data , 6 )
```

- Hàm *avg7* có tác dụng nhận vào dữ liệu của một quốc gia và cột cần tính và trả về một cột mới đã được điều chỉnh giá trị theo công thức tính trung bình 7 ngày dựa trên dữ liệu tại cột cần tính của quốc gia nhập vào.
- Ta sẽ dùng hàm này để biến đổi các cột nhiễm và tử vong của 3 quốc gia thuộc về nhóm cần tính bằng giá trị đã được chỉnh sửa theo công thức tính trung bình 7 ngày.
- Sau khi đã có dữ liệu mới, ta chỉ việc áp dụng cách làm cũ của câu *ix - 2* để vẽ các biểu đồ tương quan mới dựa trên dữ liệu trung bình 7 ngày.

Source code for Indonesia

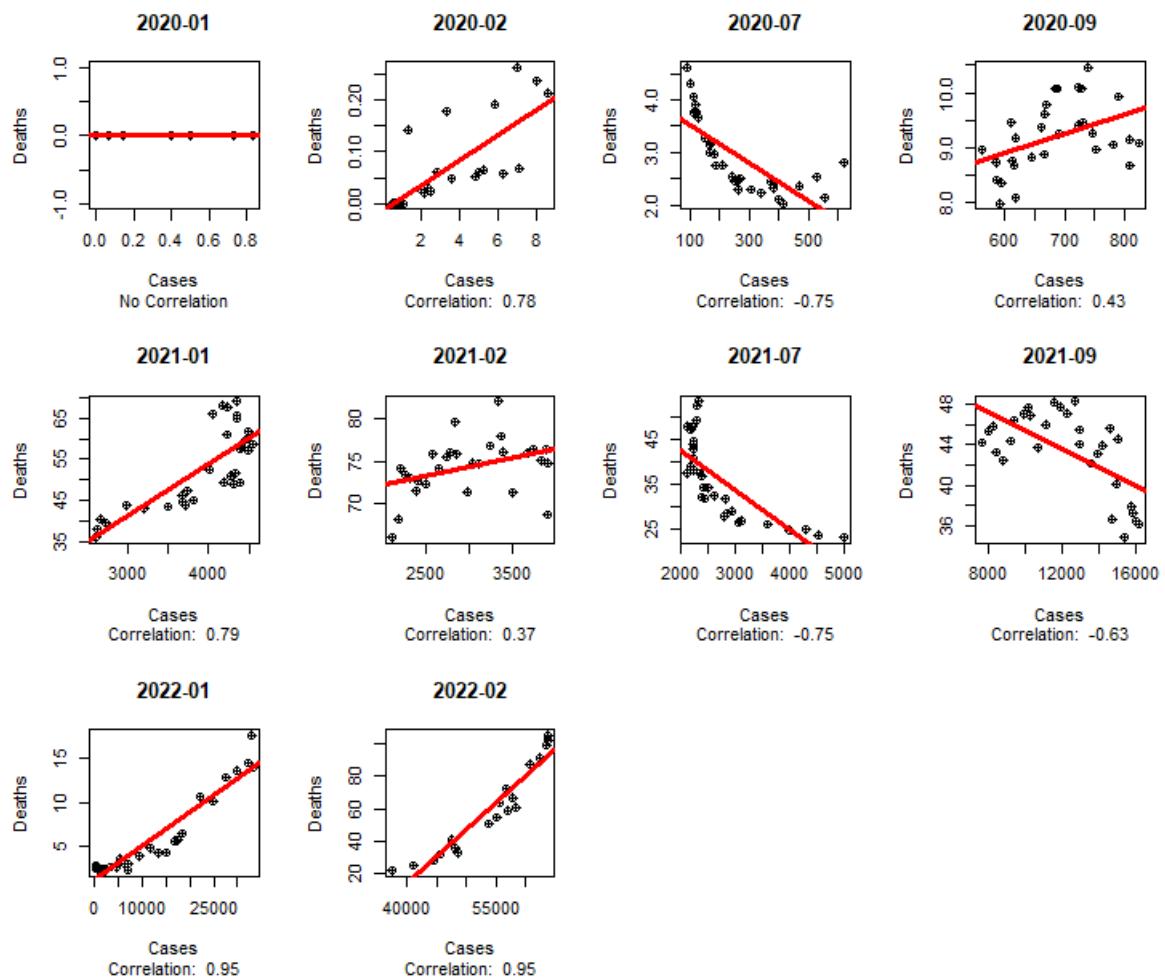
```
par(mfrow=c(3 ,4))
for (i in 1:length(my_months))
{
  ix_cor_MY(id_data,my_months[i])
}
```



Hình 69: Biểu đồ thể hiện tương quan trên từng tháng của Indonesia theo trung bình 7 ngày gần nhất

Source code for Japan

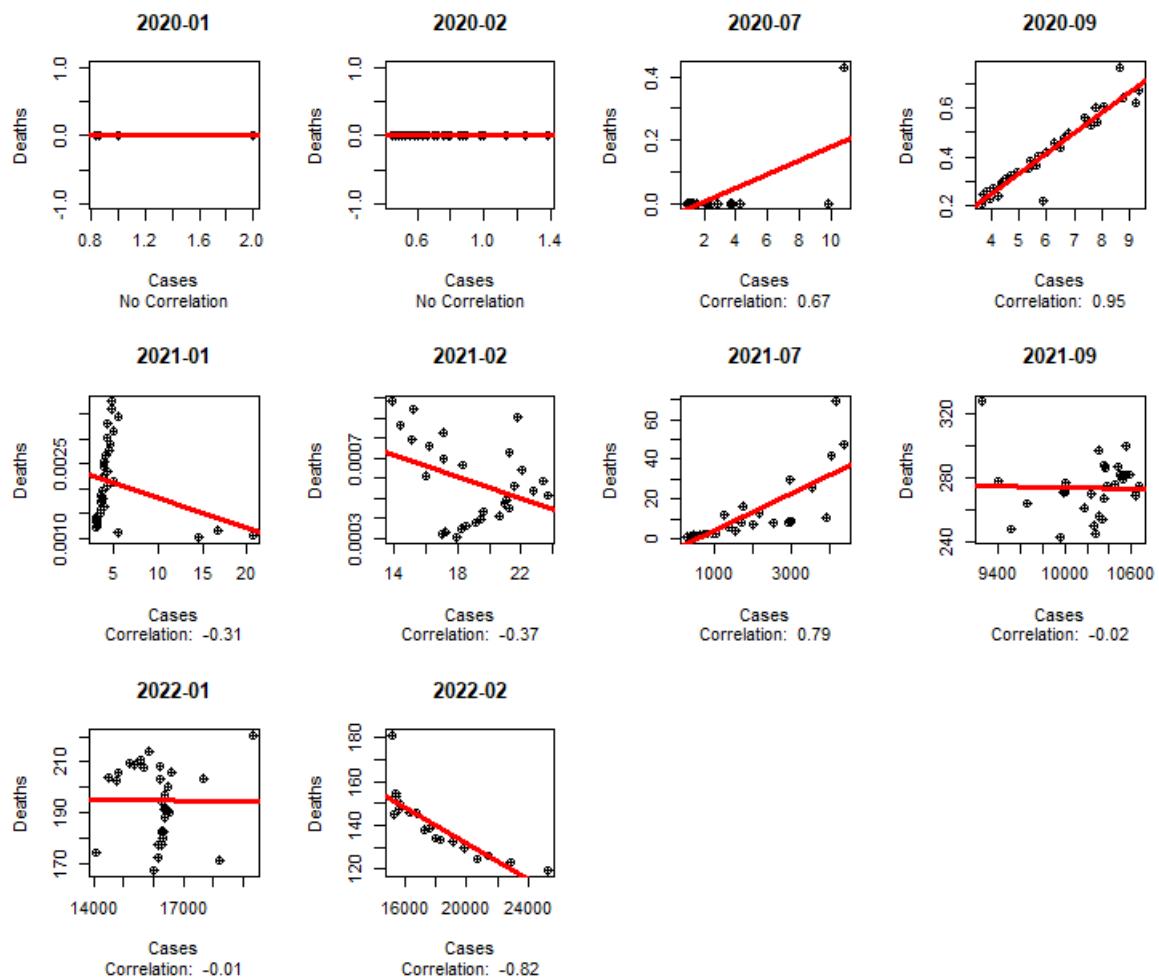
```
par(mfrow=c(3,4))
for (i in 1:length(my_months))
{
  ix_cor_MY(jp_data,my_months[i])
}
```



Hình 70: Biểu đồ thể hiện tương quan trên từng tháng của Nhật Bản theo trung bình 7 ngày gần nhất

Source code for Vietnam

```
par(mfrow=c(3,4))
for (i in 1:length(my_months))
{
  ix_cor_MY(vn_data,my_months[i])
}
```



Hình 71: Biểu đồ thể hiện tương quan trên từng tháng của Việt Nam theo trung bình 7 ngày gần nhất



x) Nhóm câu hỏi riêng

- 1) So sánh tình trạng nhiễm bệnh của các quốc gia trong 7 ngày cuối của năm cuối cùng
 - Tại Indonesia số ca mắc mới có xu hướng tăng mạnh từ ngày 13/2/2022 đến 16/2/2022 chỉ có 14/2/2022 số ca mắc giảm và đạt giá trị thấp nhất trong 7 ngày cuối của năm cuối cùng và số ca mắc đạt đỉnh vào ngày 16/2/2022 với 64718 ca mắc và sau đó có xu hướng giảm .
 - Tại Nhật Bản số ca mắc mới có xu hướng tăng mạnh từ ngày 13/2/2022 đến 17/2/2022 chỉ có 14/2/2022 số ca mắc giảm và đạt giá trị thấp nhất trong 7 ngày cuối của năm cuối cùng và số ca mắc đạt đỉnh vào ngày 17/2/2022 với 95115 ca mắc và sau đó có xu hướng giảm .
 - Tại Việt Nam số ca mắc mới có xu hướng tăng nhẹ từ ngày 13/2/2022 đến 18/2/2022 và tăng đột biến vào ngày 19/2/2022 và đạt đỉnh với 54830 ca mắc.

=> Tình trạng nhiễm bệnh trong 7 ngày cuối của năm cuối cùng của Indonesia và Nhật Bản khá giống nhau do có xu hướng tăng mạnh ở những ngày đầu tiên và giảm đi sau đó chỉ có duy nhất Việt Nam là có xu hướng tăng nhẹ liên tục và tăng đột biến vào ngày cuối cùng.
- 2) So sánh tình trạng tử vong của các quốc gia trong 7 ngày cuối của năm cuối cùng
 - Tại Indonesia số ca tử vong có xu hướng tăng mạnh từ ngày 13/2/2022 đến 18/2/2022 chỉ có 15/2/2022 số ca tử vong giảm và số ca tử vong đạt đỉnh vào ngày 18/2/2022 với 216 ca và sau đó có xu hướng giảm .
 - Tại Nhật Bản số ca tử vong có xu hướng tăng nhẹ từ ngày 13/2/2022 đến 14/2/2022 và tăng đột biến từ ngày 15/2/2022 đến 17/2/2022, đạt đỉnh vào ngày 17/2/2022 với 271 ca tử vong, 2 ngày cuối cùng giảm so với ngày 17/2/2022 nhưng vẫn có xu hướng tăng.
 - Tại Việt Nam số ca tử vong có xu hướng biến động không đồng đều từ ngày 13/2/2022 đến ngày 17/2/2022, đạt đỉnh vào ngày 17/2/2022 với 90 ca tử vong và sau đó có xu hướng giảm.

=> Tình trạng tử vong trong 7 ngày cuối của năm cuối cùng của Indonesia, Nhật Bản và Việt Nam đều khác nhau tuy nhiên tại Indonesia và Nhật Bản nhìn chung đều có xu hướng tăng đột biến và giảm sau đó, số ca tử vong tại Việt Nam biến động những ngày đầu và giảm những ngày cuối.
- 3) Cho biết các khoảng thời gian nào mà tỉ lệ tử vong tích lũy giảm mạnh nhưng tỉ lệ nhiễm bệnh tích lũy tăng mạnh hoặc ngược lại cho các quốc gia.
- 4) Với k là mốc bùng phát dịch, hãy xác định k và cho biết các khoảng thời gian bùng phát
- 5) Với k là mốc bùng tử vong, hãy xác định k và cho biết các khoảng thời gian bùng phát
- 6) Khoảng thời gian bùng phát nhiễm bệnh lớn nhất giữa các quốc gia có chồng lên nhau không, Cho biết khoảng thời gian giao nhau đó?
- 7) Khoảng thời gian bùng phát tử vong lớn nhất giữa các quốc gia có chồng lên nhau không, Cho biết khoảng thời gian giao nhau đó?

Source code for x7

```
coun_name <- unique(data[,3])
counMax <- subset(data, location == coun_name[1])
maxDeath <- data.frame(subset(counMax,
                                new_deaths == max(counMax$new_deaths, na.rm = TRUE)))
for (i in 2:length(coun_name))
{
  counMax <- subset(data, location == coun_name[i])
  maxDeath <- rbind(maxDeath, subset(counMax,
                                       new_deaths == max(counMax$new_deaths, na.rm = TRUE)))
}
maxDeath <- subset(maxDeath, duplicated(maxDeath[,4]) |
                     duplicated(maxDeath[,4], fromLast=TRUE))
maxDeath <- maxDeath[order(as.Date(maxDeath$date, format="%d/%m/%Y"))]
View(maxDeath)
unique(maxDeath[,4])
```



- Tiến hành lấy tên của các quốc gia thông qua hàm `unique()` và bỏ vào biến `coun_name`. Ta tiếp tục lấy dòng có số lượng tử vong cao nhất của từng quốc gia, bắt đầu từ quốc gia đầu tiên để lấy gốc tạo 1 data frame mới có tên biến là `maxDeath` và từ đó chạy 1 vòng lặp for để gắn các dòng có lượng tử vong cao nhất của các quốc gia còn lại vào thông qua hàm `subset()` và `max()`.
- Ta tiếp tục lọc dữ liệu của `maxDeath` thông qua câu điều kiện `duplicated(maxDeath[4])|duplicated(maxDeath[4], fromLast = TRUE)` để loại ra các quốc gia có thời gian bùng phát tử vong lớn nhất không trùng với bất kỳ quốc gia nào trong dữ liệu. Sau đó ta sẽ sắp xếp lại dữ liệu theo ngày để có thể tiện quan sát thông qua `View()`.
- Ta có thể lấy trực tiếp ra những ngày bên trong này bằng cách xài câu lệnh `unique(maxDeath[4])`. Nó sẽ liệt kê danh sách những ngày mà khi đó ít nhất 2 quốc gia có đợt bùng phát tử vong lớn nhất.

```
> unique(maxDeath[4])
[1] "8/1/2021"  "9/1/2021"  "10/1/2021" "8/5/2021" "12/6/2021"
     "1/8/2021"  "4/8/2021"  "11/8/2021" "8/10/2021" "11/10/2021"
[11] "9/11/2021" "1/12/2021" "8/12/2021" "2/1/2022"  "2/7/2022"
     "2/8/2022"  "2/10/2022" "1/12/2022" "11/13/2021" "11/18/2021"
[21] "1/28/2022" "9/28/2021" "10/25/2021" "9/30/2021" "9/29/2021"
     "7/15/2021"  "11/16/2021" "8/15/2021" "11/15/2021" "3/30/2021"
[31] "1/20/2021"  "1/22/2022" "8/23/2021" "2/15/2022" "2/18/2022"
     "9/23/2021"  "7/27/2021"  "4/24/2020" "10/16/2021" "12/26/2020"
[41] "1/28/2021"  "9/22/2021" "11/19/2021" "8/17/2021" "4/30/2021"
```

- 8) Thủ dự đoán thời gian nào dịch sẽ giảm tối thiểu hay kết thúc ở các quốc gia nhóm đã phân tích, đưa ra giải thích của nhóm
- 9) Cho nhận xét của các bạn về tình hình dịch theo các quốc mà nhóm đã phân tích
 - Số ca mắc mới hàng ngày tại Việt Nam trong năm 2020 rất ít, tăng đột biến vào giữa năm 2021 và có xu hướng tăng mạnh vào tháng 8/2021 và 9/2021 và có xu hướng giảm sau đó nhưng lại tăng trở lại từ cuối năm 2021 đến đầu 2022
 - Số ca mắc mới hàng ngày tại Indonesia tăng dần đến khoảng 2/2021 đạt đỉnh và sau đó có xu hướng giảm nhưng nhanh chóng tăng trở lại và đạt đỉnh mới vào 7/2021, sau đó số ca nhiễm giảm đi nhanh chóng nhưng bắt đầu tăng đột biến trở lại từ 1/2022 đến 2/2022
 - Số ca mắc mới hàng ngày tại Nhật Bản có xu hướng biến động không đều đến khoảng 11/2021 thì có xu hướng tăng đột biến, từ khoảng đầu đến giữa năm 2021 số ca mắc tiếp tục biến động và tăng đột biến vào 7/2021 và đạt đỉnh vào 8/2021 nhưng giảm mạnh sau đó đến đầu năm 2022 thì tiếp tục tăng mạnh trở lại.
- 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.

6 Hướng dẫn và yêu cầu

6.1 Hướng dẫn

- Cài đặt đồng thời cả R và Rstudio.
- Đọc kĩ và xử lý lại tất cả những thí dụ đã có trong file mẫu.
- Tìm hiểu kĩ cách soạn thảo văn bản bằng LaTeX và cách sử dụng phần mềm R trong các file hướng dẫn và tìm hiểu thêm trong các tài liệu khác.
- Tạo một folder chung chứa mọi thứ cần thiết để share giữa các thành viên trong nhóm trên các cloud services như [Google Drive](#) hay [Dropbox](#)...



- Dùng Doodle để lên kế hoạch họp nhóm.
- Dùng Trello để quản lý project.

6.2 Yêu cầu

Mỗi nhóm, từ 3 đến 6 sinh viên, đề xuất giải pháp. Nhóm cần nộp báo cáo trình bày về lời giải cho các câu hỏi và kết quả thực nghiệm. Đồng thời, nhóm cũng cần nộp source code, và trình bày các kết quả của nhóm trong khoảng 5 minutes.

Báo cáo và slide trình bày cần được viết dưới dạng LaTeX.

- Thời gian làm bài: **Từ ngày 21/02/2022 – 18g00 ngày 20/03/2022.**

Đối với mỗi bài toán, yêu cầu sinh viên trình bày lời giải theo lối truyền thống, sử dụng các công thức, kết quả lý thuyết trong phần kiến thức chuẩn bị. Đồng thời, sau đó trình bày kết quả tính toán và biểu đồ minh họa bằng R.

- Trình bày cả code R và kết quả tính toán trong R giống như file mẫu.
- Viết báo cáo theo đúng **bố cục như trong file mẫu** bằng LaTeX.
- Mỗi nhóm khi nộp bài **cần phải nộp theo file log (nhật ký)** ghi rõ: tiến độ công việc, phân công nhiệm vụ, trao đổi của các thành viên,...

6.3 Nộp bài

- SV chỉ nộp bài qua hệ thống BKEL: nén tất cả các file cần thiết (file .tex, file .R, ...) thành một file tên là “*LOP-NHOM-MADE.zip*”: 1-3456.zip và nộp trong mục Assignment.
- Lưu ý: mỗi nhóm **chỉ cần một thành viên là nhóm trưởng nộp bài**.

7 Cách đánh giá và xử lý gian lận

7.1 Đánh giá

Mỗi bài làm sẽ được đánh giá như sau.

Nội dung	Tỉ lệ điểm (%)
Giải đúng các bài toán bằng công thức và lập luận	30%
Các lệnh (hàm) R được sử dụng đúng đắn và hợp lý	30%
Trình bày kiến thức chuẩn bị rõ ràng, phù hợp	20%
Trình bày văn bản đẹp, đúng chuẩn	20%

7.2 Xử lý gian lận

Bài tập lớn phải được sinh viên (nhóm) TỰ LÀM. Sinh viên (nhóm) sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa các bài thu hoạch (nhất là phần kiến thức chuẩn bị). Trong trường hợp này, **TẤT CẢ** các bài nộp có sự giống nhau đều bị coi là gian lận. Do vậy sinh viên (nhóm) phải bảo vệ bài làm của mình.
- Sinh viên (nhóm) không hiểu bài làm do chính mình viết. Sinh viên (nhóm) có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những gì mình viết.

Bài bị phát hiện gian lận thì sinh viên sẽ bị xử lý theo quy định của nhà trường.



Tài liệu

- [Dal] Dalgaard, P. *Introductory Statistics with R*. Springer 2008.
- [K-Z] Kenett, R. S. and Zacks, S. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*, 2nd ed., John Wiley and Sons, 2014.
- [Ker] Kerns, G. J. *Introduction to Probability and Statistics Using R*, 2nd ed., CRC 2015.