

Visualising COVID-19

JUSTIN LO

2023-12-09

Reading in the dataset of worldwide confirmed cases and confirmed cases of china and non china countries and assigning it to a variable

```
## Rows: 56 Columns: 2
## -- Column specification -----
## Delimiter: ","
## dbl  (1): cum_cases
## date (1): date
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

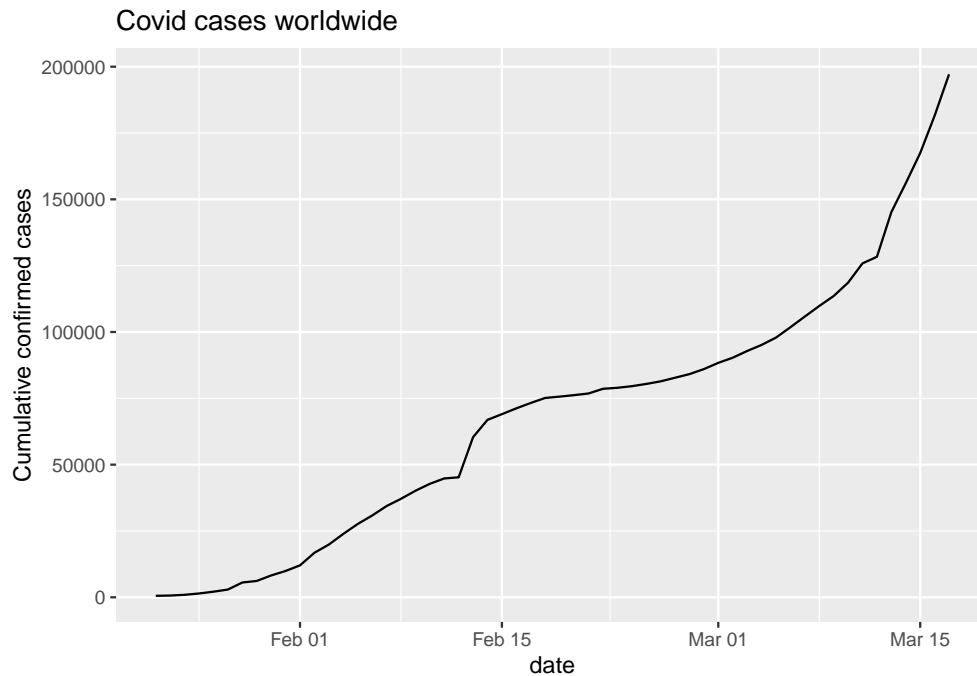
```
## # A tibble: 6 x 2
##   date      cum_cases
##   <date>      <dbl>
## 1 2020-01-22      555
## 2 2020-01-23      653
## 3 2020-01-24      941
## 4 2020-01-25     1434
## 5 2020-01-26     2118
## 6 2020-01-27     2927
```

```
## Rows: 112 Columns: 4
## -- Column specification -----
## Delimiter: ","
## chr  (1): is_china
## dbl  (2): cases, cum_cases
## date (1): date
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

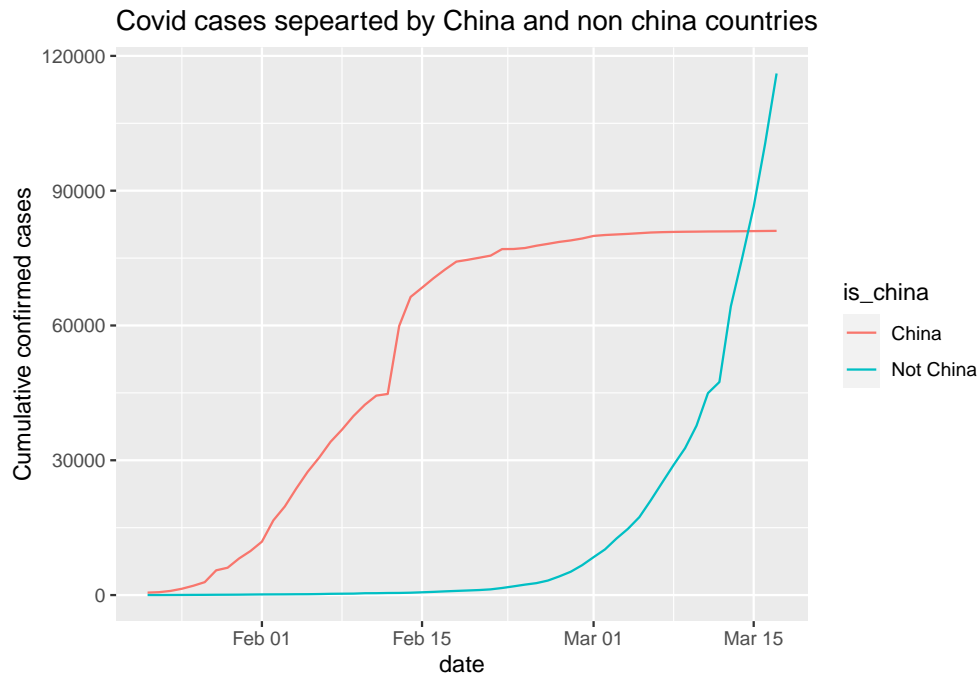
```
## # A tibble: 6 x 4
##   is_china date      cases cum_cases
##   <chr>    <date>    <dbl>    <dbl>
## 1 China  2020-01-22    548      548
## 2 China  2020-01-23     95      643
## 3 China  2020-01-24    277      920
## 4 China  2020-01-25    486     1406
## 5 China  2020-01-26    669     2075
## 6 China  2020-01-27    802     2877
```

Visualization of confirmed cases trend and confirmed cases of china and non china countries

```
plt_cum_confirmed_cases_worldwide<-ggplot(data = confirmed_cases_worldwide, aes(x=date, y=cum_cases))+g  
plt_cum_confirmed_cases_worldwide
```



```
plt_cum_confirmed_cases_china_vs_world <- ggplot(data=confirmed_cases_china_vs_world, aes(x= date, y= c  
geom_line() +  
ylab("Cumulative confirmed cases") +  
ggtitle("Covid cases sepearted by China and non china countries")  
plt_cum_confirmed_cases_china_vs_world
```

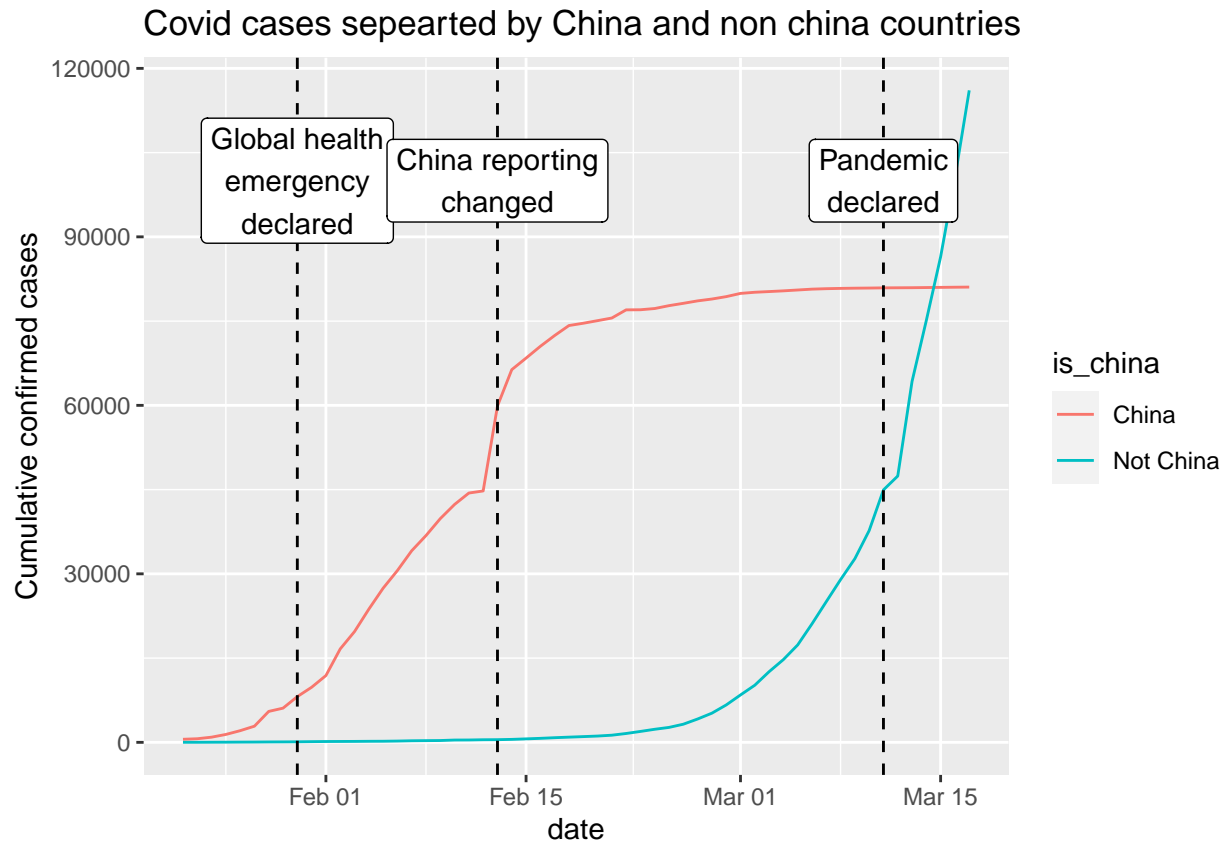


It was then realised that the huge jump in the China line on February 13, 2020 wasn't just a bad day regarding the outbreak; China changed the way it reported figures on that day (CT scans were accepted as evidence for COVID-19, rather than only lab tests).

To reflect this on the plot, we can add x-intercepts and labels to the plot

```
who_events <- tribble(
  ~ date, ~ event,
  "2020-01-30", "Global health\nemergency\ndeclared",
  "2020-03-11", "Pandemic\ndeclared",
  "2020-02-13", "China reporting\nchanged"
) %>%
  mutate(date = as.Date(date))

plt_cum_confirmed_cases_china_vs_world_2<-plt_cum_confirmed_cases_china_vs_world +
  geom_vline(aes(xintercept = date), data = who_events, linetype = "dashed") +
  geom_label(aes(date, label = event), data = who_events, y = 1e5, color="black")
plt_cum_confirmed_cases_china_vs_world_2
```



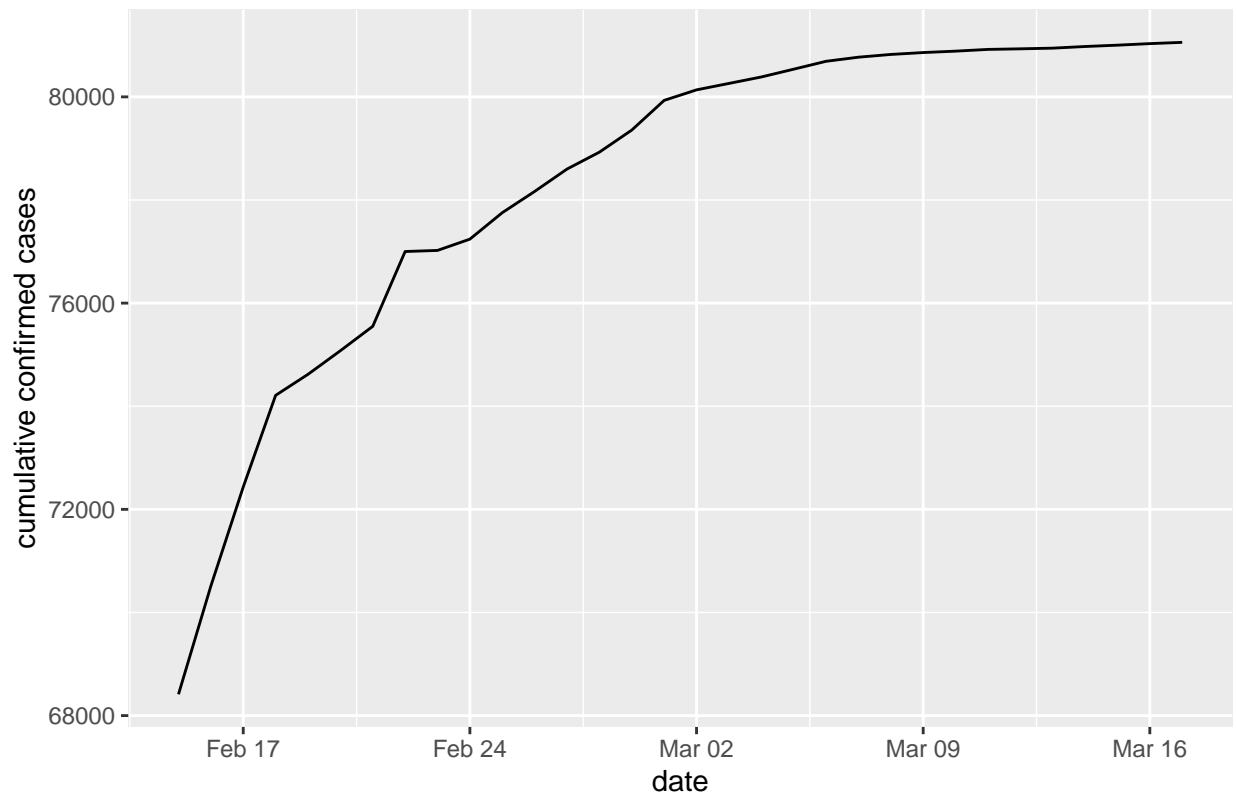
Now, to investigate china's confirmed cases after feb15. I have created a line plot showing the data of china's confirmed cases after feb 15.

```
china_after_feb15<- confirmed_cases_china_vs_world %>%
  filter(is_china=="China", date>="2020-02-15")
head(china_after_feb15)
```

```
## # A tibble: 6 x 4
##   is_china date       cases cum_cases
##   <chr>    <date>     <dbl>   <dbl>
## 1 China   2020-02-15    2055    68413
## 2 China   2020-02-16    2100    70513
## 3 China   2020-02-17    1921    72434
## 4 China   2020-02-18    1777    74211
## 5 China   2020-02-19     408    74619
## 6 China   2020-02-20     458    75077
```

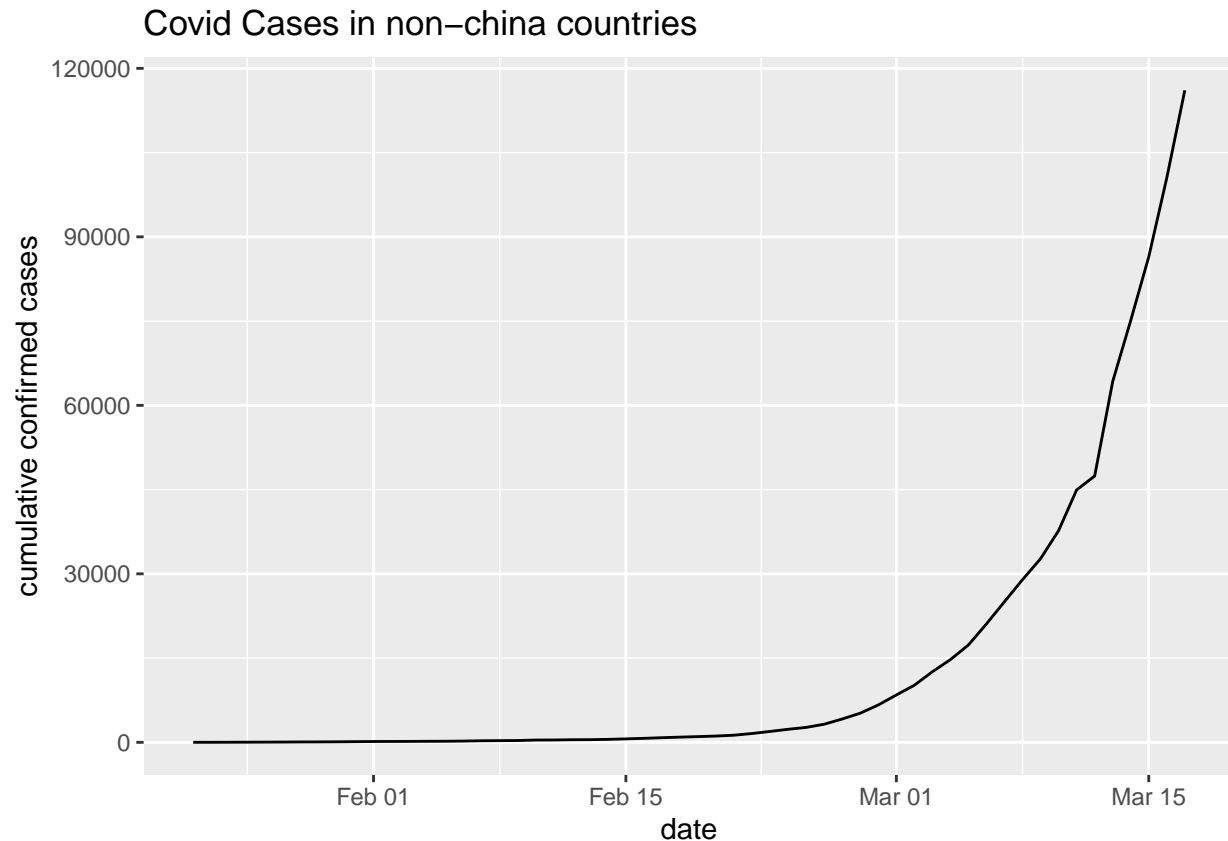
```
plt_cum_confirmed_cases_china_after_feb15<-ggplot(data= china_after_feb15, aes(x=date, y=cum_cases))+ge
plt_cum_confirmed_cases_china_after_feb15
```

Covid cases in China



To create the same plots but for non-china countries

```
no_china<- confirmed_cases_china_vs_world %>%  
  filter(is_china!="China")  
  
yes_china<- confirmed_cases_china_vs_world %>%  
  filter(is_china=="China")  
  
plt_not_china_trend_line<- ggplot(data = no_china, aes(x=date, y=cum_cases)) + geom_line() + ylab("cumulative confirmed cases")  
plt_not_china_trend_line
```



Import data on covid cases by country with China excluded. Then, calculate which country has had the highest cumulative cases

```
confirmed_cases_by_country <- read_csv("confirmed_cases_by_country.csv")
```

```
## Rows: 13272 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr  (2): country, province
## dbl  (2): cases, cum_cases
## date (1): date
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
head(confirmed_cases_by_country)
```

```
## # A tibble: 6 x 5
##   country      province date      cases cum_cases
##   <chr>         <chr>   <date>   <dbl>   <dbl>
## 1 Afghanistan <NA>    2020-01-22  0         0
## 2 Albania      <NA>    2020-01-22  0         0
## 3 Algeria      <NA>    2020-01-22  0         0
## 4 Andorra      <NA>    2020-01-22  0         0
## 5 Antigua and Barbuda <NA>    2020-01-22  0         0
## 6 Argentina    <NA>    2020-01-22  0         0
```

```
top_countries_by_total_cases<-confirmed_cases_by_country %>%
  group_by(country) %>%
  summarize(total_cases = max(cum_cases)) %>%
  top_n(7, total_cases)
```

```
top_countries_by_total_cases
```

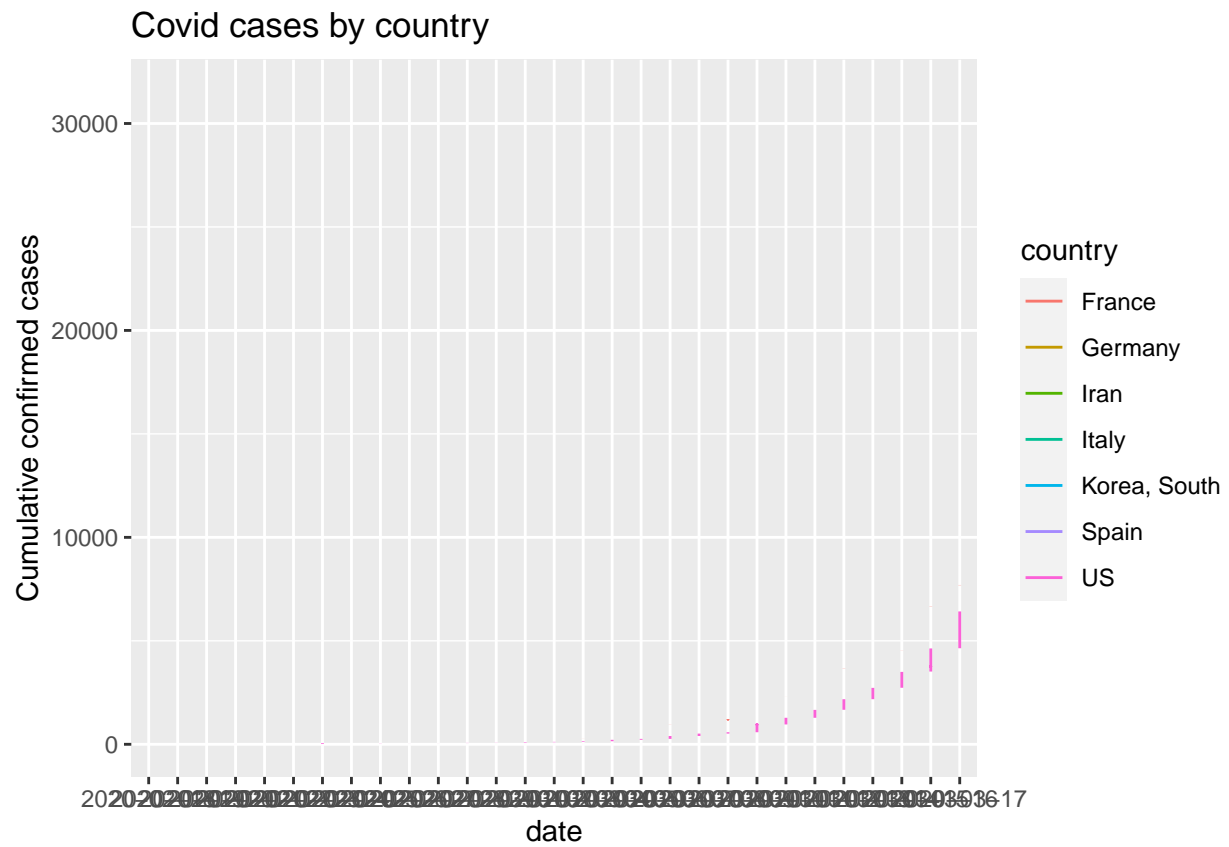
```
## # A tibble: 7 x 2
##   country      total_cases
##   <chr>         <dbl>
## 1 France         7699
## 2 Germany        9257
## 3 Iran          16169
## 4 Italy          31506
## 5 Korea, South   8320
## 6 Spain          11748
## 7 US             6421
```

To create line plots of cumulative confirmed cases by country

```
confirmed_cases_top7_outside_china <- read.csv("confirmed_cases_top7_outside_china.csv")
head(confirmed_cases_top7_outside_china)
```

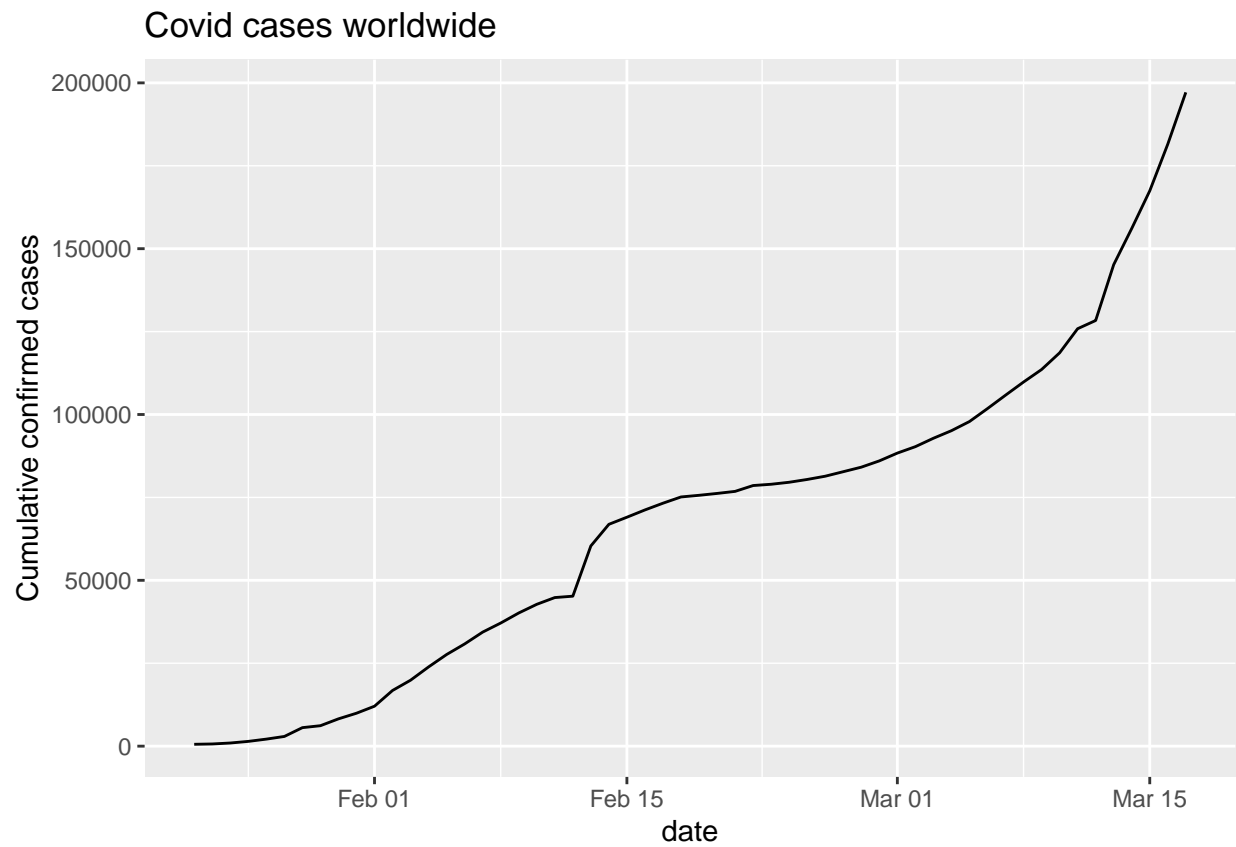
```
##      country      date cum_cases
## 1   Germany 2020-02-18      16
## 2     Iran 2020-02-18       0
## 3    Italy 2020-02-18       3
## 4 Korea, South 2020-02-18     31
## 5    Spain 2020-02-18       2
## 6      US 2020-02-18      13
```

```
plt_cum_confirmed_cases_by_country<-ggplot(confirmed_cases_top7_outside_china, aes(date, cum_cases, col=country)) +
  geom_line() +
  ylab("Cumulative confirmed cases") +
  ggtitle("Covid cases by country")
plt_cum_confirmed_cases_by_country
```

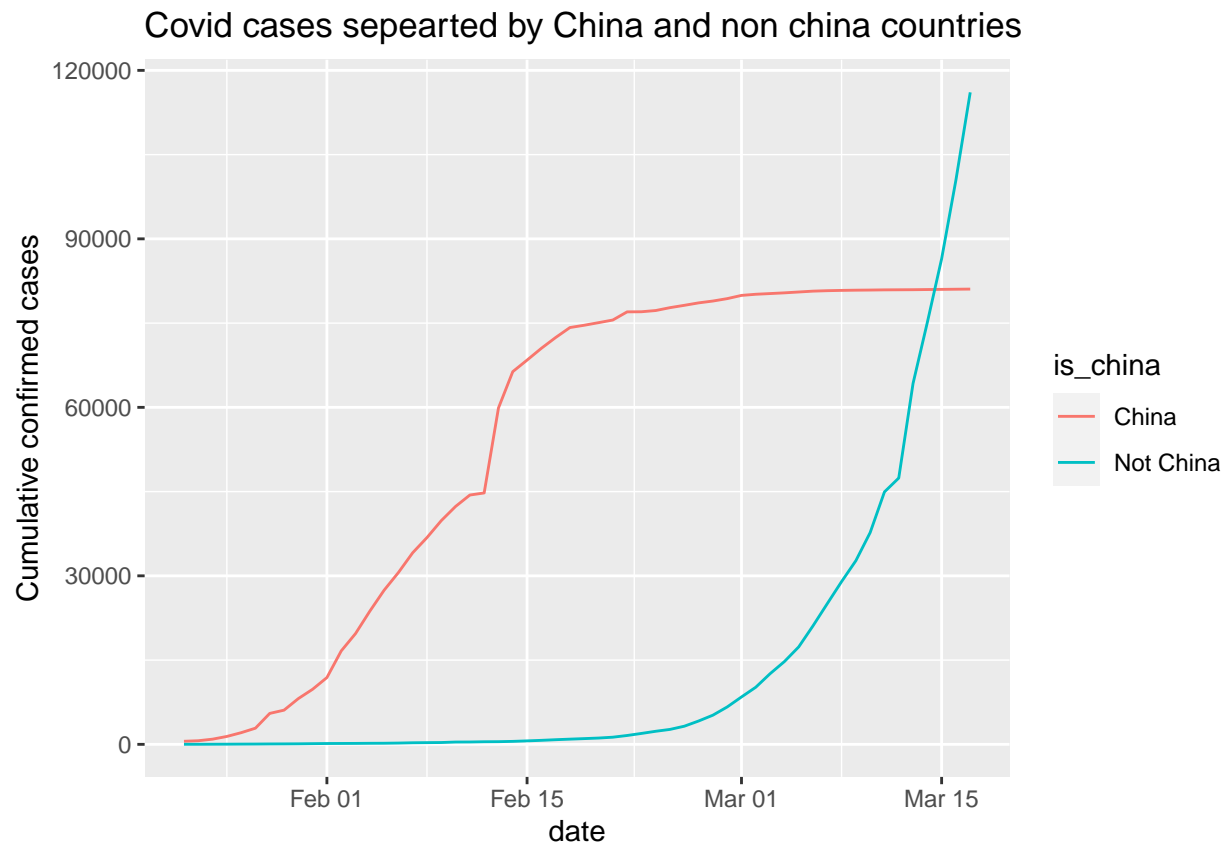


plots i want to show

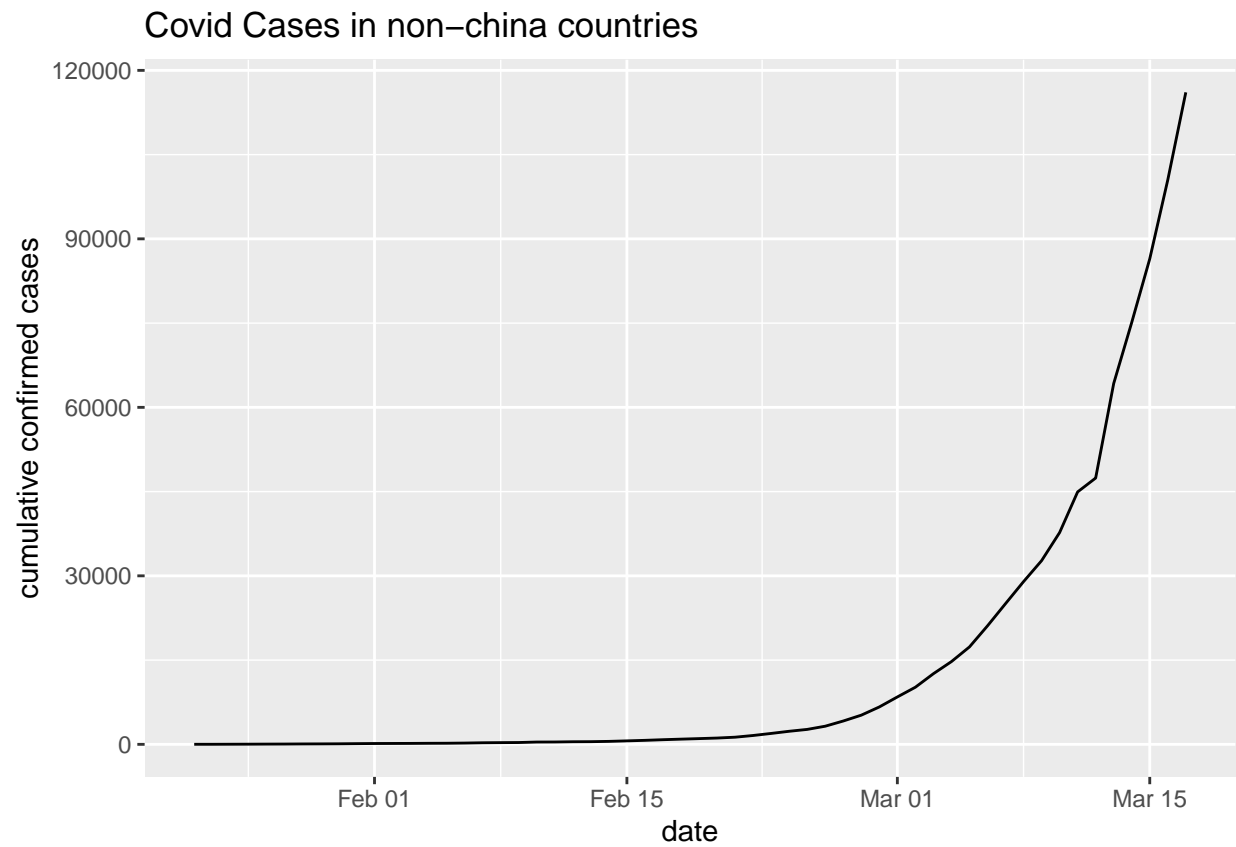
```
plt_cum_confirmed_cases_worldwide<-ggplot(data = confirmed_cases_worldwide, aes(x=date, y=cum_cases))+g
plt_cum_confirmed_cases_worldwide
```

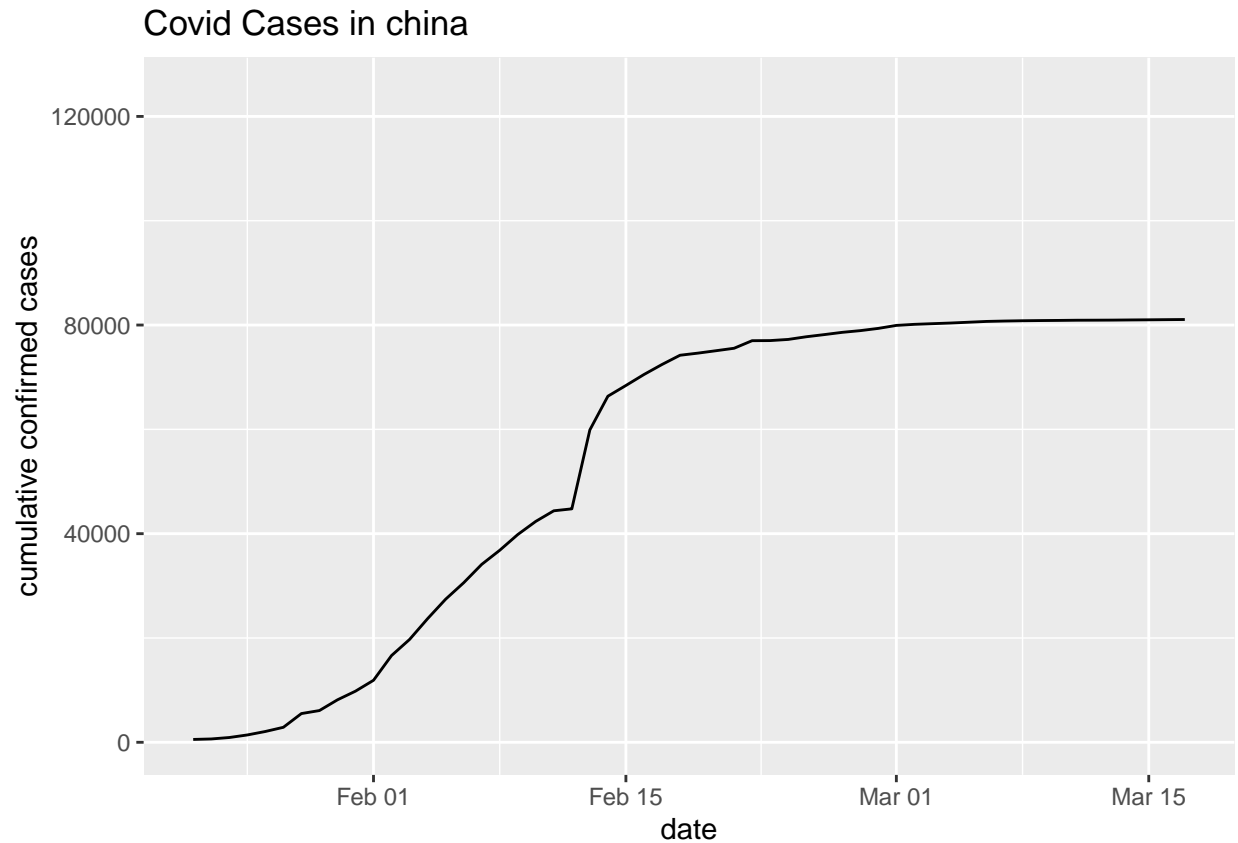
```
plt_cum_confirmed_cases_china_vs_world <- ggplot(data=confirmed_cases_china_vs_world, aes(x= date, y= c
  geom_line() +
  ylab("Cumulative confirmed cases") +
  ggtitle("Covid cases sepearted by China and non china countries")
plt_cum_confirmed_cases_china_vs_world
```



```
plt_not_china_trend_line<- ggplot(data = no_china, aes(x=date, y=cum_cases)) + geom_line() +ylab("cumul.  
plt_not_china_trend_line
```



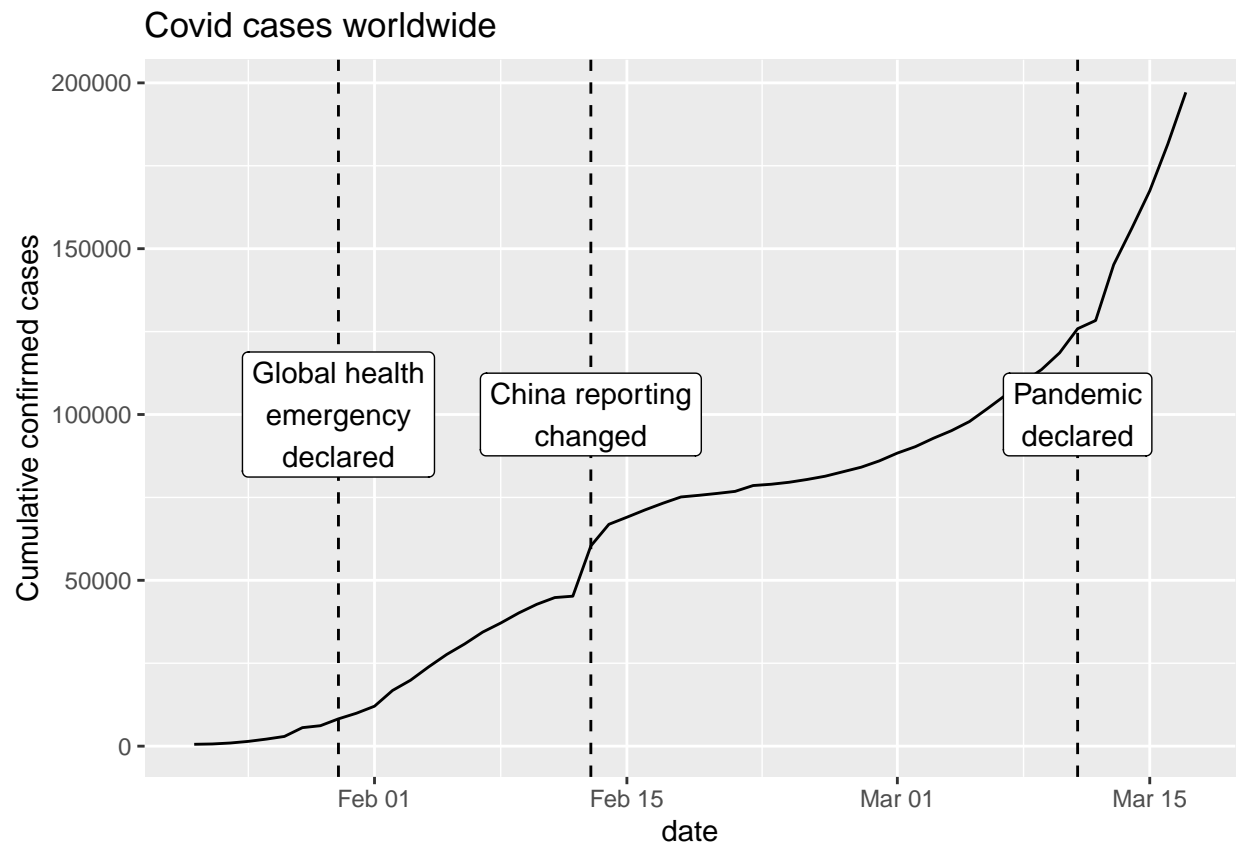
```
plt_yes_china_trend_line<- ggplot(data = yes_china, aes(x=date, y=cum_cases)) + geom_line() +ylab("cumulative confirmed cases")
plt_yes_china_trend_line
```



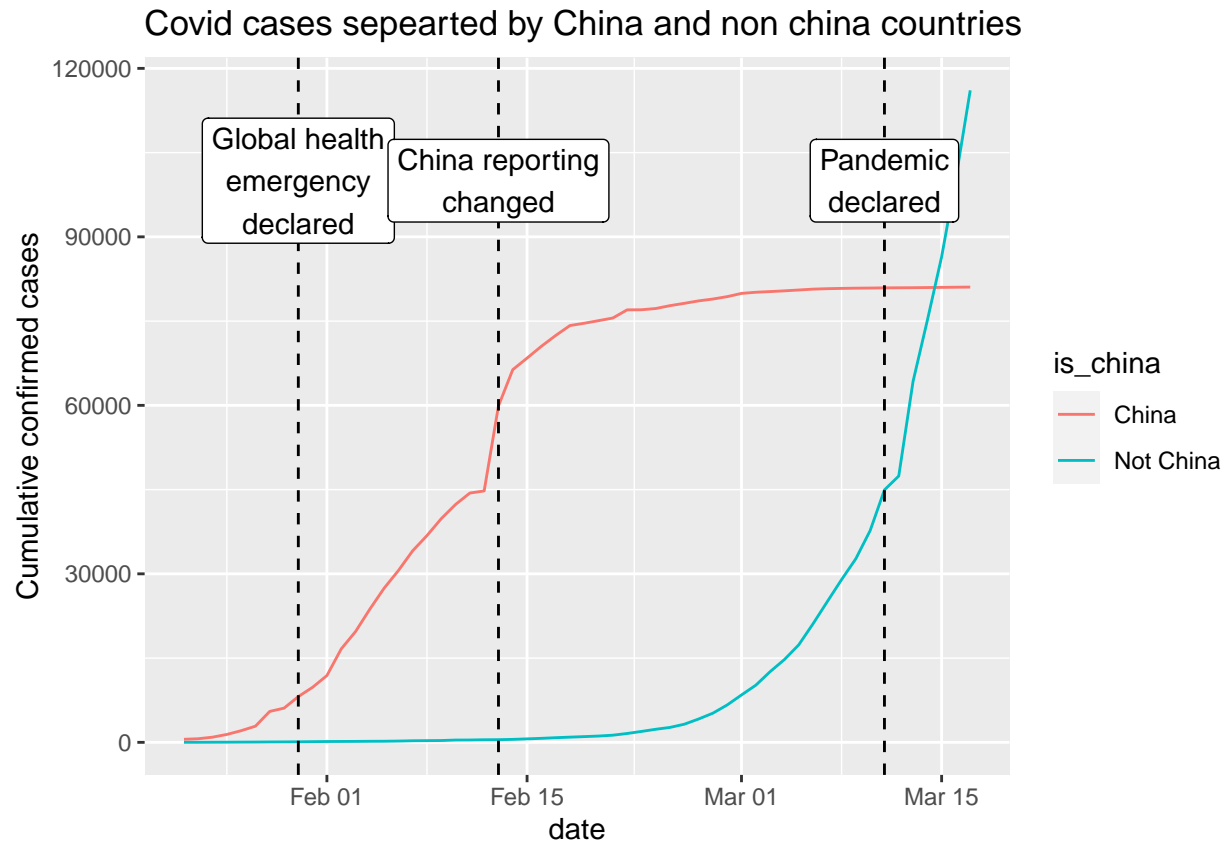
Same plots with legend

```
who_events <- tribble(
  ~ date, ~ event,
  "2020-01-30", "Global health\nemergency\ndeclared",
  "2020-03-11", "Pandemic\ndeclared",
  "2020-02-13", "China reporting\nchanged"
) %>%
  mutate(date = as.Date(date))

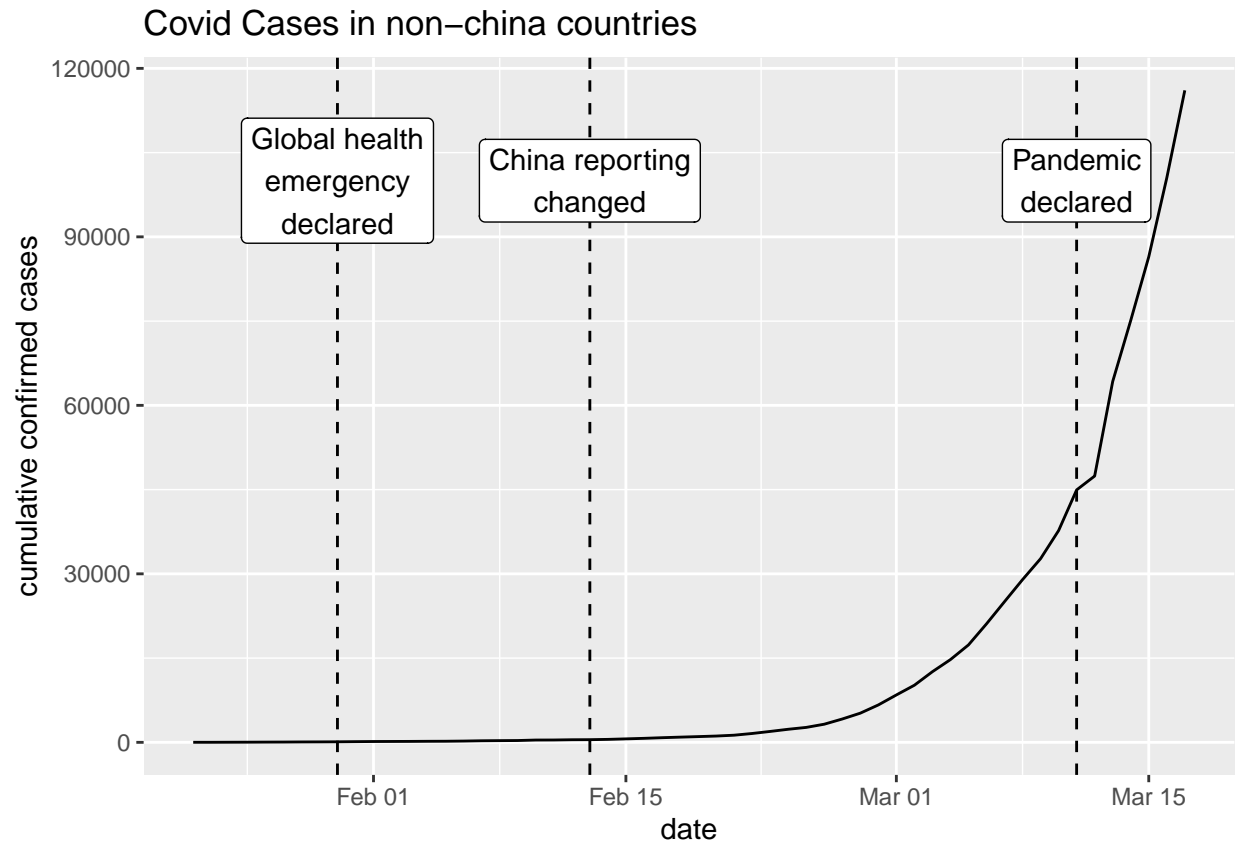
plt_cum_confirmed_cases_worldwide_legend <- ggplot(data = confirmed_cases_worldwide, aes(x=date, y=cum_ca
  geom_vline(aes(xintercept = date), data = who_events, linetype = "dashed") + geom_label(aes(date, label
plt_cum_confirmed_cases_worldwide_legend
```



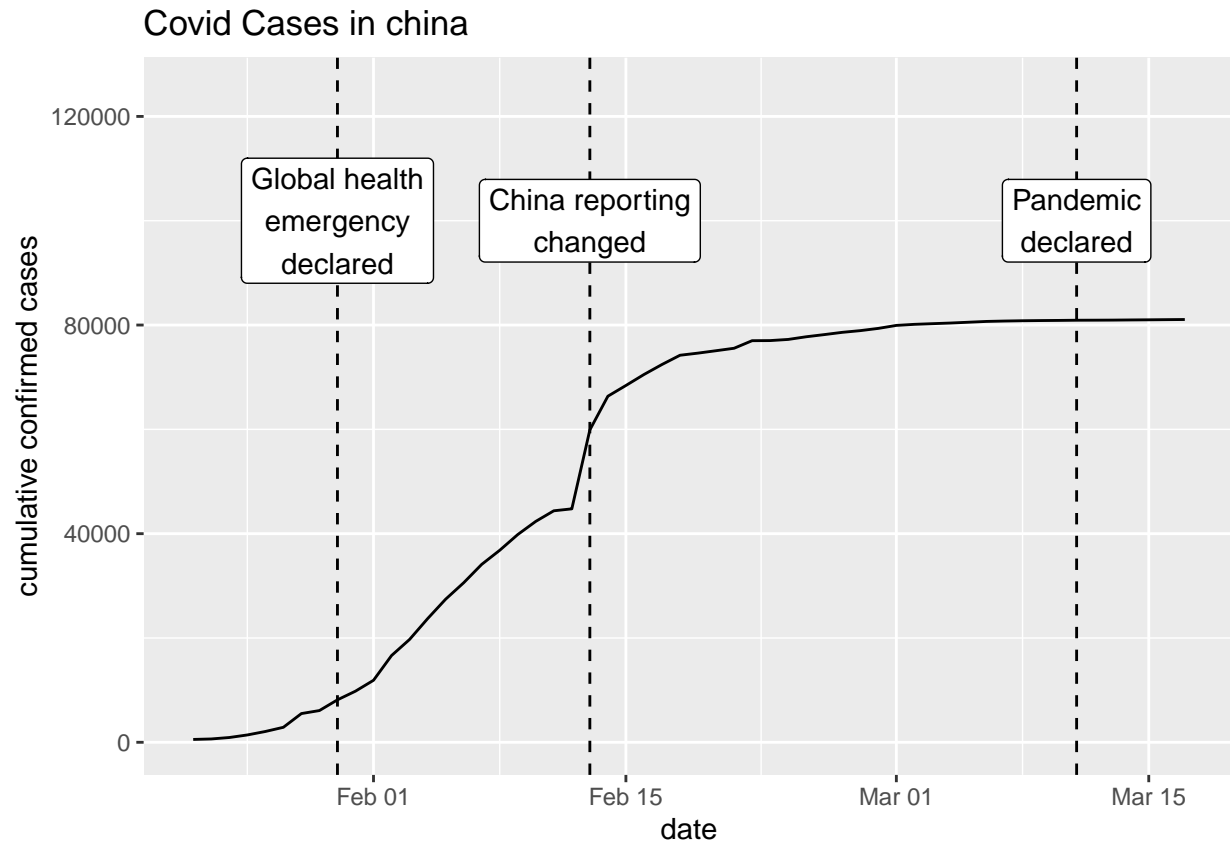
```
plt_cum_confirmed_cases_china_vs_world_legend<-plt_cum_confirmed_cases_china_vs_world +
  geom_vline(aes(xintercept = date), data = who_events, linetype = "dashed") +
  geom_label(aes(date, label = event), data = who_events, y = 1e5, color="black")
plt_cum_confirmed_cases_china_vs_world_legend
```



```
plt_not_china_trend_line_legend<- ggplot(data = no_china, aes(x=date, y=cum_cases)) + geom_line() + ylab(
  geom_vline(aes(xintercept = date), data = who_events, linetype = "dashed") +
  geom_label(aes(date, label = event), data = who_events, y = 1e5, color="black")
plt_not_china_trend_line_legend
```



```
plt_yes_china_trend_line_legend<- ggplot(data = yes_china, aes(x=date, y=cum_cases)) + geom_line() + ylab("cumulative confirmed cases") +
  geom_vline(aes(xintercept = date), data = who_events, linetype = "dashed") +
  geom_label(aes(date, label = event), data = who_events, y = 1e5, color="black")
plt_yes_china_trend_line_legend
```



Shiny 1

```
library(shiny)
library(shinydashboard)
```

```
##
## Attaching package: 'shinydashboard'

## The following object is masked from 'package:graphics':
##
## box
```

```
ui_1 <- fluidPage(
  titlePanel("COVID-19 Data Visualization"),
  tabsetPanel(
    tabPanel("Worldwide Cases", plotOutput("worldCasesPlot")),
    tabPanel("China vs Non-China", plotOutput("chinaVsNonChinaPlot")),
    tabPanel("China Only", plotOutput("chinaOnlyPlot")),
    tabPanel("Non-China Countries", plotOutput("nonChinaPlot"))
  )
)

server_1 <- function(input, output) {
  output$worldCasesPlot <- renderPlot({
```



```

    plt_cum_confirmed_cases_worldwide
  })

  output$chinaVsNonChinaPlot <- renderPlot({
    plt_cum_confirmed_cases_china_vs_world
  })

  output$chinaOnlyPlot <- renderPlot({
    plt_not_china_trend_line
  })

  output$nonChinaPlot <- renderPlot({
    plt_yes_china_trend_line
  })
}

shinyApp(ui = ui_1, server = server_1)

```

Shiny 2

```

ui_2 <- fluidPage(
  titlePanel("COVID-19 Data Visualization"),

  sidebarLayout(
    sidebarPanel(
      selectInput("selectedPlot",
        "Choose a Plot:",
        choices = c("Worldwide Cases" = "world",
                     "China Only" = "chinaOnly",
                     "Non-China Countries" = "nonChina",
                     "China vs Non-China" = "chinaVsNonChina"
        ))
    ),
    mainPanel(
      plotOutput("plotOutput")
    )
  )
)

server_2 <- function(input, output) {

  output$plotOutput <- renderPlot({
    if (input$selectedPlot == "world") {
      plt_cum_confirmed_cases_worldwide
    } else if (input$selectedPlot == "chinaVsNonChina") {
      plt_cum_confirmed_cases_china_vs_world
    } else if (input$selectedPlot == "chinaOnly") {
      plt_not_china_trend_line
    } else if (input$selectedPlot == "nonChina") {
      plt_yes_china_trend_line
    }
  })
}

```

```
}

shinyApp(ui = ui_2, server = server_2)
```

Shiny 3

```
ui_3 <- fluidPage(
  titlePanel("COVID-19 Data Visualization"),

  fluidRow(
    column(12,
      wellPanel(
        p("This dashboard presents a series of plots depicting the spread of COVID-19 global.
      )
    )
  ),

  sidebarLayout(
    sidebarPanel(
      selectInput("selectedPlot",
        "Choose a Plot:",
        choices = c("Worldwide Cases" = "world",
                     "China Only" = "chinaOnly",
                     "Non-China Countries" = "nonChina",
                     "China vs Non-China" = "chinaVsNonChina")),
      checkboxInput("showLegend", "Show important dates", value = FALSE)
    ),
    mainPanel(
      plotOutput("plotOutput"),
      uiOutput("plotDescription")
    )
  )
)

server_3 <- function(input, output) {

  output$plotDescription <- renderUI({
    switch(input$selectedPlot,
      "world" = HTML("Description for Worldwide Cases plot."),
      "chinaVsNonChina" = HTML("Description for China vs Non-China plot."),
      "chinaOnly" = HTML("Description for China Only plot."),
      "nonChina" = HTML("Description for Non-China Countries plot.))
  })

  output$plotOutput <- renderPlot({
    if (input$showLegend) {
      switch(input$selectedPlot,
        "world" = plt_cum_confirmed_cases_worldwide_legend,
        "chinaVsNonChina" = plt_cum_confirmed_cases_china_vs_world_legend,
        "chinaOnly" = plt_not_china_trend_line_legend,
        "nonChina" = plt_yes_china_trend_line_legend)
    } else {
```

```

        switch(input$selectedPlot,
              "world" = plt_cum_confirmed_cases_worldwide,
              "chinaVsNonChina" = plt_cum_confirmed_cases_china_vs_world,
              "chinaOnly" = plt_not_china_trend_line,
              "nonChina" = plt_yes_china_trend_line)
      }
    })
  }

shinyApp(ui = ui_3, server = server_3)

```