Covid19

2022-05-02

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.3.5 ✔ purrr 0.3.4  
## ✔ tibble 3.1.6 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

##   
## 다음의 패키지를 부착합니다: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

## Get current Data in the four files  
  
  
url\_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/"  
  
file\_names <- c("time\_series\_covid19\_confirmed\_global.csv",  
 "time\_series\_covid19\_deaths\_global.csv"  
 )  
urls <- str\_c(url\_in, file\_names)  
  
url\_in\_pop <- "https://raw.githubusercontent.com/datasets/population-growth-estimates-and-projections/master/data/"  
file\_name\_pop <- "population-constant-fertility.csv"  
urls\_pop <- str\_c(url\_in\_pop, file\_name\_pop)

Let’s read in the data and see what we have.

global\_cases <- read\_csv(urls[1])  
global\_deaths <- read\_csv(urls[2])  
world\_pop <- read\_csv(urls\_pop)

world\_pop$Population <- as.double(world\_pop$Population) \* 1000

## Warning: 강제형변환에 의해 생성된 NA 입니다

global\_cases\_pivot <- global\_cases %>%  
 pivot\_longer(cols = -c('Province/State',  
 'Country/Region', Lat, Long),  
 names\_to = "date",  
 values\_to = "cases") %>%  
 select(-c(Lat,Long))  
  
global\_deaths\_pivot <- global\_deaths %>%  
 pivot\_longer(cols = -c('Province/State',  
 'Country/Region', Lat, Long),  
 names\_to = "date",  
 values\_to = "deaths") %>%  
 select(-c(Lat,Long))

global\_cases\_pivot <- global\_cases\_pivot %>%  
 rename(Country\_Region = `Country/Region`, Province\_State = `Province/State`) %>%  
 mutate(date=mdy(date))  
  
global\_deaths\_pivot <- global\_deaths\_pivot %>%  
 rename(Country\_Region = `Country/Region`, Province\_State = `Province/State`) %>%  
 mutate(date=mdy(date))

global\_cases\_pivot <- global\_cases\_pivot %>% filter(cases > 0)  
global\_deaths\_pivot <- global\_deaths\_pivot %>% filter(deaths > 0)

global\_cases\_pivot$Year <- format(global\_cases\_pivot$date, format="%Y")  
global\_deaths\_pivot$Year <- format(global\_deaths\_pivot$date, format="%Y")

global\_cases\_pivot$Country\_Region2 = global\_cases\_pivot$Country\_Region  
global\_deaths\_pivot$Country\_Region2 = global\_deaths\_pivot$Country\_Region

global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Korea, North", "Country\_Region2"] <- "Dem. People's Republic of Korea"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Korea, South", "Country\_Region2"] <- "Republic of Korea"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Bolivia", "Country\_Region2"] <- "Bolivia (Plurinational State of)"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Brunei", "Country\_Region2"] <- "Brunei Darussalam"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Congo (Brazzaville)", "Country\_Region2"] <- "Congo"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Congo (Kinshasa)", "Country\_Region2"] <- "Congo"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Cote d'Ivoire", "Country\_Region2"] <- "Côte d'Ivoire"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Iran", "Country\_Region2"] <- "Iran (Islamic Republic of)"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Kosovo", "Country\_Region2"] <- "Republic of Korea"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Laos", "Country\_Region2"] <- "Lao People's Democratic Republic"   
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Russia", "Country\_Region2"] <- "Russian Federation"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Syria", "Country\_Region2"] <- "Syrian Arab Republic"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Taiwan\*", "Country\_Region2"] <- "China, Taiwan Province of China"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="US", "Country\_Region2"] <- "United States of America"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Venezuela", "Country\_Region2"] <- "Venezuela (Bolivarian Republic of)"  
global\_cases\_pivot[global\_cases\_pivot$Country\_Region2=="Vietnam", "Country\_Region2"] <- "Viet Nam"  
  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Korea, North", "Country\_Region2"] <- "Dem. People's Republic of Korea"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Korea, South", "Country\_Region2"] <- "Republic of Korea"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Bolivia", "Country\_Region2"] <- "Bolivia (Plurinational State of)"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Brunei", "Country\_Region2"] <- "Brunei Darussalam"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Congo (Brazzaville)", "Country\_Region2"] <- "Congo"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Congo (Kinshasa)", "Country\_Region2"] <- "Congo"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Cote d'Ivoire", "Country\_Region2"] <- "Côte d'Ivoire"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Iran", "Country\_Region2"] <- "Iran (Islamic Republic of)"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Kosovo", "Country\_Region2"] <- "Republic of Korea"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Laos", "Country\_Region2"] <- "Lao People's Democratic Republic"   
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Russia", "Country\_Region2"] <- "Russian Federation"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Syria", "Country\_Region2"] <- "Syrian Arab Republic"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Taiwan\*", "Country\_Region2"] <- "China, Taiwan Province of China"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="US", "Country\_Region2"] <- "United States of America"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Venezuela", "Country\_Region2"] <- "Venezuela (Bolivarian Republic of)"  
global\_deaths\_pivot[global\_deaths\_pivot$Country\_Region2=="Vietnam", "Country\_Region2"] <- "Viet Nam"

Below Country has no Population data Antarctica, Burma, Diamon Princess, Kosovo, Moldova, MS Zaandam, Tanzania, West Bank and Gaza

global\_cases\_pivot <- merge(global\_cases\_pivot,world\_pop, by.x=c("Country\_Region2","Year"), by.y=c("Region","Year"), all.x = TRUE)  
global\_deaths\_pivot <- merge(global\_deaths\_pivot,world\_pop, by.x=c("Country\_Region2","Year"), by.y=c("Region","Year"), all.x = TRUE)

global\_cases\_pivot <- global\_cases\_pivot[!(is.na(global\_cases\_pivot$Population)),]  
global\_deaths\_pivot <- global\_deaths\_pivot[!(is.na(global\_deaths\_pivot$Population)),]

Calculate average cases per thousand people to compare countries.

global\_cases\_pivot$case\_per\_thousand <- global\_cases\_pivot$cases / global\_cases\_pivot$Population \* 1000  
global\_deaths\_pivot$death\_per\_thousand <- global\_deaths\_pivot$deaths / global\_deaths\_pivot$Population \* 1000

average\_cases = global\_cases\_pivot %>% group\_by(Country\_Region) %>% summarise(average\_case\_per\_thousand = mean(case\_per\_thousand), .groups = 'drop')  
average\_cases <- average\_cases[!is.na(average\_cases$average\_case\_per\_thousand),]  
average\_cases <- average\_cases[average\_cases$Country\_Region!="Korea, North",]  
worst\_case\_country = average\_cases[average\_cases$average\_case\_per\_thousand==max(average\_cases$average\_case\_per\_thousand), "Country\_Region"]  
best\_case\_country = average\_cases[average\_cases$average\_case\_per\_thousand==min(average\_cases$average\_case\_per\_thousand), "Country\_Region"]  
  
average\_deaths = global\_deaths\_pivot %>% group\_by(Country\_Region) %>% summarise(average\_death\_per\_thousand = mean(death\_per\_thousand), .groups = 'drop')  
average\_deaths <- average\_deaths[!is.na(average\_deaths$average\_death\_per\_thousand),]  
worst\_death\_country = average\_deaths[average\_deaths$average\_death\_per\_thousand==max(average\_deaths$average\_death\_per\_thousand), "Country\_Region"]  
best\_death\_country <- average\_deaths[average\_deaths$average\_death\_per\_thousand==min(average\_deaths$average\_death\_per\_thousand), "Country\_Region"]

North Korea has the lowest average cases per thousand people

library(ggplot2)  
library(hrbrthemes)

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

## Please use hrbrthemes::import\_roboto\_condensed() to install Roboto Condensed and

## if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow

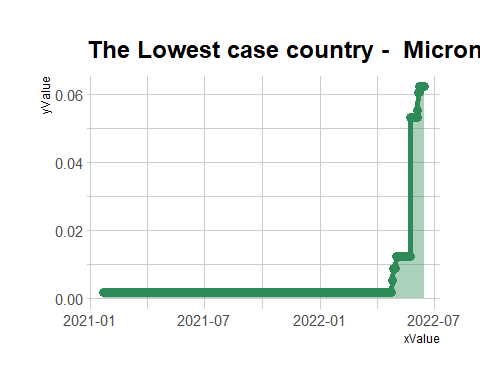
xValue <- global\_cases\_pivot[global\_cases\_pivot$Country\_Region==best\_case\_country[[1]], "date"]  
yValue <- global\_cases\_pivot[global\_cases\_pivot$Country\_Region==best\_case\_country[[1]], "case\_per\_thousand"]  
ggplot(global\_cases\_pivot[global\_cases\_pivot$Country\_Region==best\_case\_country, ], aes(x=xValue, y=yValue)) +  
 geom\_area(fill="seagreen", alpha=0.4) +  
 geom\_line(color="seagreen", size=2) +  
 geom\_point(size=3, color="seagreen") +  
 theme\_ipsum() +  
 ggtitle(paste("The Lowest case country - ", best\_case\_country[[1]]))

## Warning in grid.Call(C\_stringMetric, as.graphicsAnnot(x$label)): 윈도우즈 폰트데  
## 이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_stringMetric, as.graphicsAnnot(x$label)): 윈도우즈 폰트데  
## 이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_stringMetric, as.graphicsAnnot(x$label)): 윈도우즈 폰트데  
## 이터베이스에서 찾을 수 없는 폰트페밀리입니다

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다

## Warning in grid.Call.graphics(C\_text, as.graphicsAnnot(x$label), x$x, x$y, : 윈  
## 도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다

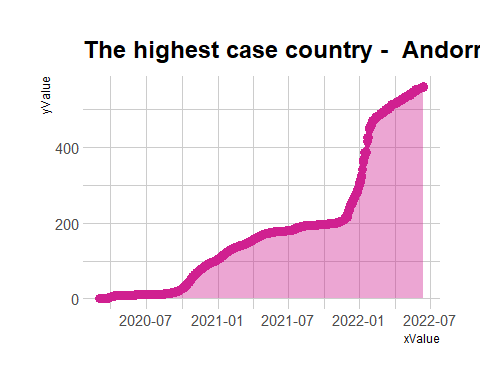


xValue <- global\_cases\_pivot[global\_cases\_pivot$Country\_Region==worst\_case\_country[[1]], "date"]  
yValue <- global\_cases\_pivot[global\_cases\_pivot$Country\_Region==worst\_case\_country[[1]], "case\_per\_thousand"]  
ggplot(global\_cases\_pivot[global\_cases\_pivot$Country\_Region==best\_case\_country, ], aes(x=xValue, y=yValue)) +  
 geom\_area(fill="violetred", alpha=0.4) +  
 geom\_line(color="violetred", size=2) +  
 geom\_point(size=3, color="violetred") +  
 theme\_ipsum() +  
 ggtitle(paste("The highest case country - ", worst\_case\_country[[1]]))

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다

## Warning in grid.Call.graphics(C\_text, as.graphicsAnnot(x$label), x$x, x$y, : 윈  
## 도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : 윈도우  
## 즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다



global\_cases\_pivot <- global\_cases\_pivot %>%  
 unite("Combined\_Key",  
 c(Province\_State, Country\_Region),  
 sep = ", ",  
 na.rm =TRUE,  
 remove = FALSE)