Aggregating\_Covid\_19

Bbosa Robert

29/04/2021

Table of Contents

# Pulling the coronvirus data from John Hopkins repo <https://github.com/CSSEGISandData/COVID-19>

## Pulling confirmed cases

conf\_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/  
master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/  
time\_series\_covid19\_confirmed\_global.csv"  
  
raw\_conf <- read.csv(file = conf\_url, stringsAsFactors = FALSE, header = TRUE)  
  
lapply(1:ncol(raw\_conf), function(i){  
 if(all(is.na(raw\_conf[, i]))){  
 raw\_conf <<- raw\_conf[, -i]  
 return(print(paste("Column", names(raw\_conf)[i], "is missing", sep = " ")))  
 } else {  
 return(NULL)  
 }  
})

## Pulling death cases

death\_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master  
/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_deaths\_global.csv"  
  
raw\_death <- read.csv(file =death\_url, stringsAsFactors = FALSE, fill =FALSE)  
  
lapply(1:ncol(raw\_death), function(i){  
 if(all(is.na(raw\_death[, i]))){  
 raw\_death <<- raw\_death[, -i]  
 return(print(paste("Column", names(raw\_death)[i], "is missing", sep = " ")))  
 } else {  
 return(NULL)  
 }  
})

## Pulling recovered cases

raw\_rec <- read.csv(file = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19  
/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series  
/time\_series\_covid19\_recovered\_global.csv", stringsAsFactors = FALSE, fill =FALSE)  
  
lapply(1:ncol(raw\_rec), function(i){  
 if(all(is.na(raw\_rec[, i]))){  
 raw\_rec <<- raw\_rec[, -i]  
 return(print(paste("Column", names(raw\_rec)[i], "is missing", sep = " ")))  
 } else {  
 return(NULL)  
 }  
})

# Transforming the data from wide to long.

## Creating new data frames for confirmed cases

library(tidyr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(readr)  
  
df\_conf <- raw\_conf[, 1:4]  
  
for(i in 5:ncol(raw\_conf)){  
  
 raw\_conf[,i] <- as.integer(raw\_conf[,i])  
 # raw\_conf[,i] <- ifelse(is.na(raw\_conf[, i]), 0 , raw\_conf[, i])  
 print(names(raw\_conf)[i])  
  
 if(i == 5){  
 df\_conf[[names(raw\_conf)[i]]] <- raw\_conf[, i]  
 }   
 else {  
 df\_conf[[names(raw\_conf)[i]]] <- raw\_conf[, i] - raw\_conf[, i - 1]  
 }  
  
}  
  
  
df\_conf1 <- df\_conf %>% tidyr::pivot\_longer(cols = dplyr::starts\_with("X"),  
 names\_to = "date\_temp",  
 values\_to = "cases\_temp")  
  
df\_conf1$date\_temp <- sub("X","", df\_conf1$date\_temp)  
df\_conf1$date\_temp <- as.Date(df\_conf1$date\_temp, format='%m.%d.%y')

## Creating new data frame for death cases

df\_death <- raw\_death[, 1:4]  
  
for(i in 5:ncol(raw\_death)){  
 print(i)  
 raw\_death[,i] <- as.integer(raw\_death[,i])  
 raw\_death[,i] <- ifelse(is.na(raw\_death[, i]), 0 , raw\_death[, i])  
  
 if(i == 5){  
 df\_death[[names(raw\_death)[i]]] <- raw\_death[, i]  
 } else {  
 df\_death[[names(raw\_death)[i]]] <- raw\_death[, i] - raw\_death[, i - 1]  
 }  
}  
  
  
df\_death1 <- df\_death %>% tidyr::pivot\_longer(cols = dplyr::starts\_with("X"),  
 names\_to = "date\_temp",  
 values\_to = "cases\_temp")  
  
df\_death1$date\_temp <- sub("X","", df\_death1$date\_temp)  
df\_death1$date\_temp <- as.Date(df\_death1$date\_temp, format='%m.%d.%y')

## Creating new data frame for recovered cases

df\_rec <- raw\_rec[, 1:4]  
  
for(i in 5:ncol(raw\_rec)){  
 print(i)  
 raw\_rec[,i] <- as.integer(raw\_rec[,i])  
 raw\_rec[,i] <- ifelse(is.na(raw\_rec[, i]), 0 , raw\_rec[, i])  
  
 if(i == 5){  
 df\_rec[[names(raw\_rec)[i]]] <- raw\_rec[, i]  
 } else {  
 df\_rec[[names(raw\_rec)[i]]] <- raw\_rec[, i] - raw\_rec[, i - 1]  
 }  
}  
  
  
df\_rec1 <- df\_rec %>% tidyr::pivot\_longer(cols = dplyr::starts\_with("X"),  
 names\_to = "date\_temp",  
 values\_to = "cases\_temp")  
  
df\_rec1$date\_temp <- sub("X","", df\_rec1$date\_temp)  
df\_rec1$date\_temp <- as.Date(df\_rec1$date\_temp, format='%m.%d.%y')

# Aggregate the data

## Aggregate daily data for confirmed cases

#Rename the variable for the date  
names(df\_conf1)[names(df\_conf1)=="date\_temp"]<-"date"  
  
#Then aggregate the data  
df\_conf2 <- df\_conf1 %>%  
 dplyr::group\_by(Province.State, Country.Region, Lat, Long, date) %>%  
 dplyr::summarise(cases = sum(cases\_temp)) %>%  
 dplyr::ungroup() %>%  
 dplyr::mutate(type = "confirmed",  
 Country.Region = trimws(Country.Region),  
 Province.State = trimws(Province.State))

## `summarise()` regrouping output by 'Province.State', 'Country.Region', 'Lat', 'Long' (override with `.groups` argument)

head(df\_conf2)

## # A tibble: 6 x 7  
## Province.State Country.Region Lat Long date cases type   
## <chr> <chr> <dbl> <dbl> <date> <int> <chr>   
## 1 "" Afghanistan 33.9 67.7 2020-01-22 0 confirmed  
## 2 "" Afghanistan 33.9 67.7 2020-01-23 0 confirmed  
## 3 "" Afghanistan 33.9 67.7 2020-01-24 0 confirmed  
## 4 "" Afghanistan 33.9 67.7 2020-01-25 0 confirmed  
## 5 "" Afghanistan 33.9 67.7 2020-01-26 0 confirmed  
## 6 "" Afghanistan 33.9 67.7 2020-01-27 0 confirmed

tail(df\_conf2)

## # A tibble: 6 x 7  
## Province.State Country.Region Lat Long date cases type   
## <chr> <chr> <dbl> <dbl> <date> <int> <chr>   
## 1 Zhejiang China 29.2 120. 2021-04-23 0 confirmed  
## 2 Zhejiang China 29.2 120. 2021-04-24 0 confirmed  
## 3 Zhejiang China 29.2 120. 2021-04-25 0 confirmed  
## 4 Zhejiang China 29.2 120. 2021-04-26 1 confirmed  
## 5 Zhejiang China 29.2 120. 2021-04-27 0 confirmed  
## 6 Zhejiang China 29.2 120. 2021-04-28 11 confirmed

## Aggregate daily data for death cases

#Rename the variable for the date  
names(df\_death1)[names(df\_death1)=="date\_temp"]<-"date"  
  
df\_death2 <- df\_death1 %>%  
 dplyr::group\_by(Province.State, Country.Region, Lat, Long, date) %>%  
 dplyr::summarise(cases = sum(cases\_temp)) %>%  
 dplyr::ungroup() %>%  
 dplyr::mutate(type = "death",  
 Country.Region = trimws(Country.Region),  
 Province.State = trimws(Province.State))

## `summarise()` regrouping output by 'Province.State', 'Country.Region', 'Lat', 'Long' (override with `.groups` argument)

head(df\_death2)

## # A tibble: 6 x 7  
## Province.State Country.Region Lat Long date cases type   
## <chr> <chr> <dbl> <dbl> <date> <int> <chr>  
## 1 "" Afghanistan 33.9 67.7 2020-01-22 0 death  
## 2 "" Afghanistan 33.9 67.7 2020-01-23 0 death  
## 3 "" Afghanistan 33.9 67.7 2020-01-24 0 death  
## 4 "" Afghanistan 33.9 67.7 2020-01-25 0 death  
## 5 "" Afghanistan 33.9 67.7 2020-01-26 0 death  
## 6 "" Afghanistan 33.9 67.7 2020-01-27 0 death

tail(df\_death2)

## # A tibble: 6 x 7  
## Province.State Country.Region Lat Long date cases type   
## <chr> <chr> <dbl> <dbl> <date> <int> <chr>  
## 1 Zhejiang China 29.2 120. 2021-04-23 0 death  
## 2 Zhejiang China 29.2 120. 2021-04-24 0 death  
## 3 Zhejiang China 29.2 120. 2021-04-25 0 death  
## 4 Zhejiang China 29.2 120. 2021-04-26 0 death  
## 5 Zhejiang China 29.2 120. 2021-04-27 0 death  
## 6 Zhejiang China 29.2 120. 2021-04-28 0 death

## Aggregate daily data for recovered cases

#Rename the variable for the date  
names(df\_rec1)[names(df\_rec1)=="date\_temp"]<-"date"  
  
df\_rec2 <- df\_rec1 %>%  
 dplyr::group\_by(Province.State, Country.Region, Lat, Long, date) %>%  
 dplyr::summarise(cases = sum(cases\_temp)) %>%  
 dplyr::ungroup() %>%  
 dplyr::mutate(type = "recovered",  
 Country.Region = trimws(Country.Region),  
 Province.State = trimws(Province.State))

## `summarise()` regrouping output by 'Province.State', 'Country.Region', 'Lat', 'Long' (override with `.groups` argument)

head(df\_rec2)

## # A tibble: 6 x 7  
## Province.State Country.Region Lat Long date cases type   
## <chr> <chr> <dbl> <dbl> <date> <int> <chr>   
## 1 "" Afghanistan 33.9 67.7 2020-01-22 0 recovered  
## 2 "" Afghanistan 33.9 67.7 2020-01-23 0 recovered  
## 3 "" Afghanistan 33.9 67.7 2020-01-24 0 recovered  
## 4 "" Afghanistan 33.9 67.7 2020-01-25 0 recovered  
## 5 "" Afghanistan 33.9 67.7 2020-01-26 0 recovered  
## 6 "" Afghanistan 33.9 67.7 2020-01-27 0 recovered

tail(df\_rec2)

## # A tibble: 6 x 7  
## Province.State Country.Region Lat Long date cases type   
## <chr> <chr> <dbl> <dbl> <date> <int> <chr>   
## 1 Zhejiang China 29.2 120. 2021-04-23 0 recovered  
## 2 Zhejiang China 29.2 120. 2021-04-24 0 recovered  
## 3 Zhejiang China 29.2 120. 2021-04-25 0 recovered  
## 4 Zhejiang China 29.2 120. 2021-04-26 0 recovered  
## 5 Zhejiang China 29.2 120. 2021-04-27 0 recovered  
## 6 Zhejiang China 29.2 120. 2021-04-28 2 recovered

# Aggregate all the data frames into one.

coronavirus <- dplyr::bind\_rows(df\_conf2, df\_death2, df\_rec2) %>% as.data.frame()

# Export the coronavirus into a working directory

write.csv(coronavirus,"C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/coronavirus.csv", row.names = FALSE)  
  
writexl::write\_xlsx(x = coronavirus, path = "C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/coronavirus.xlsx", col\_names = TRUE)