Contents

COVIZ-19: data visualizations related to coronavirus COVID-10 a.k.a. SARS 2.0 1

COVIZ-19: data visualizations related to coronavirus COVID-10 a.k.a. SARS 2.0

By Brandon Janes

```
#rm(list=ls())
#setwd("/Users/brandonjanes/Desktop/coVIZ/coVIZ-19")

library(ggplot2)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------- tidyverse 1.2.1 --
```

```
## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   0.8.3     v dplyr   0.8.1
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::first()  masks xts::first()
## x dplyr::lag()    masks stats::lag()
## x dplyr::last()   masks xts::last()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(dplyr)

library(tidyr)

library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose

## The following objects are masked from 'package:xts':
##
##     first, last
```

```
confirmed <- read.csv("./data/csse_covid_19_time_series/time_series_19-covid-Confirmed.csv", stringsAsF
# CONFIRMED TIME SERIES PLOT
# eliminated two inner string columns
#confirm <- select(confirmed, -(Country.Region:Long))
confirm <- select(confirmed, -(Lat:Long))
matriz <- select(confirm, -(Province.State))
matriz[-1] = as.numeric(unlist(matriz[-1]))
matriz[is.na(matriz)] <- 0
#groupby to combine all into Counrty.Region
df <- matriz %>% group_by(Country.Region) %>% summarise_all(funs(sum))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## please use list() instead
##
##   # Before:
##   funs(name = f(.))
##
##   # After:
##   list(name = ~ f(.))
## This warning is displayed once per session.
```

```r
# reflect over y = -x
transpose_df <- as.data.frame(t(df))
# create function to move the first row (location names) to the names of the columns
header.true <- function(df) {
  names(df) <- as.character(unlist(df[1,]))
  df[-1,]
}
transpose_header <- header.true(transpose_df)
date <- rownames(transpose_header)
# create a new column of dates as different datatype
rownames(transpose_header) <- NULL
# format column for countries as numeric
country <- colnames(transpose_header)
country <- country[-c(1)]
for (i in country)
{
    transpose_header[[i]] <- as.numeric(as.character(transpose_header[[i]]))
}
## Finally, have dates and countries together
## side by side
corona <- cbind(date,transpose_header)

# clean string in dates
corona$date = sub('X', '', corona$date)
corona$date<-substr(corona$date,0,7)
corona$date = as.Date(corona$date, format='%m.%d.%y')
##############################
# Here's the final object
covid <- corona

############################
```

```r
#ggplot(covid,aes(x=date, y=Argentina)) +   geom_line(color="red")+
#theme_classic()+ labs(y = "Confirmed cases in the Argentina", x = "total cases", title = "Argentina ti

confirm <- select(confirmed, -(Lat:Long))
matriz <- select(confirm, -(Province.State))
matriz[-1] = as.numeric(unlist(matriz[-1]))
matriz[is.na(matriz)] <- 0
# group by
df <- matriz %>% group_by(Country.Region) %>% summarise_all(funs(sum))

transpose_df <- as.data.frame(t(df))

# make country names column headers
header.true <- function(df) {
  names(df) <- as.character(unlist(df[1,]))
  df[-1,]
}
transpose_header <- header.true(transpose_df)
# filter for the countries we want to see
new_df <- transpose_header %>% select(Argentina,Brazil,Chile)
# make the index the first row
```

```r
d<-setDT(new_df, keep.rownames = TRUE)[]

# flatten the matrix
mdata <- melt(d, id=c("rn"))

#### rename columns
df2 <- rename(mdata, date = rn)
df3 <- rename(df2, country = variable)
df4 <- rename(df3, confirmed = value)
# clean for dates
df4$date = sub('X', '', df4$date)
df4$date = as.Date(df4$date, format='%m.%d.%y')
# set classes
df4$confirmed = as.numeric(as.character((df4$confirmed)))
#df4$confirmed = as.integer(df4$confirmed)
df4$country = as.character(df4$country)
#df4$country = as.factor(df4$country)

###################################
```

```r
ggplot(df4, aes(date,confirmed, color=country)) + geom_line() + theme_classic()+ labs(y = "confirmed cas
```