National Technical University of Athens

Programming Tools and Technologies for Data Science

Exploratory Data Analysis
of CSSE's COVID-19 Dataset

Christos Katsakioris
ckatsak@cslab.ece.ntua.gr
03002964

PhD Candidate
Computing Systems Laboratory
School of Electrical and Computer Engineering

Athens, Greece

Thursday 14th January, 2021

Table of Contents

List of Figures

1 Introduction

This work is a basic Exploratory Data Analysis of the COVID-19 Data provided by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University[1].

For the development, I leveraged the power of the R language on a GNU/Linux `x86_64` platform, as provided through the official `r-base` Docker images[2] of the project (which are based on the Debian `testing` release), as well as the `Vim`[3] text editor.

The development has relied upon the modern principles of structured programming and taken place in a modular, function-based fashion, rather than employing a non-modular, top-down scripting process. As a disclaimer, and with respect to this assignment's requirements to include the code within this document *and* to not exceed the limit of 15 pages, this practice might sometimes be convenient, but it might as well sometimes be not. In any case, the source code has been made publicly available in its whole for further inspection; it is hosted on GitHub and can be accessed at https://github.com/ckatsak/covid-eda-r.

2 Initial Processing

2.1 Entry Point

Omitting the shebang declaration, the introductory and documentation comments and the definition of global constants, we begin by importing the libraries upon which there is dependence:

```
1 library(data.table) # for all sorts of processing
2 library(lubridate)  # for `mdy()`
3 library(ggplot2)    # for plotting
```

The number of external dependencies has been deliberately kept to a minimum, to ease our procedure's reproducibility in the context of the course.

At the bottom of the source code file, we define the entry point of our program. This is merely an output width configuration for the case of interactive execution, or a `main()` function otherwise.

```
1 main <- function() {
2   eda(processing()$dt)
3 }
4
5 if (interactive()) {
6   options("width" = 120)
7 } else {
8   main()
9 }
```

2.2 Processing Steps

Let us start by focusing on the processing of the given dataset following the given instruction steps of the assignment. All of the processing has been included in a single R function, `processing()`.

```r
processing <- function() {
  # (Down)load the latest data into a new data.table, excluding some columns
  exclusions <- c("Province/State", "Lat", "Long")
  confirmed  <- fread(CONFIRMED_URL, header = TRUE, drop = exclusions)
  deaths     <- fread(DEATHS_URL,    header = TRUE, drop = exclusions)

  # Appropriately rename the variable related to the country
  setnames(confirmed, "Country/Region", "Country")
  setnames(deaths,    "Country/Region", "Country")

  # Reshape each data.table from wide to long format
  confirmed <- melt(confirmed, id.vars = c("Country"), variable.name = "date",
                    value.name = "confirmed", variable.factor = FALSE)
  deaths    <- melt(deaths,    id.vars = c("Country"), variable.name = "date",
                    value.name = "deaths",    variable.factor = FALSE)

  # Convert each date variable from character to a date object using `mdy()`
  confirmed <- confirmed[, date := mdy(date)]
  deaths    <- deaths[   , date := mdy(date)]

  # Group them by (Country, date) summing deaths
  confirmed <-confirmed[, .(confirmed = sum(confirmed)), by = .(Country, date)]
  deaths    <-deaths[   , .(deaths    = sum(deaths)),    by = .(Country, date)]

  # Merge the two datasets into one
  dt <- merge(confirmed, deaths)

  # Calculate the total number of confirmed cases as well as the total number
  # of deaths, worldwide
  #
  # NOTE: Since the recorded data are cumulative, σ(most_recent) -> aggregate
  confirmed <- sum(dt[date == max(date)][, confirmed])
  deaths    <- sum(dt[date == max(date)][, deaths])

  # Sort by country and date
  dt <- dt[order(Country, date)]

  # Calculate daily increases
  dt[, ":="(
       confirmed.ind = confirmed - shift(confirmed, 1, type = "lag", fill = 0),
       deaths.inc    = deaths    - shift(deaths,    1, type = "lag", fill = 0)
     ),
     by = .(Country)
  ]

  # Return the results as a list
  ret <- list(confirmed, deaths, dt)
```

```
  names(ret) <- c("confirmed", "deaths", "dt")
  ret
}
```

Rather than repeating the explanation of the implementation of each step, we have annotated the source code in the above listing with comments that indicate the effects of each of the included statements. We explicitly note, however, that step 1 has been merged with the code for the initial retrieval of the data in the beginning of the function, step 3 has preceded step 2 in our implementation, and step 10 leverages `data.table`'s `shift()` function specifying the `type` parameter equal to `"lag"` (rather than employing the `lag()` function of the `stats` package).

3 Aggregate Stats

The EDA is bootstrapped through the following function, which is called by function `main()`:

```
eda <- function(dt) {
  plot_summary(calc_summary(dt))
  plot_seasons(determine_hemispheres(dt))
}
```

This section focuses on the first part of the conducted EDA, which addresses aggregate statistics deduced from the dataset. These are calculated in `calc_summary()` and plotted (after being further processed) in `plot_summary()`.

```
calc_summary <- function(dt) {
  countries <- unique(dt$Country)
  aggr_dt <- function(group) {
    dt[Country %in% group,
       .(
         total.deaths         = sum(deaths.inc),
         total.confirmed      = sum(confirmed.ind),
         total.ratio          = sum(deaths.inc) / sum(confirmed.ind),
         mean.daily.confirmed = mean(confirmed.ind),
         mean.daily.deaths    = mean(deaths.inc)
        ),
       by = .(Country)]
  }
  all_agdt <- aggr_dt(countries)
  mean_dt <- function(name, group) {
    all_agdt[Country %in% group,
             .(
               Country              = name,
               total.deaths         = sum(total.deaths),
               total.confirmed      = sum(total.confirmed),
               total.ratio          = mean(total.ratio),
               mean.daily.confirmed = mean(mean.daily.confirmed),
               mean.daily.deaths    = mean(mean.daily.deaths)
```

```
24                )]
25    }
26    world <- mean_dt("WORLD", countries)
27    eu    <- mean_dt("EUROPEAN UNION", EU)
28    nord  <- mean_dt("NORDIC COUNTRIES", NORDIC)
29    brics <- mean_dt("BRICS", BRICS)
30    balk  <- mean_dt("BALKAN COUNTRIES", BALKANS)
31    gulf  <- mean_dt("GULF COUNTRIES", GULF)
32
33    rbind(all_agdt, world, eu, nord, brics, balk, gulf)
34  }
```

Function `calc_sum()`, using the auxiliary function `aggr_dt` (lines 3-13) and the closure `mean_dt` over the local variable `all_agdt` (lines 15-25), calculates the aggregate statistics for some predefined geographic and economic groups of countries that would be interesting to compare (lines 26-31), and returns them through the enrichment of the initially given `data.table` (line 33).

The resulting `data.table` is piped into the `plot_summary` function, which further transforms it to facilitate and conduct the creation of the plots.

In the following listing, lines 2-22 create a temporary factor variable and two stack barplots (Figures 3 and 4) for the total confirmed COVID-19 cases and the total deaths attributed to the virus, respectively, for three major economic groups of countries: the European Union, the BRICS (Brazil, Russia, India, China and South Africa) and the US (which is considered as a whole separate economy here, due to both its vast size and its unique structure).

Subsequently, lines 24-44 repeat a similar procedure, though this time for geographic groups: countries that are mostly or partially (>= 25% of their area) in the Balkan Peninsula according to [4], the countries that constitute the Arab states of the Persian Gulf according to [5] and the Nordic countries. These are presented in Figures 1 and 2.

```
1   plot_summary <- function(dt) {
2     # Stack barplots for economic groups
3     econ <- dt[Country %in% c(EU, BRICS, "US")
4                ][, econ.grp := as.factor(ifelse(Country %in% EU,
5                                                 "European Union",
6                                                 ifelse(Country %in% BRICS,
7                                                        "BRICS",
8                                                        "US")))]
9     ggplot(econ) + aes(x = econ.grp, y = total.confirmed, fill = Country) +
10      geom_bar(position = "stack", stat = "identity") +
11      labs(title = "Confirmed COVID-19 cases per economic group") +
12      labs(subtitle = "Dataset from CSSE, Johns Hopkins University") +
13      labs(x = "", y = "Confirmed COVID-19 Cases") + theme_bw() +
14      geom_text(aes(label = Country), position = position_stack(vjust=.5), size=2)
15    ggsave("aggr_econ_conf.png")
16    ggplot(econ) + aes(x = econ.grp, y = total.deaths, fill = Country) +
17      geom_bar(position = "stack", stat = "identity") +
18      labs(title = "COVID-19 deaths per economic group") +
19      labs(subtitle = "Dataset from CSSE, Johns Hopkins University") +
```

```
    labs(x = "", y = "COVID-19 Deaths") + theme_bw() +
    geom_text(aes(label = Country), position = position_stack(vjust=.5), size=2)
  ggsave("aggr_econ_deaths.png")

  # Stack barplots for geographic groups
  geogr <- dt[Country %in% c(BALKANS, NORDIC, GULF)
              ][, geo.grp := as.factor(ifelse(Country %in% BALKANS,
                                              "Balkan",
                                              ifelse(Country %in% NORDIC,
                                                     "Nordic",
                                                     "Gulf")))]
  ggplot(geogr) + aes(x = geo.grp, y = total.confirmed, fill = Country) +
    geom_bar(position = "stack", stat = "identity") +
    labs(title = "Confirmed COVID-19 cases per geographic group") +
    labs(subtitle = "Dataset from CSSE, Johns Hopkins University") +
    labs(x = "", y = "Confirmed COVID-19 Cases") + theme_bw() +
    geom_text(aes(label = Country), position = position_stack(vjust=.5), size=2)
  ggsave("aggr_geo_conf.png")
  ggplot(geogr) + aes(x = geo.grp, y = total.deaths, fill = Country) +
    geom_bar(position = "stack", stat = "identity") +
    labs(title = "COVID-19 deaths per geographic group") +
    labs(subtitle = "Dataset from CSSE, Johns Hopkins University") +
    labs(x = "", y = "COVID-19 Deaths") + theme_bw() +
    geom_text(aes(label=Country), position=position_stack(vjust=.5), size=2)
  ggsave("aggr_geo_deaths.png")

  # Lollipop plots for death-to-confirmed ratios
  aggrs <- c("WORLD", "EUROPEAN UNION", "NORDIC COUNTRIES", "BRICS",
             "BALKAN COUNTRIES", "GULF COUNTRIES", "US")
  bottom25 <- dt[!Country %in% c("MS Zaandam", "Diamond Princess")
                 ][order(-total.ratio)][1:25]
  ratioz <- rbind(dt[Country %in% aggrs], bottom25)[order(-total.ratio)]
  ratioz$Country <- reorder(ratioz$Country, ratioz$total.ratio)
  ggplot(ratioz) + aes(x = Country, y = total.ratio) +
    geom_segment(aes(x = Country, xend = Country, y = 0, yend = total.ratio),
                 color = ifelse(ratioz$Country %in% aggrs, "blue", "dark red"),
                 size  = ifelse(ratioz$Country %in% aggrs, 1.3, 0.7) ) +
    geom_text(aes(label=sprintf("%d out of %d", total.deaths,total.confirmed)),
              vjust = .5, hjust = -.15, color="magenta", size = 1.5) +
    ylim(c(0, .31)) +
    geom_point(color = ifelse(ratioz$Country %in% aggrs, "blue", "dark red"),
               size  = ifelse(ratioz$Country %in% aggrs, 3, 1)) +
    coord_flip() + labs(x = "", y = "Ratio") + theme_bw() +
    labs(title = "Ratio of COVID-19 Deaths to Confirmed COVID-19 Cases") +
    labs(subtitle = "Dataset from CSSE, Johns Hopkins University")
  ggsave("aggr_ratio_lollipop.png")
}
```

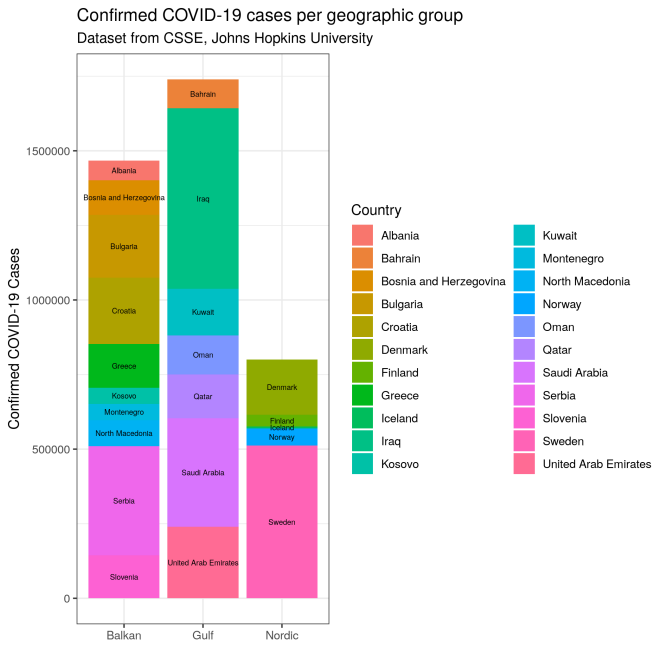

Figure 1: Confirmed COVID-19 cases per geographic group

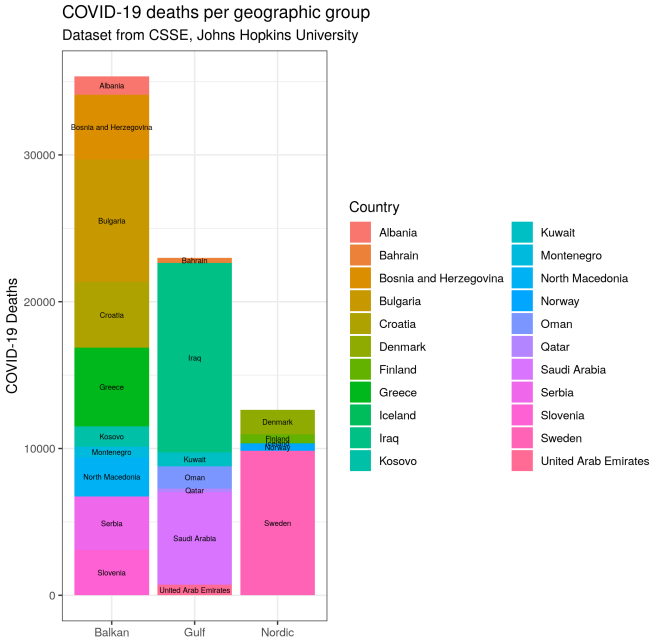

Figure 2: COVID-19 deaths per geographic group

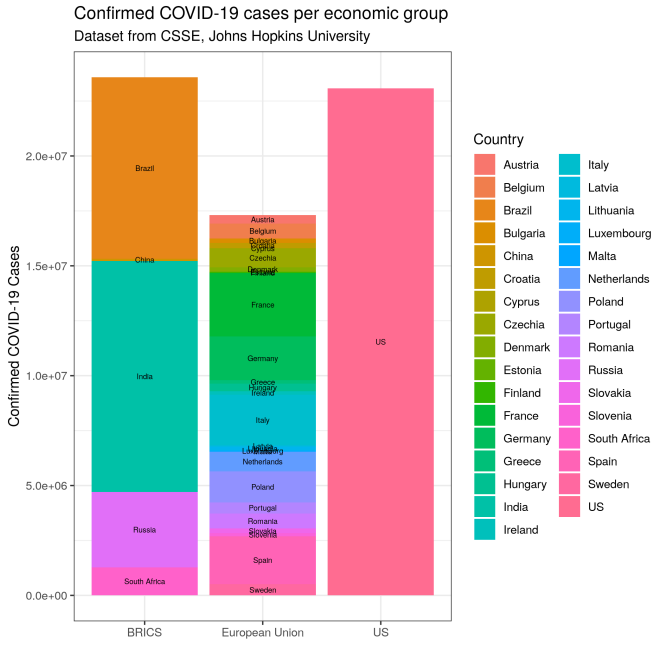

Figure 3: Confirmed COVID-19 cases per economic group

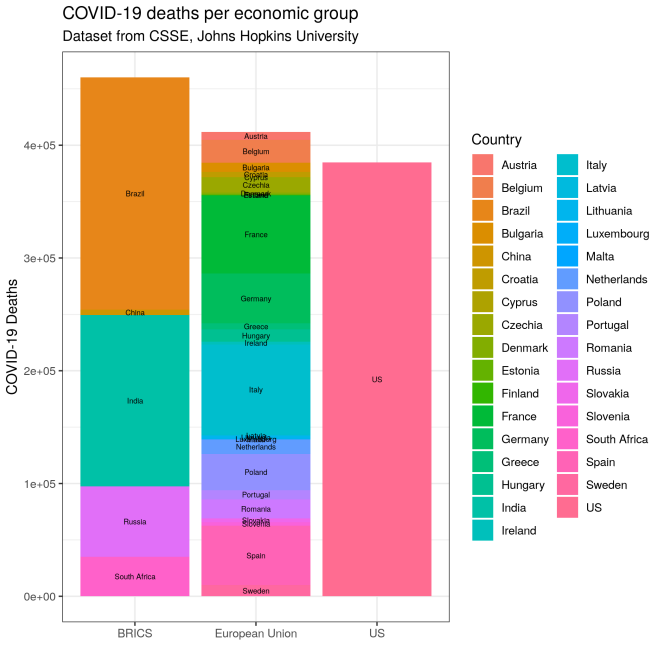

Figure 4: COVID-19 deaths per economic group

Finally, lines 46-65 sort all countries (minus the reported cruise ships in the dataset) by their ratio of total COVID-19 confirmed cases to the total number of deaths attributed to the virus, and keep the worst 25 of them. These are merged with the joint stats of all previous groups of countries (both economic and geographic) and presented in Figure 5, each annotated with their actual number of deaths and confirmed COVID-19 cases.

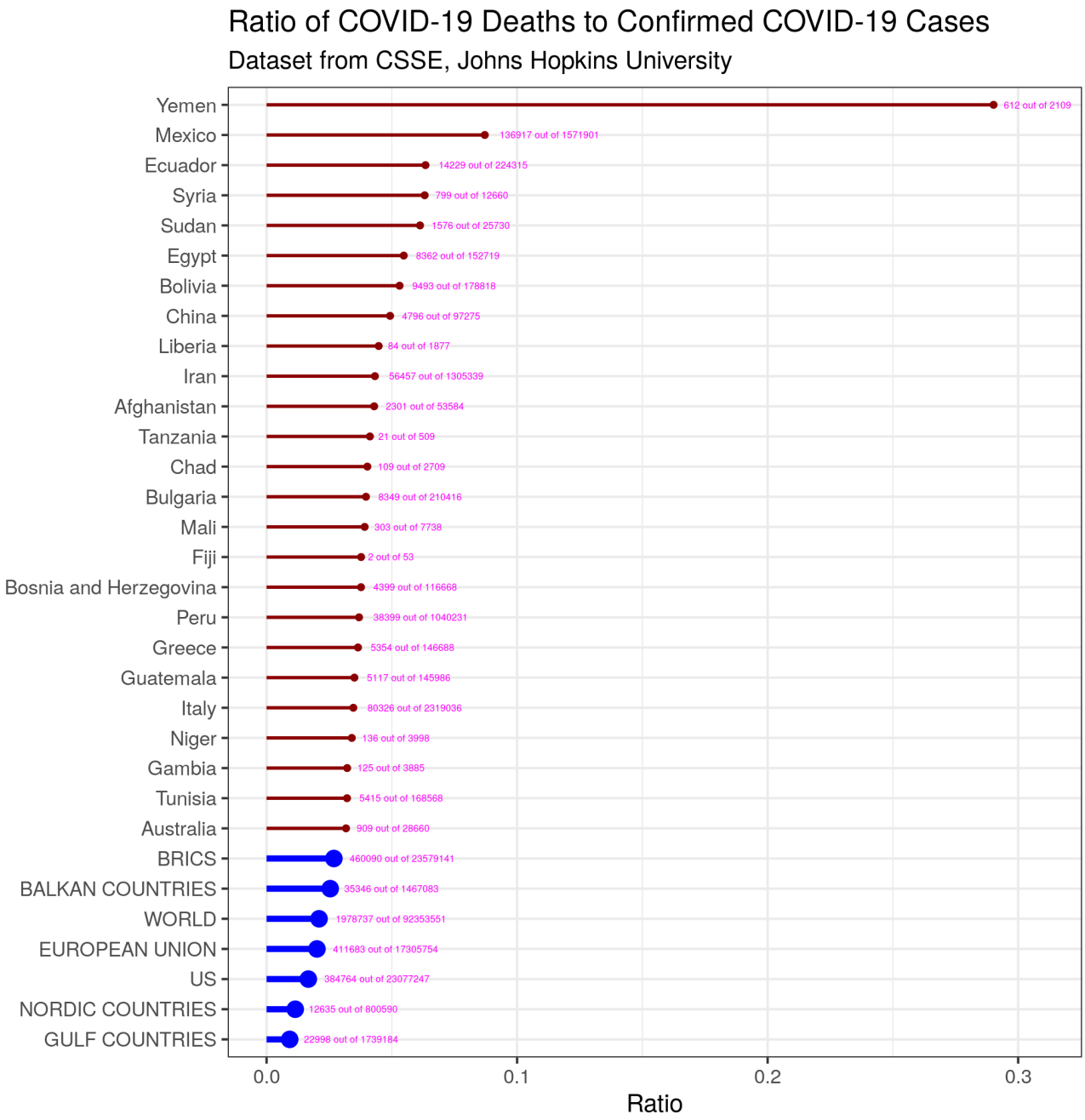

Figure 5: Ratio of COVID-19 Deaths to Confirmed COVID-19 Cases.

As it is made obvious by the diagram, countries that are either poor or isolated tend to have a higher ratio of deaths attributed to COVID-19 to confirmed cases of the virus, albeit larger countries, like Australia and Italy also appear in the bottom 25 of them. Sadly, Greece also appears in this list – as a matter of fact it appears to be the third worst European country and the second worst within the European Union (after Bulgaria).

4 Seasons

A quite common question related to COVID-19 is whether the seasons might affect the number of confirmed cases and the number of deaths all around the globe. The dataset that we have in our disposal could actually allow me to answer this question, as long as we are capable of actually determining the season for each of the countries observed.

4.1 Determining the Hemispheres

To roughly determine the season of each country, we take advantage of one of the variables that we had earlier removed from our dataset, the Latitude, which should be enough to compare the velocity of the spread of the virus across countries with different seasons at a given time. The function `determine_hemispheres()`, which is shown below, given a `data.table` as returned by the `processing()` function that was explained earlier, returns it enriched with a new variable: `Hemisphere`. Each observation can have three values for the `Hemisphere` variable: `Northern`, `Southern` or `Equator`, indicating whether an observation refers to a country located in the northern or southern hemisphere, or ± 10 degrees from Earth's equator, respectively.

Note that in this case, where we know that our dataset is very small, we have developed a standalone function that performs the retrieval and processing of the `data.table` from scratch. Had the dataset been bigger or our computing (or networking) capabilities restricted, we would probably refrain from doing so. Instead, we would modify the initial data processing procedure to include this calculation as well.

```
1  determine_hemispheres <- function(dt) {
2    lats <- fread(DEATHS_URL, header = TRUE,
3                  select = c("Country/Region", "Lat")
4      )[Lat != 0
5       ][, .(lat = mean(Lat)), by = c("Country/Region")
6         ][, Hemisphere := as.factor(ifelse(lat > 10,
7                                            "Northern",
8                                            ifelse(lat < -10,
9                                                   "Southern",
10                                                  "Equator")))
11        ][, lat := NULL]
12   setnames(lats, "Country/Region", "Country")
13
14   merge(dt, lats, by = c("Country"))
15 }
```

In the beginning (lines 2-3), `determine_hemispheres()` retrieves the columns of interest from one of the two CSV files of the dataset (the smaller between them, but this is not important) anew and filters out (line 4) any invalid observations (i.e., a couple of `N/A` and the two cruise ships included in the dataset). It then calculates a "mean" latitude for each country (by grouping by them (line 5) – recall that the original CSV includes multiple observations per country per date, according to the availability of data for each country's regions), and moves on to classify them based this value as `Northern`, `Southern` or `Equator` countries (lines 6-10). Finally, it cleans the `data.table` from the temporary variable used for this classification (line 11), and merges the

new `data.table` into the given one (line 14), after properly renaming their common variable to `"Country"` (line 12).

4.2 Visualization

To visualize the data, a standalone function has been developed: `plot_seasons()`. Given a `data.table` as produced by `determine_hemispheres`, after performing some additional calculations, it plots the time series data for the daily confirmed cases and the daily deaths per hemisphere, and stores the results to the local filesystem. It is presented in the listing below and subsequently it is explained.

```
1  plot_seasons <- function(dt) {
2    hem_series <- dt[,
3                     .(confirmed.ind = sum(confirmed.ind),
4                       deaths.inc    = sum(deaths.inc)),
5                     by = .(date, Hemisphere)
6                    ][,
7                      ":="(confirmed = cumsum(confirmed.ind),
8                           deaths    = cumsum(deaths.inc)),
9                      by = .(Hemisphere)]
10   hem_sum <- hem_series[,
11                         .(confirmed = sum(confirmed.ind),
12                           deaths    = sum(deaths.inc)),
13                        by = .(Hemisphere)]
14
15   first_date <- hem_series[date == min(date), date][1]
16   last_date <- hem_series[date == max(date), date][1]
17   ggplot(data = hem_series) +
18     aes(x = date, y = confirmed.ind, color = Hemisphere) +
19     geom_line(size = .5) +
20     geom_smooth() +
21     aes(xmin = first_date, xmax = last_date) +
22     scale_x_date(date_labels = "%b %Y",
23                  limit = c(as.Date("2020-01-21"),as.Date("2021-01-19")),
24                  expand = c(0, 0)) +
25     scale_y_log10() +
26     labs(title = "Daily confirmed COVID-19 cases") +
27     labs(subtitle = "Dataset from CSSE, Johns Hopkins University") +
28     labs(x = "", y = "") +
29     theme_bw() +
30   ggsave("hem_series_daily_cases.png")
31   ggplot(data = hem_series) +
32     aes(x = date, y = deaths.inc, color = Hemisphere) +
33     geom_line(size = .5) +
34     geom_smooth() +
35     aes(xmin = first_date, xmax = last_date) +
36     scale_x_date(date_labels = "%b %Y",
```

```
37                  limit = c(as.Date("2020-01-21"),as.Date("2021-01-19")),
38                  expand = c(0, 0)) +
39     labs(title = "Daily deaths due to COVID-19") +
40     labs(subtitle = "Dataset from CSSE, Johns Hopkins University") +
41     labs(x = "", y = "") +
42     theme_bw() +
43   ggsave("hem_series_daily_deaths.png")
```

The function starts with the creation of a `data.table` and a few calculations on it (via chaining). A new `data.table` is created (`hem_series`) (lines 2-6), which contains the daily confirmed cases and the daily number of deaths per `Hemisphere`. Chained is the calculation of the respective cumulative variables (lines 6-9), i.e., the cumulative number of confirmed COVID-19 cases as well as the cumulative number of deaths because of the virus. Last, `hem_sum` (lines 10-13) stores aggregate statistics calculated for each Hemisphere. The respective plots for the cumulative variables and the aggregate statistics are not presented here, to adhere to the assignment's strict requirement to be at most 15 pages long.

Subsequently, using `ggplot2` again (lines 17-30 and 31-43), we plot the daily stats. The results are presented in Figures 6 and 7.

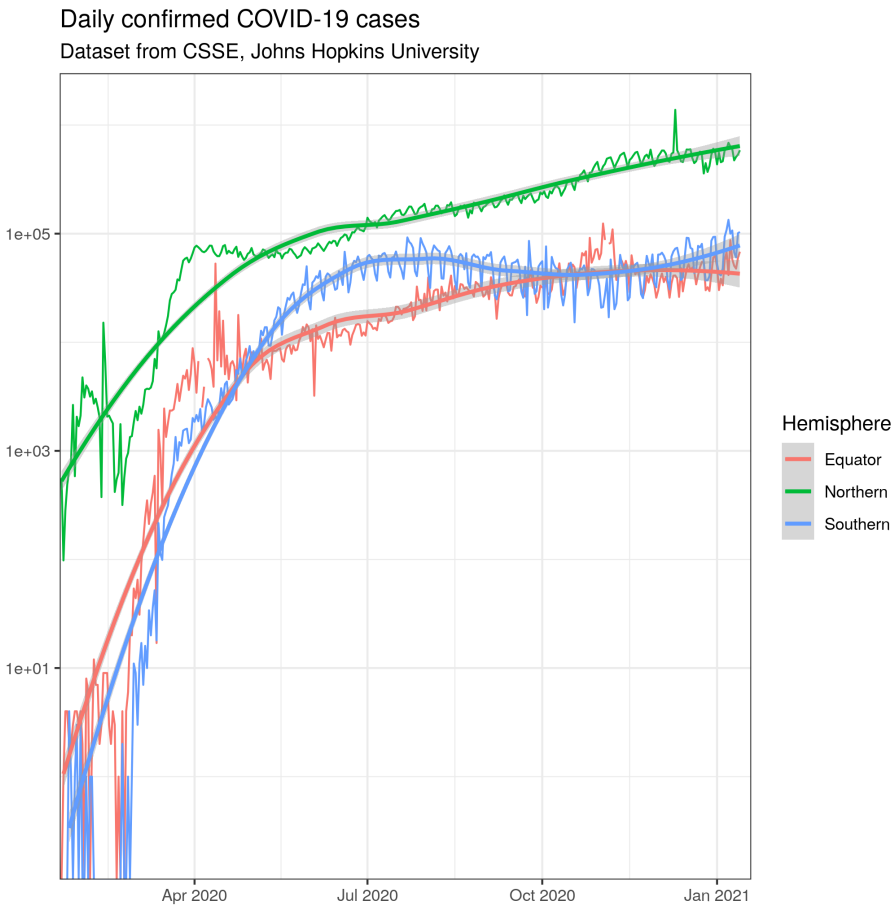

Figure 6: Daily confirmed COVID-19 cases per hemisphere by season (in logarithmic scale).

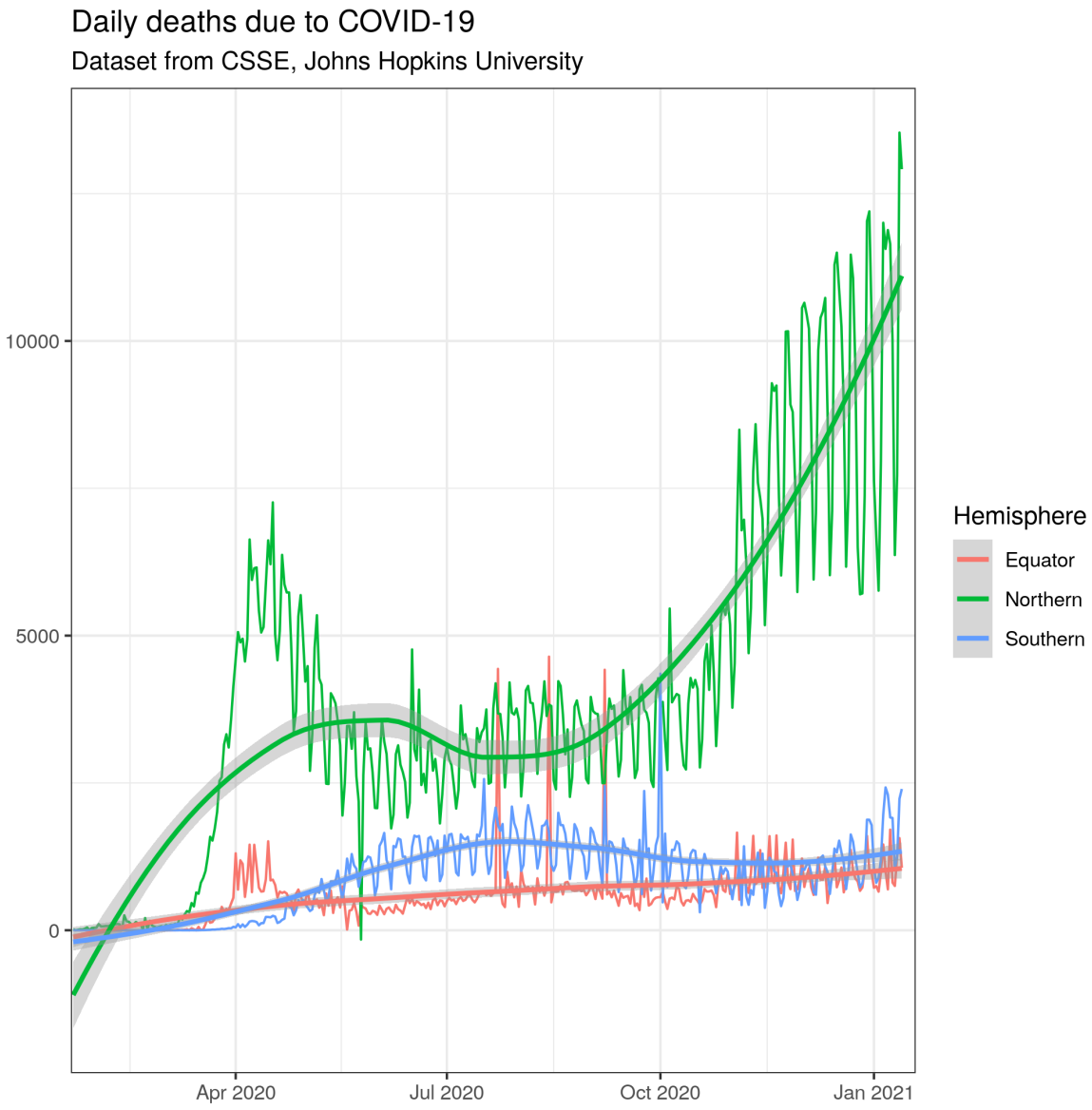

Figure 7: Daily COVID-19 deaths per hemisphere by season (in linear scale).

4.3 Discoveries

Judging from the figures above, we can verify our hypothesis that seasons affect the spread of the virus, but only up to a certain point.

As we can see in both diagrams, on June, July and August, both the confirmed COVID-19 cases and the daily number of deaths are increased for the southern hemisphere and either increased or at least stabilized for the northern hemisphere. During these months countries of the southern hemisphere go through winter, whereas countries of the northern hemisphere go through spring. On the other hand, later within the year things are getting worse for the countries of the northern hemisphere (as the winter approaches), while they seem to get better for countries of the southern hemisphere (where summer is getting closer and closer).

Countries on the equator appear to be only moderately affected by seasons in comparison with the other two groups. Their plots indicates that they follow the general trend of the global spread

of the virus more strictly than the other two groups.

4.3.1 A Note on the Recent Spike

At this point, it is worthwhile to comment on the obvious spike observed at the plot of confirmed cases for the northern hemisphere, which, had the plot been in linear scale, would definitely dominate the reader's attention with respect to the figure. Judging solely by the figure, our first intuition was that it concerns some major social event that might have occurred a few days earlier and vastly increased the spread of the virus for a few days. For instance, United States' Thanksgiving day could be one such social event, which takes place in late November.

Leveraging the power of R, we moved on to verify the hypothesis. What follows is a sequence of R commands run in interactive mode that indicate the rationale of our attempts to find out a reasonable explanation for this spike. If run in this order, their results back the conclusions that immediately follow them.

```
> source("covid19-eda.R")
> tmp <- determine_hemispheres(processing()$dt)
> # Locate the incident:
> tmp[Hemisphere == 'Northern'][confirmed.ind == max(confirmed.ind)]
> # Explore the days before and after the incident:
> tmp[Hemisphere == 'Northern'][date == '2020-12-10'][order(-confirmed.ind)]
> tmp[Hemisphere == 'Northern'][date > '2020-12-1'][
+     date < '2020-12-20'][order(-confirmed.ind)][1:20]
> tmp[Country == 'Turkey'][date > '2020-12-1'][date < '2020-12-20']
> tmp[Hemisphere == 'Northern'][order(-confirmed.ind)][1:20]
```

Even though for 18 out of the 20 days checked the US was indeed topping the daily confirmed cases stats (possibly explained by the preceding Thanksgiving), it turns out that the huge spike was caused by Turkey (with India being the country filling the last spot of the top 20 of this period).

After verifying the correctness of our own calculations, and to actively support the openness of data which enables information flows and improves the knowledge overall, our first thought was to contact the team catering the dataset to address the issue. To our great surprise (and relief), it turns out that this issue is well-known, and as a matter of fact exists as a standalone announcement in the form of a dedicated GitHub Issue[6] at the repository at hand. In brief, during that period Turkey had recently changed the criteria for qualifying a patient as a COVID-19 case. To rectify their statistics thus far, Turkey reported *all* of its past confirmed cases (according to the new criteria) as new data within a single day (823225 confirmed cases on December 10th, 2020 alone), hence causing the big spike across the whole Hemisphere's data.

References

[1] "GitHub Repository `CSSEGISandData/COVID-19`." https://github.com/CSSEGISandData/COVID-19. last accessed on Thursday 14th January, 2021.

[2] "Official Docker images for the R language." https://hub.docker.com/_/r-base. last accessed on Thursday 14th January, 2021.

[3] "Vim – the ubiquitous text editor." https://www.vim.org/. last accessed on Thursday 14th January, 2021.

[4] "Balkans, Wikipedia." https://en.wikipedia.org/wiki/Balkans#Balkan_Peninsula. last accessed on Thursday 14th January, 2021.

[5] "Arab states of the Persian Gulf, Wikipedia." https://en.wikipedia.org/wiki/Arab_states_of_the_Persian_Gulf. last accessed on Thursday 14th January, 2021.

[6] "GitHub Repository CSSEGISandData/COVID-19, Issue 3484." https://github.com/CSSEGISandData/COVID-19/issues/3484, 2020. last accessed on Thursday 14th January, 2021.