COVID19 Data Analysis

Bojan Jovanović

30 january 2024

Prerequisites

To be able to successfully complete our analysis, we have to include some R libraries we will need in the process. Those include **tinytex, tidyverse, dplyr and ggplot2**.

```
library("tinytex")
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library("dplyr")
library("ggplot2")
```

Importing COVID 19 Data

Before we continue with our analysis, we have to import the data first. In this report, we will be analyzing "*COVID 19 data*", which is provided to us by John Hopkins University. The dataset includes **data about covid cases and deaths in US and worldwide** from 22nd of January 2020, up until the 9th march of 2023. The dataset is not updated with new data anymore.

More info about this dataset can be found at this link.

The dataset includes information about **Country/State, latitude and longitude**, as well as the **cumulative number of cases/deaths** for every day from January 22nd 2020 to 9th of March 2023. Every row in data corresponds to a single geographical area, while there is a single column for each date.

```
## NOTE: IF ERROR HAPPENS IN THIS CHUNK ON FIRST RUN, RUN IT ONCE AGAIN!
## FOR SOME WEIRD REASON, DATA IMPORT CAN FAIL ON FIRST RUN
## HOWEVER, IT SUCCED ON SUBSEQUENT ATTEMPTS
us_cases_loc <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/css
us_deaths_loc <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/c
```

```
global_cases_loc <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data
global_deaths_loc <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_dat

us_cases_data <- read_csv(us_cases_loc)
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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.
```

```
us_deaths_data <- read_csv(us_deaths_loc)
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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.
```

```
global_cases_data <- read_csv(global_cases_loc)
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
global_deaths_data <- read_csv(global_deaths_loc)
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

Tidying Our Data

Filtering out Columns we Won’t Need

For our further analysis, we will need country level data. Thus we will select *Country/State*, *Latitude* and *Longitude* columns. As we will be working with total cases/deaths for each country, we will select *only the* *latest date column*, which shows the total number of recorded cases/deaths.

Renaming Columns

There is a **slight difference in the naming of columns** between global and US datasets. We will fix this discrepancy by renaming columns. This will prove very beneficial when we get to joining our data. Also, we will rename the final date column into cases/deaths, to better showcase what its value mean.

Aggregating Data

In those datasets, most of the countries have data reported at country level. However, larger countries like United States, China or Australia, have their data reported at State/Province levels. We will aggregate this data, so **all values reported are at the Country level**.

Calculating Death Rates

In our analysis, we will be working with death rates. To calculate death rates, we have to join cases and deaths datasets. Then, we will calculate the death rate by **dividing the total number of deaths by the total number of cases**.

Removing Nonsensical Data

After performing aforementioned steps, we have compiled a dataset appropriate for our following analysis. However, by further checking the values in it, we have found some nonsensical data. For example, Antartica has the **death rate of zero** (and just 11 cases), just like records named **Summer Olympics 2020 and Winter Olympics 2022**. Also, North Korea has the **death rate of 600%**, with one case and six deaths.

Clearly, such data is not correct. So, **we will remove all the records with the death rate of zero and over 100%**.

Also, we will remove records with **latitude and longitude of zero**. Such records are not corresponding to any countries (One such record is for Diamond Princess cruise ship).

```r
## FILTER OUT COLUMNS
us_cases_filtered <- us_cases_data %>% select(Country_Region, Lat, Long_, "3/9/23")
us_deaths_filtered <- us_deaths_data %>% select(Country_Region, Lat, Long_, "3/9/23")
global_cases_filtered <- global_cases_data %>% select("Country/Region", Lat, Long, "3/9/23")
global_deaths_filtered <- global_deaths_data %>% select("Country/Region", Lat, Long, "3/9/23")

## RENAME COLUMNS
us_cases_filtered <- us_cases_filtered %>% rename("Country/Region" = "Country_Region",
                                                  "Long" = "Long_", "Cases" = "3/9/23")
us_deaths_filtered <- us_deaths_filtered %>% rename("Country/Region" = "Country_Region",
                                                  "Long" = "Long_", "Deaths" = "3/9/23")
global_cases_filtered <- global_cases_filtered %>% rename("Cases" = "3/9/23")
global_deaths_filtered <- global_deaths_filtered %>% rename("Deaths" = "3/9/23")

## AGGREGATE DATA
us_cases_aggregated <- us_cases_filtered %>% group_by(`Country/Region`) %>%
                      summarise(Lat = mean(Lat, na.rm = TRUE),
                                Long = mean(Long, na.rm = TRUE),
                                Cases = sum(Cases))
```

```
us_deaths_aggregated <- us_deaths_filtered %>% group_by(`Country/Region`) %>%
                      summarise(Lat = mean(Lat, na.rm = TRUE),
                                Long = mean(Long, na.rm = TRUE),
                                Deaths = sum(Deaths))
global_cases_aggregated <- global_cases_filtered %>% group_by(`Country/Region`) %>%
                      summarise(Lat = mean(Lat, na.rm = TRUE),
                                Long = mean(Long, na.rm = TRUE),
                                Cases = sum(Cases))
global_deaths_aggregated <- global_deaths_filtered %>% group_by(`Country/Region`) %>%
                      summarise(Lat = mean(Lat, na.rm = TRUE),
                                Long = mean(Long, na.rm = TRUE),
                                Deaths = sum(Deaths))

## JOIN DATA AND CALCULATE DEATH RATE
us_joined <- us_cases_aggregated %>%
              inner_join(us_deaths_aggregated) %>%
              mutate("Death Rate" = Deaths/Cases)
```

```
## Joining with `by = join_by(`Country/Region`, Lat, Long)`
```

```
global_joined <- global_cases_aggregated %>%
              inner_join(global_deaths_aggregated) %>%
              mutate("Death Rate" = Deaths/Cases)
```

```
## Joining with `by = join_by(`Country/Region`, Lat, Long)`
```

```
data_tidy <- bind_rows(us_joined, global_joined)

## REMOVE NONSENSICAL DATA
data_tidy <- data_tidy %>% filter((Lat != 0 | Long != 0) &
                              `Death Rate` > 0 & `Death Rate` < 1)

summary(data_tidy)
```

```
##  Country/Region          Lat              Long             Cases        
##  Length:194         Min.   :-38.416   Min.   :-175.20   Min.   :     5014  
##  Class :character   1st Qu.:  4.292   1st Qu.: -11.57   1st Qu.:    49402  
##  Mode  :character   Median : 17.125   Median :  19.93   Median :   344994  
##                     Mean   : 18.481   Mean   :  16.88   Mean   :  4022515  
##                     3rd Qu.: 39.047   3rd Qu.:  46.70   3rd Qu.:  1985288  
##                     Max.   : 64.963   Max.   : 178.06   Max.   :103802702  
##      Deaths             Death Rate       
##  Min.   :      1.0   Min.   :0.0001906  
##  1st Qu.:    601.8   1st Qu.:0.0058394  
##  Median :   3843.5   Median :0.0108267  
##  Mean   :  41266.1   Mean   :0.0142321  
##  3rd Qu.:  19802.0   3rd Qu.:0.0191756  
##  Max.   :1123836.0   Max.   :0.1807451
```

Data Analysis

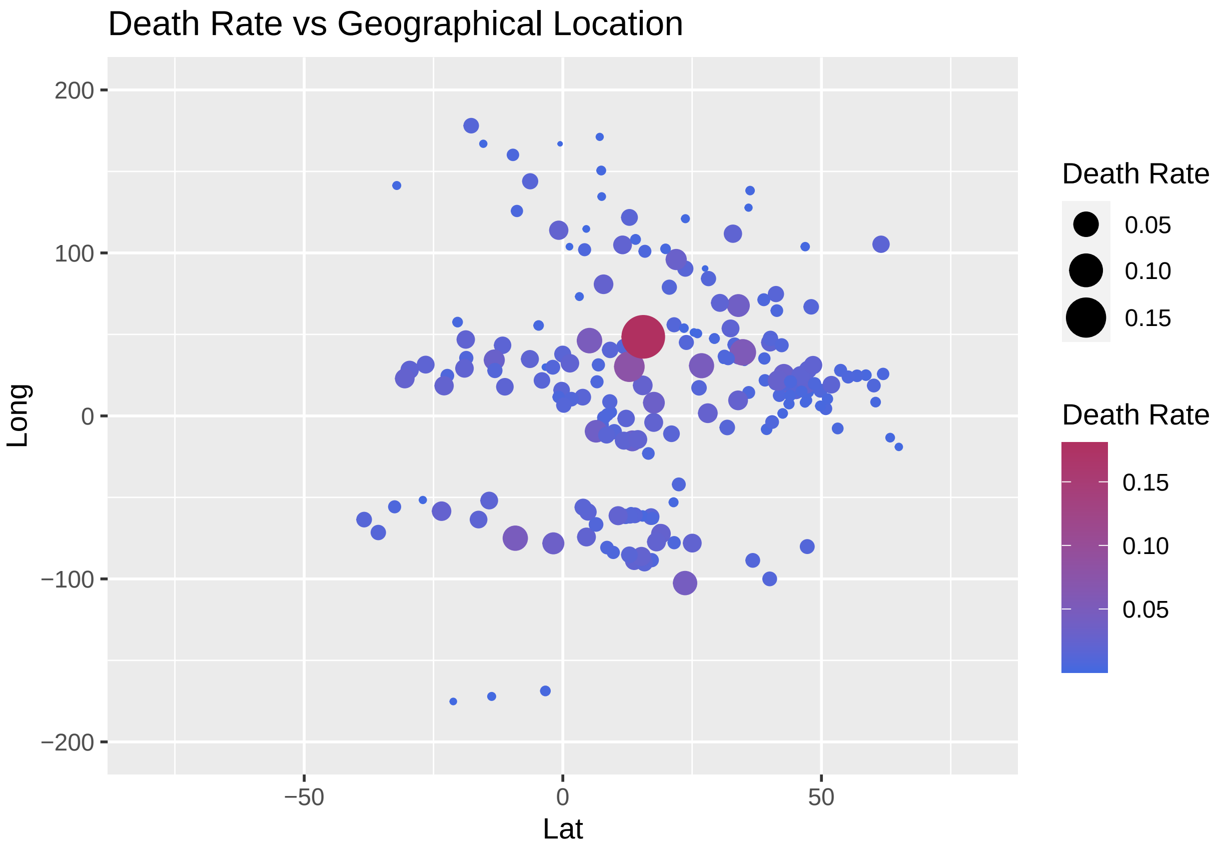
In this part of our report, we will analyze our prepared data. Our main research question will be **whether geographical location (Latitude/Longitude) of a country affects the death rate from COVID**

19 infection.

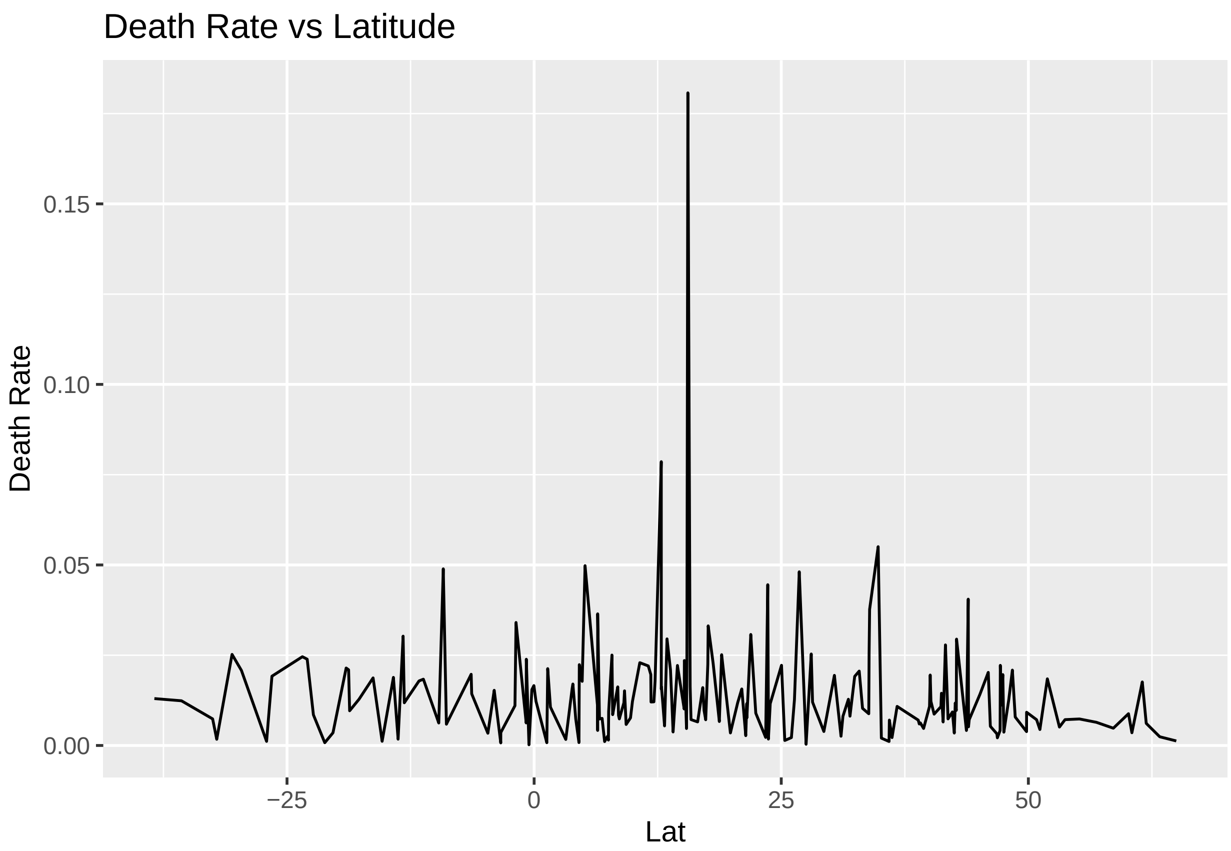
Before everything else, we will try to roughly estimate whether there is some effect on COVID 19 deadlieness due to the geographical location. To do so, we will create **a scatterplot with markers of different sizes**, correlating latitude and longitude to the death rate.

We will also create **line plots**, comparing the death rate against latitude and longitude independently. Hopefully, we will detect some trends there.

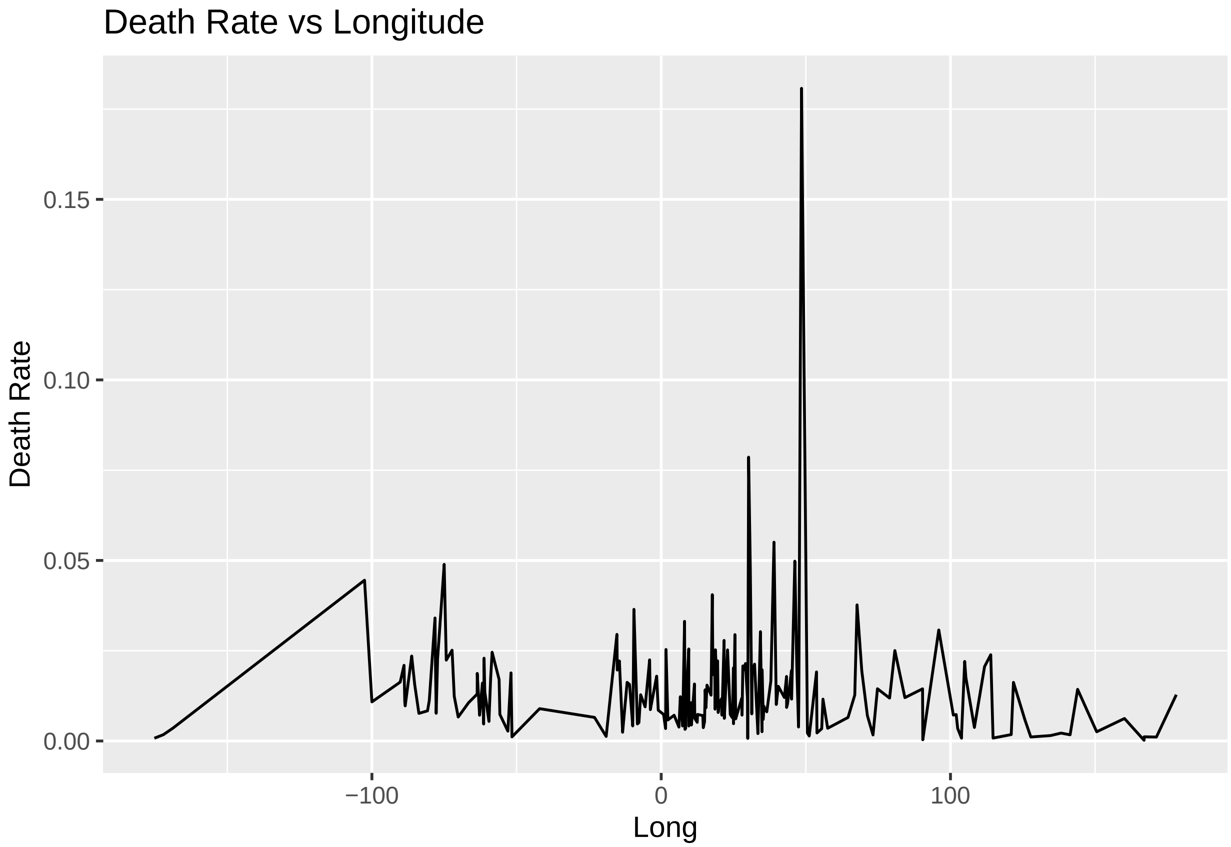
```
## PLOT DEATH RATE AGAINT LATITUDE/LONGITUDE
ggplot(data=data_tidy) +
  geom_point(aes(x=Lat, y=Long, size=`Death Rate`, color=`Death Rate`)) +
  scale_size_continuous(range = c(0.3, 7)) +
  scale_color_gradient(low="royalblue", high="maroon") +
  scale_x_continuous(limits=c(-80,80)) +
  scale_y_continuous(limits=c(-200,200)) +
  ggtitle("Death Rate vs Geographical Location")
```



```
## PLOT DEATH RATE AGAINST LATITUDE
ggplot(data=data_tidy) +
  geom_line(aes(x=Lat, y=`Death Rate`)) +
  ggtitle("Death Rate vs Latitude")
```



```
## PLOT DEATH RATE AGAINS LONGITUDE
ggplot(data=data_tidy) +
  geom_line(aes(x=Long, y=`Death Rate`)) +
  ggtitle("Death Rate vs Longitude")
```



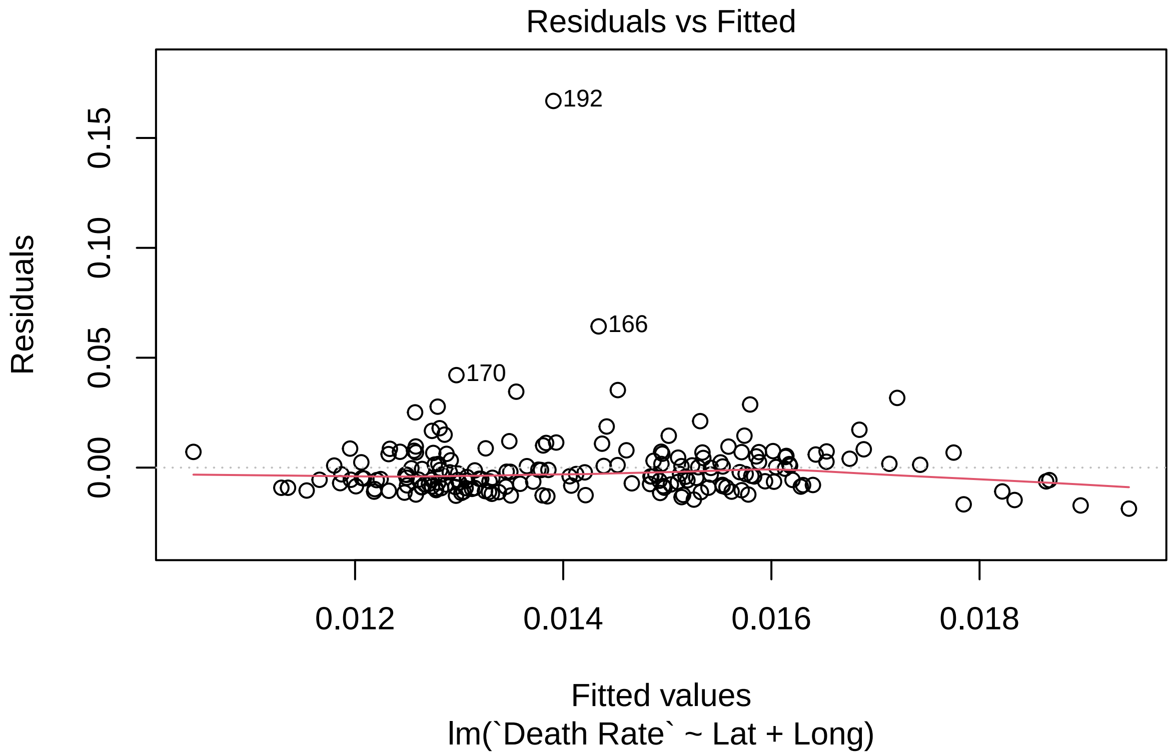
Analyzing out Plots

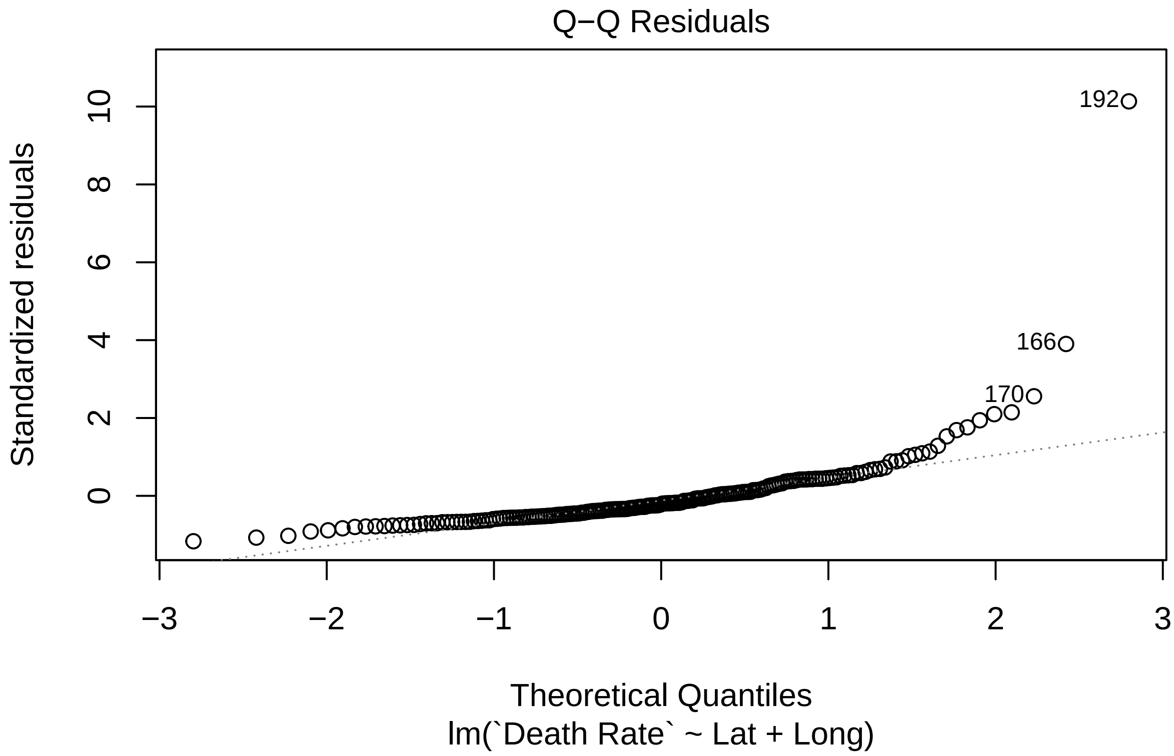
From the plots we have created, there is **no clear trend between the latitude/longitude and death rate**. However, we do see few observations with greater death rate being grouped around longitude of 40 degrees.

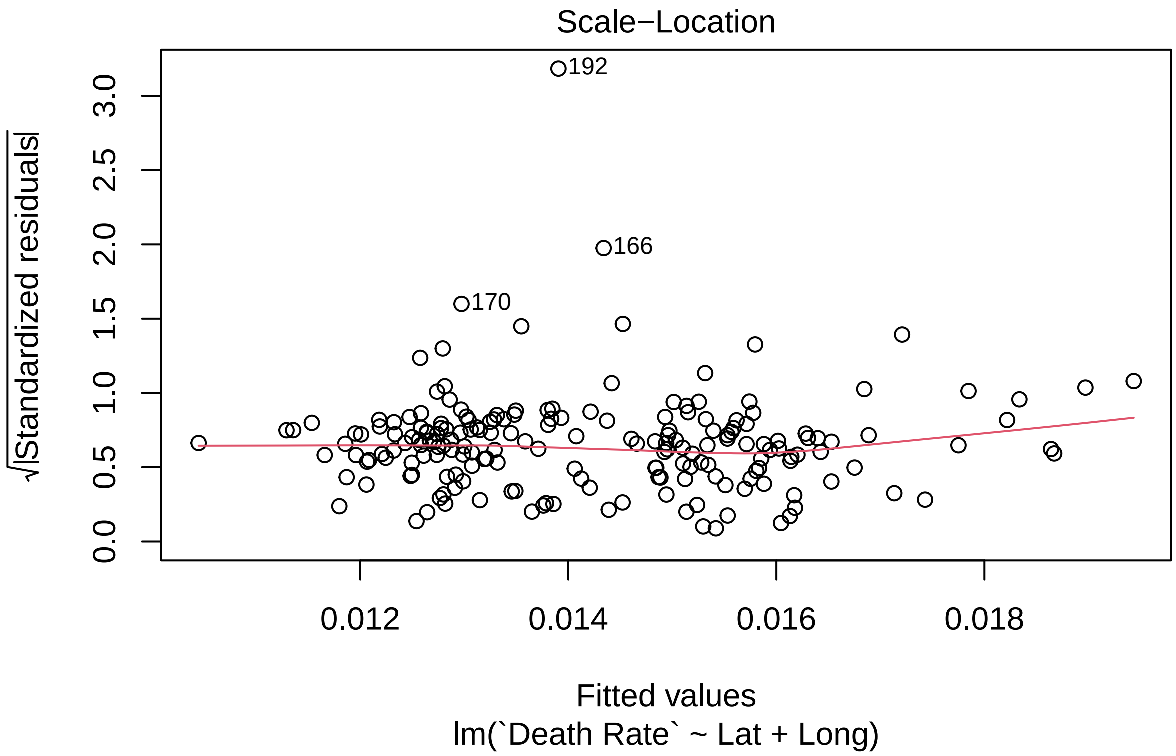
To analyze our observations further, we will fit **a standard linear regression model** on our data, hoping it will reveal something we did miss from our plots. We will try to predict the outcome of Death Rate by predictors Latitude and Longitude.

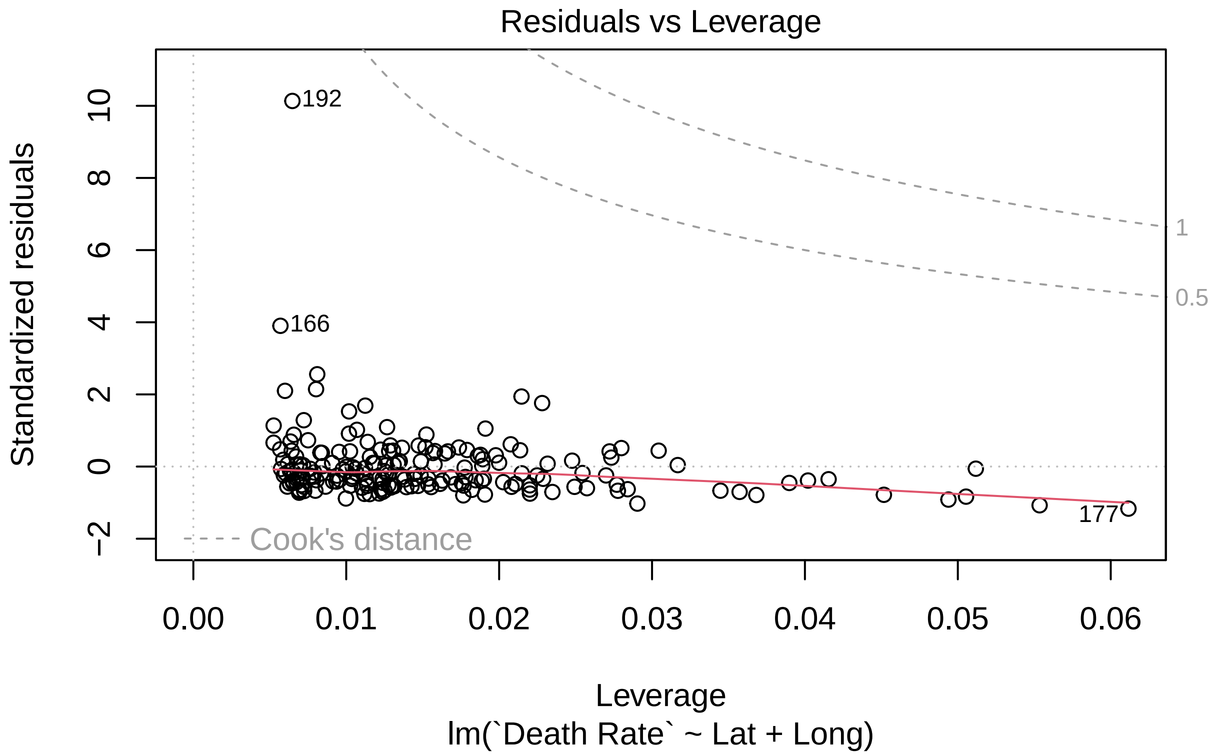
```r
## FITTING LOGISTIC REGRESSION
model <- lm(formula=`Death Rate`~Lat+Long, data=data_tidy)

## DIAGNOSTIC PLOTS
plot(model)
```


Residuals vs Fitted
Residuals
0.15
0.10
0.05
0.00
192
166
170
0.012
0.014
0.016
0.018
Fitted values
lm(`Death Rate` ~ Lat + Long)


Q−Q Residuals
Standardized residuals
0
2
4
6
8
10
−3
−2
−1
0
1
2
3
192
166
170
Theoretical Quantiles
lm(`Death Rate` ~ Lat + Long)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
192
166
170
0.012
0.014
0.016
0.018
Fitted values
lm(`Death Rate` ~ Lat + Long)



```
## MODEL SUMMARY
summary(model)
```

```
## 
## Call:
## lm(formula = ‘Death Rate‘ ~ Lat + Long, data = data_tidy)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.018662 -0.008424 -0.003505  0.004497  0.166840 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.553e-02  1.530e-03  10.147   <2e-16 ***
## Lat         -5.607e-05  5.071e-05  -1.106    0.270    
## Long        -1.551e-05  1.811e-05  -0.857    0.393    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.01652 on 191 degrees of freedom
## Multiple R-squared:  0.01078,    Adjusted R-squared:  0.0004221 
## F-statistic: 1.041 on 2 and 191 DF,  p-value: 0.3552
```

We fitted our model as described previously. However, after we fit any model, we should **diagnose how well it fitted our data**. We have ploted default R diagnostics plots and analyzed them.

By looking at Q-Q residuals plot and Residuals vs Fitted, we se that residuals roughly follow a linear trend, Normal distribution and doesn't deviate much from the constant variance, with few outliers acros the range of observations. Thus, we will conclude that **linear regression assumptions are sufficiently satisfied** and we will continue with model interpretation.

Conslussion of the Analysis

By looking at the summary of the model, we see coefficients for latitude and longitude are not statistically significant. Thus, we did not find any evidence for our research question.

By taking our plots into account and the model we have fitted, it seems like **there is no relation between geografical position and death rate** from ĆOVID 19.

Bias Identification

Data Reporting Bias

The most obvious bias for me is the **bias related to data reporting**. We know that different countries used different methodologies for reporting COVID 19 deaths, which could possibly skew our data.

We have to admit that there is a possibility of **malicious data reporting**. For example, China had been acussed by many countries of hiding cases and deaths, so news of novel Coronavirus don't negativelly affect their trade business. Austria was also accused of hiding data, to preserve their skiing season in Alps.

There are many other countries accused of **hiding true death rate**. We know that might be true by looking at the reports of **excess deaths** in those countries. For example, excess death analysis of data in my homecountry indicates possibility of up to 3 times more deaths from COVID 19 than what was reported.

Undesigned Experiment Bias

When performing an research, the best way to go about it is to properly design a study before going out to collect the exact data we need. However, this wasn't the case in this report.

We were given this dataset and we were forced to do our best with what we have. Thus, **it is possible we missed many variables which could have affected the death rate**. For example, we don't have information about wealth of a certain country in our datasets.

Wealth could certainly affect the death rate of people in a certain country, by providing its residents with better nutrition, better healthcare, work in hygienic conditions and other stuff affecting their overall well-being, as well as their likelihood of catching a virus.

Personal Biases

We all see our world through our biased minds. Thus, there is always a possibility of those **personal biasses affecting our analysis**.

For example, many stories about Vitamin D efficacy against COVID 19 have surfaced during the pandemic. Personaly, I did believe this hypothesis about Vitamin D.

Vitamin D levels depend on whether you live in sunny region or not, so there is a possibility that my belief about Vitamin D unconsciously affected my hypothesis of geographical location affecting the death rate instead of reasonable thinking.

To rectify my bias, **I try not to make any claims without finding decent statistical proof** for what I am about to claim. Hopefully, this reduces the chance of invalidating my research due to my biases.

Session Info

To make this report reproducible, the configuration used will be shown below. This list includes R version, Operating system, locale and packages in use.

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=Serbian (Latin)_Serbia.utf8
## [2] LC_CTYPE=Serbian (Latin)_Serbia.utf8
## [3] LC_MONETARY=Serbian (Latin)_Serbia.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=Serbian (Latin)_Serbia.utf8
##
## time zone: Europe/Belgrade
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.3 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
##  [5] purrr_1.0.2     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1
##  [9] ggplot2_3.4.4   tidyverse_2.0.0 tinytex_0.49
##
## loaded via a namespace (and not attached):
##  [1] bit_4.0.5         gtable_0.3.4      highr_0.10        crayon_1.5.2
##  [5] compiler_4.3.2    tidyselect_1.2.0  parallel_4.3.2    scales_1.3.0
##  [9] yaml_2.3.8        fastmap_1.1.1     R6_2.5.1          labeling_0.4.3
## [13] generics_0.1.3    curl_5.2.0        knitr_1.45        munsell_0.5.0
## [17] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.3       utf8_1.2.4
## [21] stringi_1.8.3     xfun_0.41         bit64_4.0.5       timechange_0.3.0
## [25] cli_3.6.2         withr_3.0.0       magrittr_2.0.3    digest_0.6.34
## [29] grid_4.3.2        vroom_1.6.5       rstudioapi_0.15.0 hms_1.1.3
## [33] lifecycle_1.0.4   vctrs_0.6.5       evaluate_0.23     glue_1.7.0
## [37] farver_2.1.1      fansi_1.0.6       colorspace_2.1-0  rmarkdown_2.25
## [41] tools_4.3.2       pkgconfig_2.0.3   htmltools_0.5.7
```