Covid Analysis

Dor Kanfer

25/5/2021

Data Wrangling and Plotting

Submission Instructions

This lab will be submitted in pairs using GitHub (if you don’t have a pair, please contact us).
Please follow the steps in the Git Classroom assignment to create your group’s Lab 1 repository.
Important: your team’s name must be `FamilyName1_Name1_and_FamilyName2_Name2`.
You can collaborate with your partner using the git environment; You can either make commits straight to master, or create individual branches (recommended). However, once done, be sure to merge your branches to master - you will be graded using the most recent master version - your last push and merge before the deadline.
Please do not open/review other peoples’ repositories - we will be notified by GitHub if you do.

Your final push should include this Rmd file (with your answers filled-in), together with the html file that is outputted automatically by knitr when you knit the Rmd. Anything else will be disregarded. In addition, please adhere to the following file format:
`Lab_2_FamilyName1_Name1_and_FamilyName2_Name2.Rmd/html`

Submission Deadline: 19/5/2021 at 23:59

The only allowed libraries are the following (**please do not add your own**):

Analysis of the World Covid-19 Dataset

The `world-of-data` website hosts world-wide epidemiological data on the Corona Virus (COVID-19). The dataset is compiled by the Johns Hopkins University Center for Systems Science and Engineering (JHU CCSE) from various sources, and follows The dataset contains data since January 2020. For the data and more information about it, please visit here (https://github.com/owid/covid-19-data/tree/master/public/data).

You can see several nice visualizations of the data here (https://ourworldindata.org/covid-vaccinations)

In this lab we will focus on analyzing the Covid-19 cases, deaths and vaccinations data over time for different countries.

General Guidance: - Your solution should be submitted as a full Rmd report integrating text, code, figures and tables. For each question, describe first in the text of your solution what you’re trying to do, then include the relevant code, then the results (e.g. figures/tables) and then a textual description of them.

- In most questions the extraction/manipulation of relevant parts of the data-frame can be performed using commands fron the `tidyverse` and `dplyr` R packages, such as `head`, `arrange`, `aggregate`, `group-by`, `filter`, `select`, `summaries`, `mutate` etc.
- When displaying tables, show the relevant columns and rows with meaningful names, and descirbe the results.
- When displaying figures, make sure that the figure is clear to the reader, axis ranges are appropriate, labels for the axis , title and different curves/bars are displayed clearly (font sizes are large enough), a legend is shown when needed etc. Explain and descrie in text what is shown in the figure.
- In many cases, data are missing (e.g. NA). Make sure that all your calculations (e.g. taking the maximum, average, correlation etc.) take this into account. Specifically, the calculations should ignore the missing values to allow us to compute the desired results for the rest of the values (for example, using the option `na.rm = TRUE`).

QUESTIONS

1. [3 pts] First, load the complete covid19 dataset in csv format from the world-of-data world-of-data (https://github.com/owid/covid-19-data/tree/master/public/data) into a data-frame in R.
   Change if needed the class of the `date` variable to `Date` and check that the class is correct.
2. [6 pt] List in a table the top 5 countries in terms of current `total_cases_per_million`. Show only the country, last-date, and the total number of cases per million.
   Repeat the same with two additional separate tables for top 5 countries for `total_deaths_per_million` and `total_vaccinations_per_hundred`.
3. [12 pts]

a. Write a function that recieves as input the data-frame, and a column name as string. The function plots the value of the input column as a function of the date for each of the six continents (`Africa`, `Asia`, `Europe`, `North America`, `Oceania`, `South America`), shown on the same graph with different colors or symbols. Make sure that the difference between the continents is visualized clearly, use meaningful axis and plot labels, and add an informative legend. NA or other no-number values should not be displayed.

b. Use the function written in a. and plot of the number of `new_cases` for the continents. Next, make a similar plot for the *log* of the *smoothed* number of new cases. Which plot is easier to interpret? explain. Similarly, make two additional separate plots for the *log* of the *smoothed* number of `new_deaths` and `new_vaccinations` as a function of date for the continents. Describe the plotted results.

4. [12 pts] We would like to make a similar plot to the ones in qu. 3 for the number of new tests for each continent. However, some of the variables like `new_tests` and `new_test_smoothed` are not given at the continent level, but only at the individual country level. We therefore need to *complete* them for each continent.

a. Write a function that recieves as input the data-frame and a column to complete, and computes for each continent the corresponding values. The value for a given continent and a specific date (represented in one row of the data-frame) should be a *weighted average* over the values of all countires in the corresponding continent for the same date, with weights proportional to the individual countries' `population`.
Guidance: Make sure you only update rows corresponding to entire continents (rows corresponding to individual countries should remain the same)

b. Apply the function from a. to fill the `new_tests_smoothed` column for the continents, and plot the *log* of the *smoothed* number per continent vs. date using the function from qu. 3.

5. [14 pt]

Loading [MathJax]/jax/output/HTML-CSS/jax.js

a. Create a new data-frame with one row per country, that for each countrie will store as columns the current `total_cases_per_million` and `total_deaths_per_million` , in addition to the country name (`location`).
Next, make a scatter plot showing these two columns. Compute a linear regression line of the number of deaths per million as a function of the number of cases per million and add the fitted regression line to the plot. What is the slope and what does it represent?

b. Find for each country the date at which the number of new `cases` was maximal, and the date at which the number of new `deaths` was maximal, and add them to the data-frame from a.
Make a scatter plot with a linear regression line as in a. Is the slope close to one? why? What is the intercept and what does it represent ?

6. [9 pt] We want to compute the world-wide number of `new_cases` , `new_deaths` and `new_vaccinations` by month. Aggregate the country-level data and store the results in a new dataframe called `monthly` with each row corresponding to a month, and columns correponding to the worldwide number of new cases, deaths or vaccinations in this month.
Show the three columns in three different barplots.
Guidance: (i) Beware to not double-count cases/deaths/vaccinations. (ii) Treat each month seperately (e.g. March 2020 and March 2021 are different).

7. [9 pt] Add to the covid data-frame a new column called `death_rate` , defined for `location` and `date` as the number of `total_deaths` divided by the number of `total_cases` . This column represents the probability of a diagnosed Covid-19 case to die from the disease.
Next, make a histogram of the current death rates over all countries with 50 bins.
List in a table the top 3 countries having the highest death rate.

8. [9 pt] Given that most vaccinations (specifically *Pfizer* and *Moderna*) are given in two-doses, we want to investigate whether different countries employ different vaccination strategies. While some countries vaccinate only individuals for which there are two doseas of the vaccine given at proximity in time (usually less than one month apart), other countries first use the available vaccines to vaccninate as many poeple as possible using one dose, and may delay the second dose for these individuals.
Create an additional column called `two_dose_fraction` , defined as the number of *fully vaccinated* people divided by the number of *vaccinated* people.
Next, plot for `Israel` , `United Kingdom` and `United States` this value as a function of date, on the same plot with different colors. What do you think are the vaccination strategies for the different countries based on these plots? explain.

9. [14 pt] We want to use the data in order to study the time delay between the diagnosis of Covid-19 and the death from Covid-19 for cases not surviving the disease. For two functions of time $X(t)$ and $Y(t)$ (here t is discrete, representing for example days) we define the *cross-correlation* as follows:
$cross_{corr}(\Delta_t; X, Y) = Corr(X(t), Y(t + \Delta_t))$.
That is, the cross-correlation function at the time-delay Δ_t for two vectors of length n is obtained by computing the Pearson correlation coefficient of the vector $X[1, \ldots, n - \Delta_t]$ with the vector $Y[\Delta_t + 1, \ldots, n]$, for $\Delta_t > 0$. For $\Delta_t < 0$ we replace the role of X and Y in the formula.

a. Write a function that recieves as input the data-frame, a country name and the name of two columns, and computes the value of their cross-correlation in this country for time delay of up to two months apart, that is for all values of Δ_t between -60 days and 60 days. The function should return a vector of length 121 representing these correlations.

b. Compute the cross correlation between the number of `new_cases` and `new_deaths` for *Canada*, and plot it as a function of Δ_t. At what time delay is the cross correlation maximized? what is your interpretation of this time-delay?

10. [12 pt] Finally, we want to examine if the data shows evidence for the effectiveness of the vaccines in reducing the number of Covid-19 cases. Compute the *ratio* between the *current* number of smoothed new

cases (at April 23rd, 2021), and the *maximal* number of smoothed new cases for each country.
Extract also the total number of vaccinations per hundred people for each country at April 1st, 2021. (We allow an approximately three weeks delay between the vaccinations and their effect).
Make a scatter-plot of the two with the ratio shown in *log*, i.e. comparing the vaccination rate to the *log* of the reduction in the number of current daily cases compared to its maximum.
Mark in red in the scatter-plot the points corresponding to Israel and to United Kingdom. How effective are vaccinations for these two countries based on the plot? do you see other countries where the effect of vaccination seems very different?

Solution:

Write your solutions here seperately for each question in the followng format:

SOLUTIONS:

1.

```
df = read.csv(file.choose())
df$date = as.Date(df$date)
class(df$date)
```

```
## [1] "Date"
```

we can see that the class of the date variable is “Date”.

2.

```
df$total_vac_per_100_no.na = df$total_vaccinations_per_hundred
df$total_vac_per_100_no.na[is.na(df$total_vac_per_100_no.na)] = 0

head(df%>%
       arrange(desc(df$total_cases_per_million))
     %>% filter(date == max(date))
     %>% select(location,date,total_cases_per_million),5)
```

```
##     location       date total_cases_per_million
## 1    Andorra 2021-05-04                172341.9
## 2 Montenegro 2021-05-04                155737.8
## 3    Czechia 2021-05-04                152847.0
## 4 San Marino 2021-05-04                149301.7
## 5   Slovenia 2021-05-04                116779.7
```

```
head(df%>%
       arrange(desc(df$total_deaths_per_million))
     %>% filter(date == max(date))
     %>% select(location,date,total_deaths_per_million),5)
```

```
##                  location       date total_deaths_per_million
## 1                 Hungary 2021-05-04                 2903.104
## 2                 Czechia 2021-05-04                 2747.320
## 3 Bosnia and Herzegovina 2021-05-04                 2655.743
## 4              San Marino 2021-05-04                 2651.895
## 5              Montenegro 2021-05-04                 2413.774
```

```
head(df%>%
       arrange(desc(df$total_vac_per_100_no.na))
     %>% filter(date == max(date))
     %>% select(location,date,total_vac_per_100_no.na),5)
```

```
##      location       date total_vac_per_100_no.na
## 1 Afghanistan 2021-05-04                       0
## 2      Africa 2021-05-04                       0
## 3     Albania 2021-05-04                       0
## 4     Algeria 2021-05-04                       0
## 5     Andorra 2021-05-04                       0
```

[MY SOLUTION TEXT - DESCRIPTION OF RESULTS]

3.

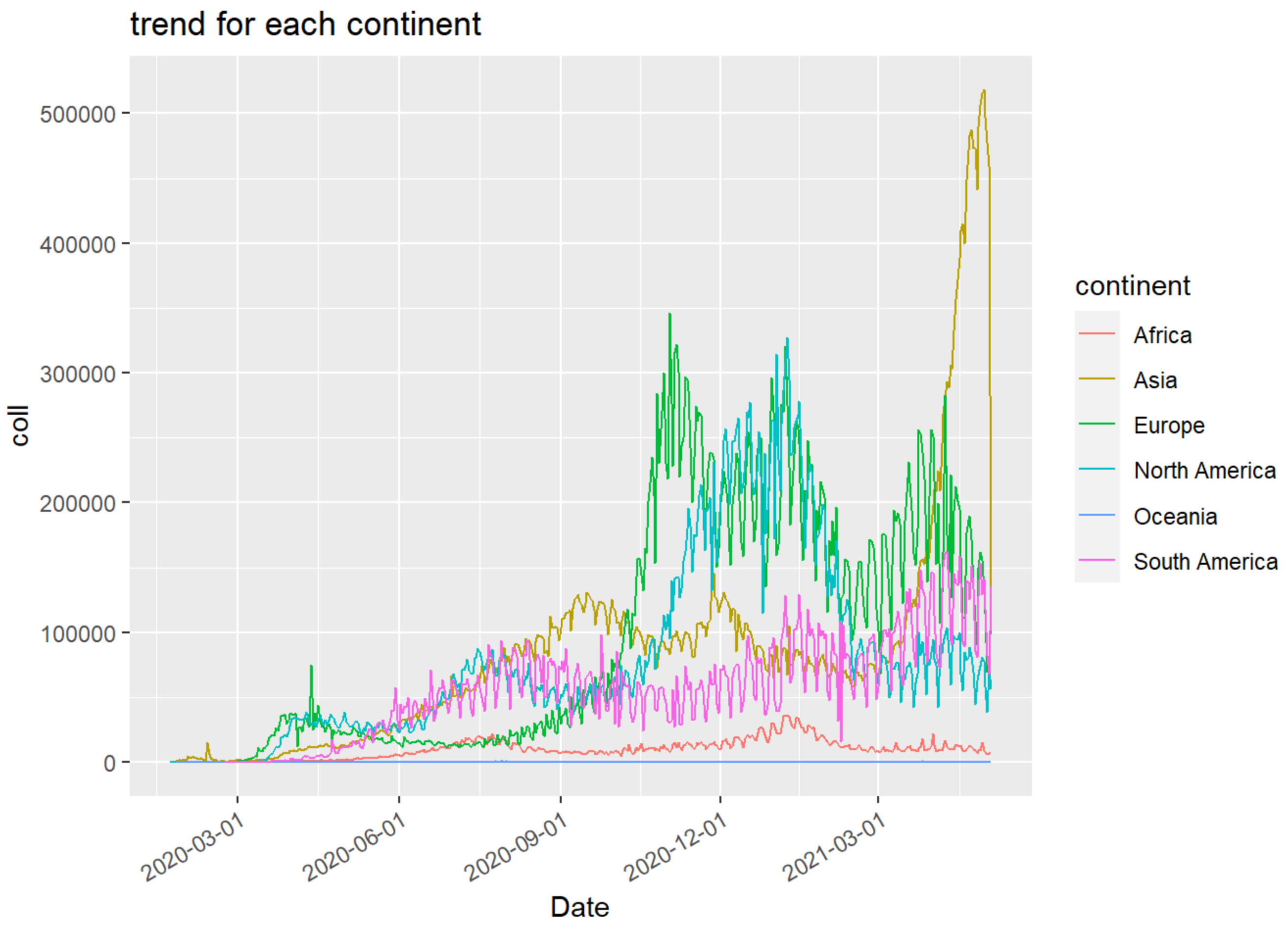
Write a function that receives as input the data-frame, and a column name as string. The function plots the value of the input column as a function of the date for each of the six continents.

a

```
df_clean = df[!(df$continent==""),]

f1 = function(data,coll) {
  options(scipen = 999)
  agg = aggregate(coll~continent+date, data, sum)
  ggplot(agg, aes(x = as.Date(date), y = coll, color = continent))+
    geom_line() + labs(y = "coll",x = "Date")+ scale_x_date(date_breaks = "3 months")+
    ggtitle("trend for each continent") + theme(axis.text.x = element_text(angle=30, hjust=1))
}
```

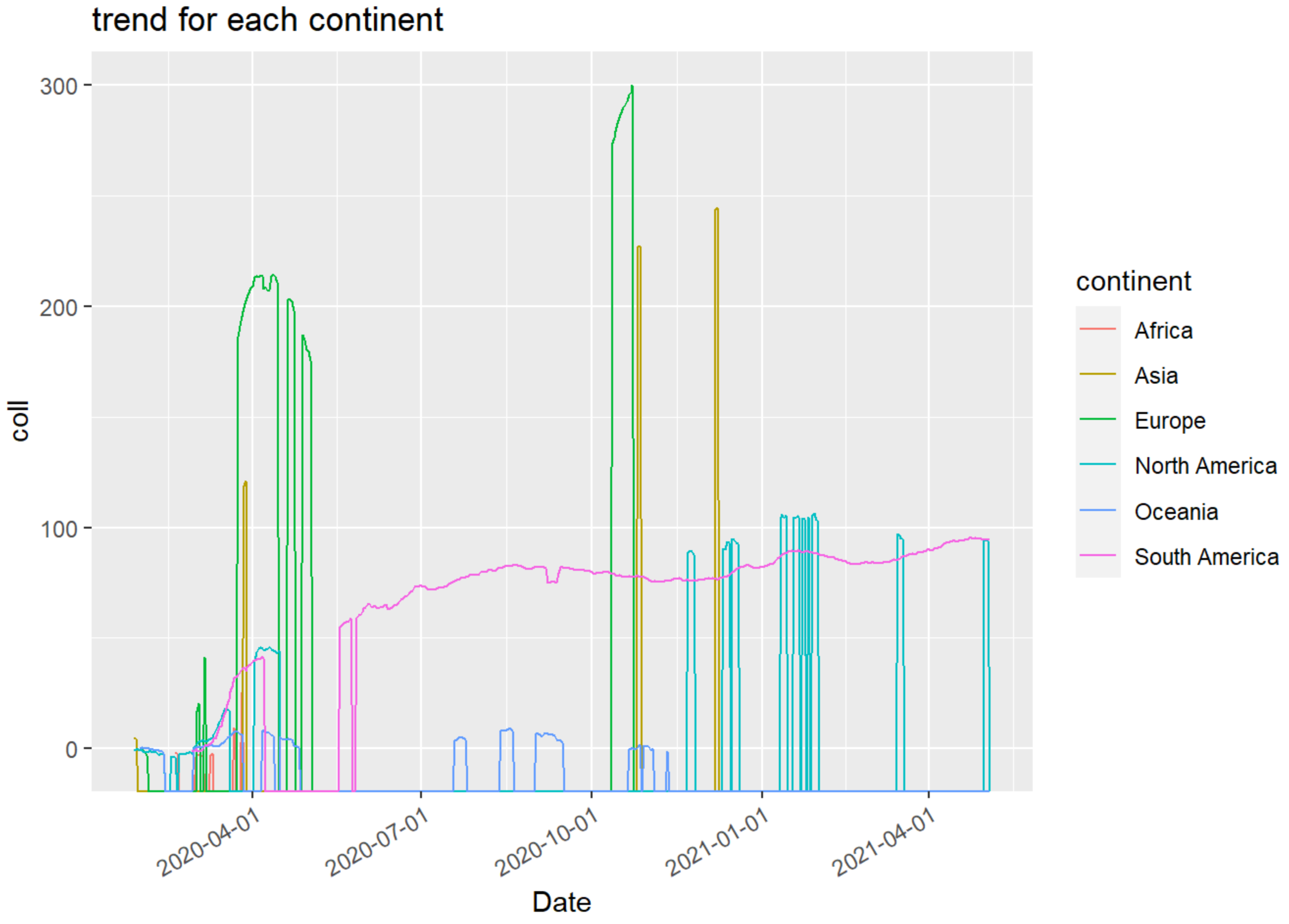
b

```
f1(df_clean,df_clean$new_cases)
```

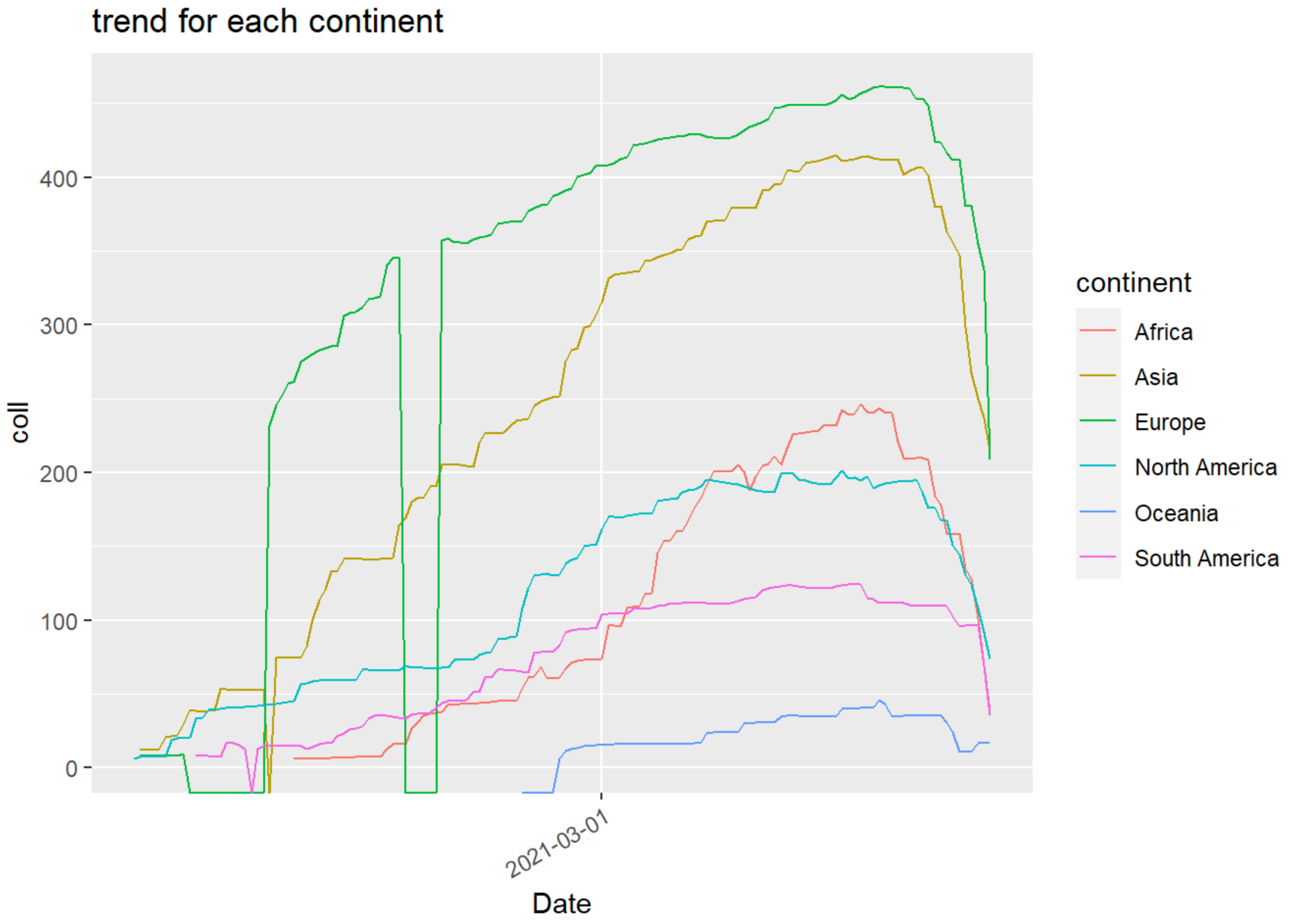


```
f1(df_clean,log(df_clean$new_cases_smoothed))
```

```
## Warning in log(df_clean$new_cases_smoothed): NaNs produced
```



```
f1(df_clean,log(df_clean$new_vaccinations_smoothed))
```



we can see that the second plot (the plot of log(new_cases)) is way more easier to interpret because when we do log we reduce the variance in ylab (new_cases_smoothed) so it easier to note the change between the continents in numbers. the max value in ylab is around 300, and in the first plot, the max value in ylab is around 500,000. we can see from the last plot that there is a positive correlation between the Date and the new_vaccinations_smoothed.

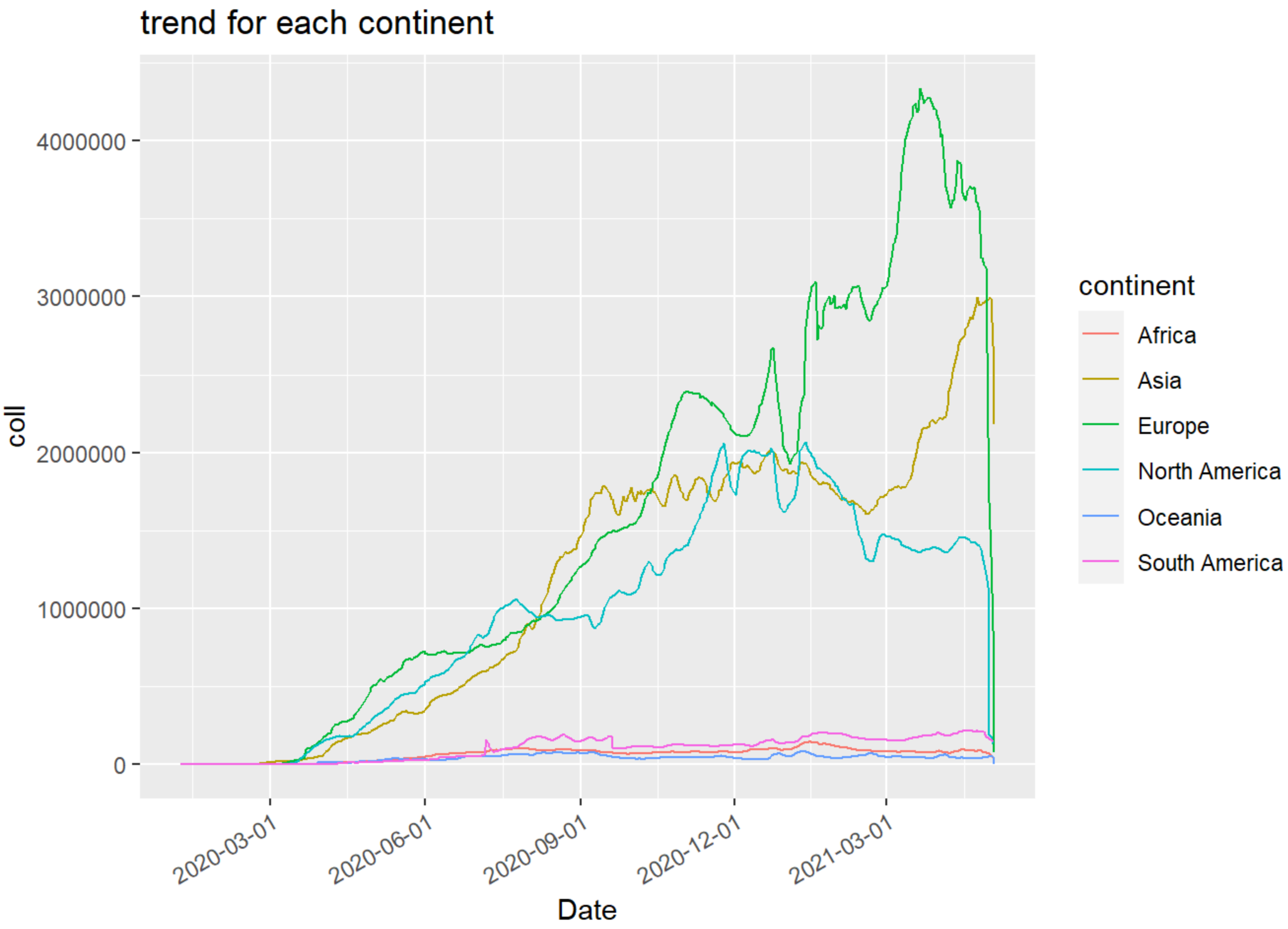
4.

a

```
#f2 = function(df_clean, col_com) {
 # ag4 = aggregate(new_tests~continent+date, df_clean,sum)
  #ag5 = aggregate(new_tests_smoothed~continent+date, df_clean,sum)
  #df4 = full_join(ag4,ag5)

#}

#ggplot(ag5,aes(x = date ,y = log(new_tests_smoothed), col = continent)) + geom_line()
```

b

```
f1(df_clean,df_clean$new_tests_smoothed)
```



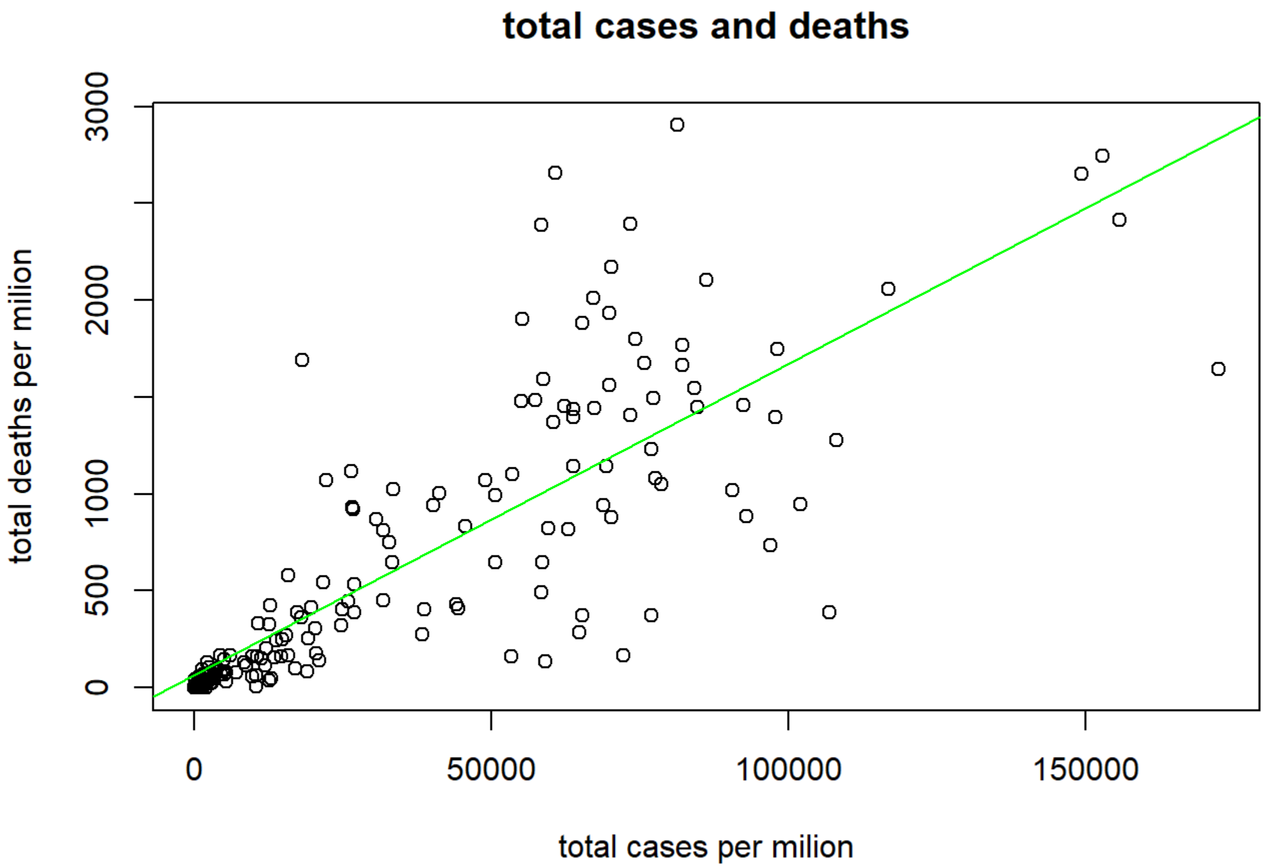
5.

Loading [MathJax]/jax/output/HTML-CSS/jax.js

Create a new data-frame with one row per country, that for each country will store as columns the current total_cases_per_million and total_deaths_per_million, in addition to the country name (location). We use aggregate for the columns and use full_join for merge them.

a

```
agg2a = aggregate(total_deaths_per_million~location, df, max)
agg2b = aggregate(total_cases_per_million~location, df, max)
comb_data = full_join(agg2a, agg2b, by = "location")
lm1 = lm(comb_data$total_deaths_per_million~comb_data$total_cases_per_million)

plot(comb_data$total_cases_per_million,comb_data$total_deaths_per_million,
     main = "total cases and deaths",xlab = "total cases per milion",
     ylab = "total deaths per milion")
abline(lm1, col = "green")
```



```
summary(lm1)
```

```
##
## Call:
## lm(formula = comb_data$total_deaths_per_million ~ comb_data$total_cases_per_million)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1393.43   -96.89   -60.76    84.71  1614.35
##
## Coefficients:
##                                     Estimate Std. Error t value
## (Intercept)                       63.8796598 39.6021480   1.613
## comb_data$total_cases_per_million  0.0160784  0.0008101  19.848
##                                              Pr(>|t|)
## (Intercept)                                     0.108
## comb_data$total_cases_per_million <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 406.9 on 188 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.6769, Adjusted R-squared:  0.6752
## F-statistic:   394 on 1 and 188 DF,  p-value: < 0.00000000000000022
```

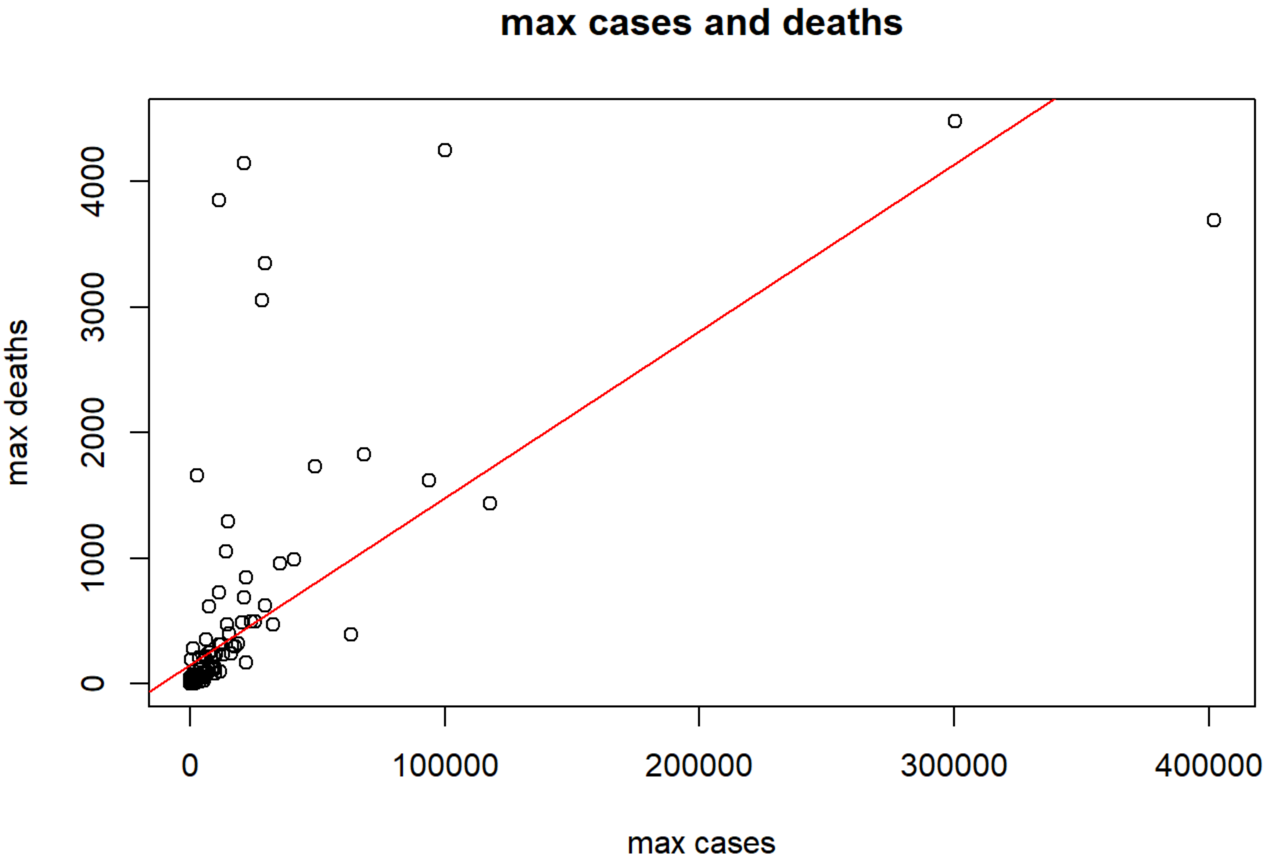
the slope is 0.0160784

b

```
max_cases = df_clean%>% filter(!is.na(new_cases))%>% select(location,date, new_cases)%>%
group_by(location) %>% top_n(1,new_cases) %>% top_n(1,date)
names(max_cases)[names(max_cases) == "date"] = "max_cases_date"
max_cases = na.omit(max_cases)
max_deaths = df_clean%>% filter(!is.na(new_deaths))%>% select(location,date, new_deaths)%>%
  group_by(location) %>% top_n(1,new_deaths) %>% top_n(1,date)
names(max_deaths)[names(max_deaths) == "date"] = "max_deaths_date"
max_deaths = na.omit(max_deaths)

comb_max = full_join(max_cases, max_deaths, by = "location")
comb_max = na.omit(comb_max)

lm2 = lm(comb_max$new_deaths~comb_max$new_cases)
summary(lm2)
```

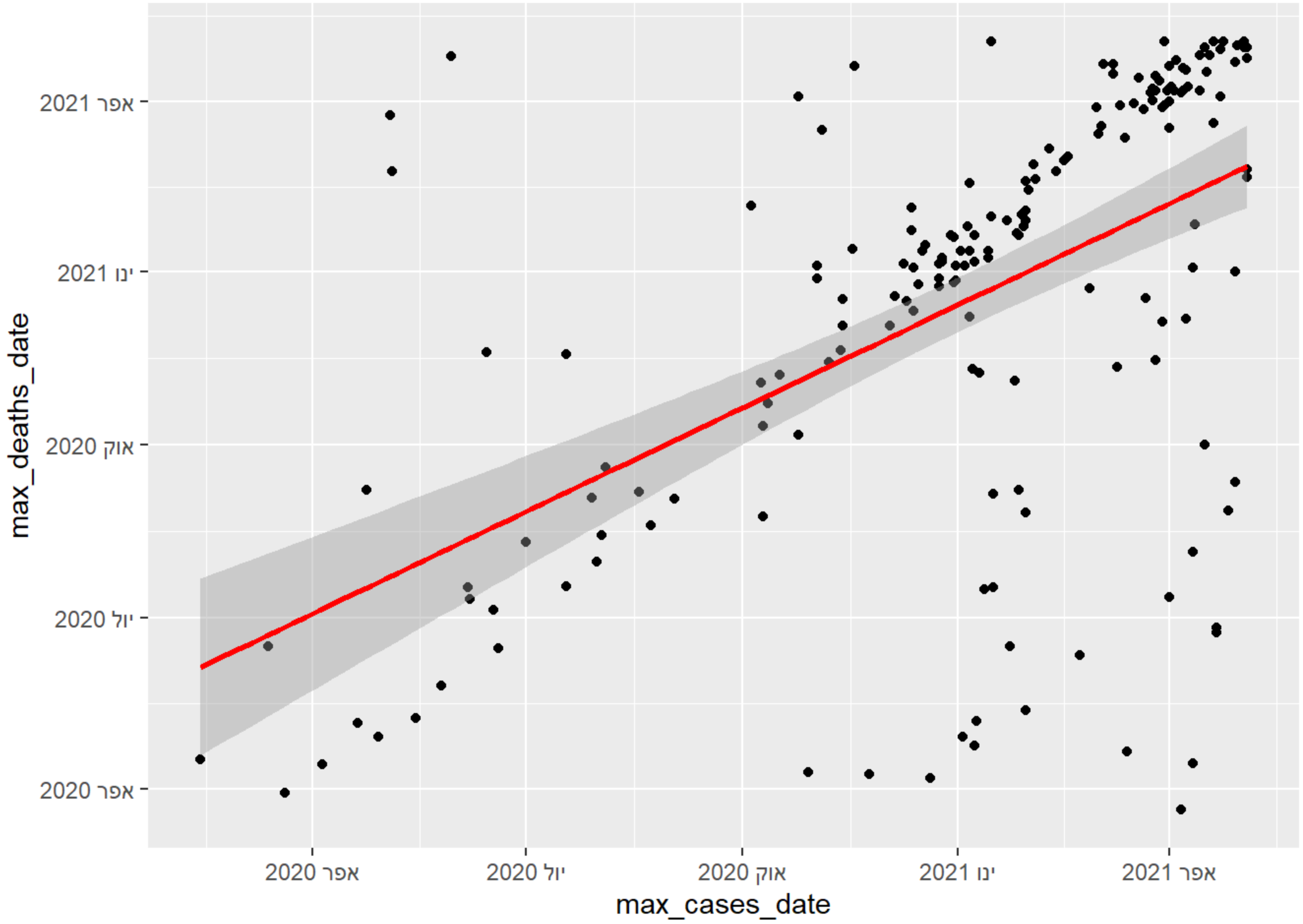
```
##
## Call:
## lm(formula = comb_max$new_deaths ~ comb_max$new_cases)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1804.2  -152.1  -145.3  -101.0  3706.9
##
## Coefficients:
##                      Estimate Std. Error t value             Pr(>|t|)
## (Intercept)        152.312209  44.836000   3.397             0.000838 ***
## comb_max$new_cases   0.013286   0.001086  12.231 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 581.8 on 180 degrees of freedom
## Multiple R-squared:  0.4539, Adjusted R-squared:  0.4508
## F-statistic: 149.6 on 1 and 180 DF,  p-value: < 0.00000000000000022
```

```
plot(comb_max$new_cases,comb_max$new_deaths,xlab = "max cases",ylab = "max deaths",
     main = "max cases and deaths")
abline(lm2,col = "red")
```



```
comb_df = full_join(comb_data, comb_max, by = "location")
comb_df = na.omit(comb_df)
comb_data = comb_df

lm3 = lm(comb_max$max_deaths_date~comb_max$max_cases_date)
options(scipen = 999)
ggplot(comb_data,aes(x = max_cases_date, y = max_deaths_date)) +
  geom_point() + geom_smooth(method="lm",color = "red")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

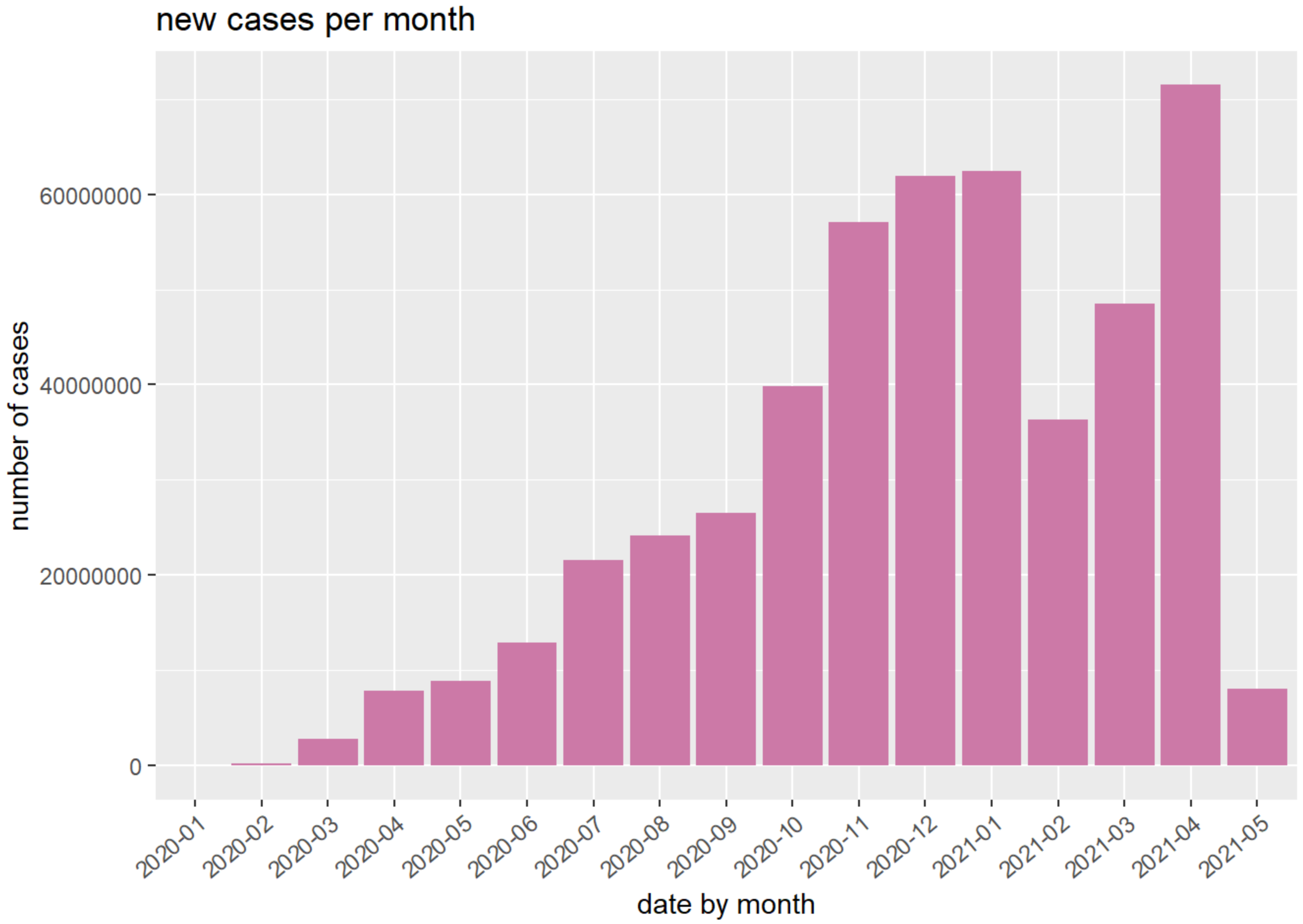


we show the max of new cases and new deaths by the slope. If the numbers of both cases will be close the slope will be close to 1.

6.

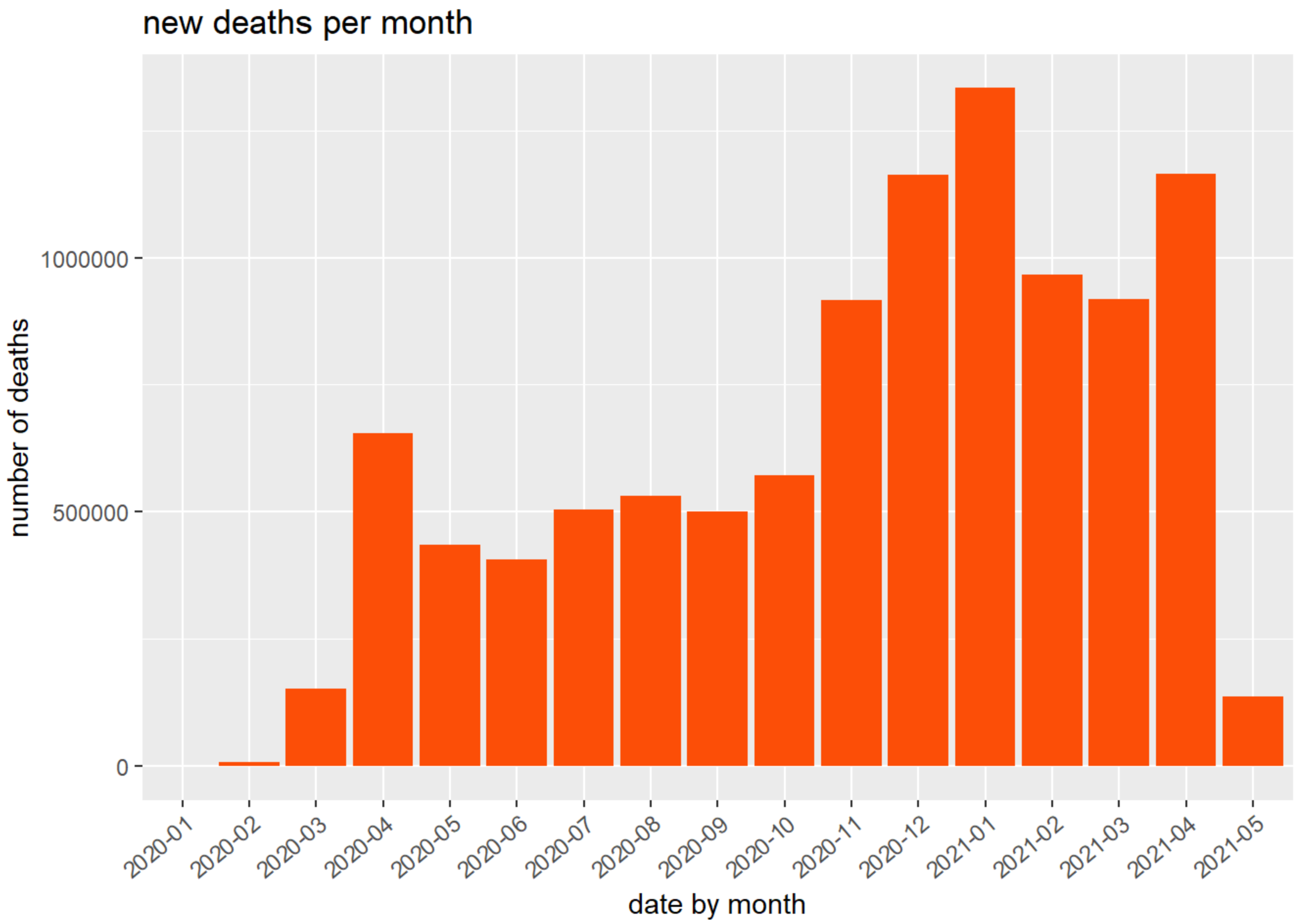
We want to compute the world-wide number of new_cases, new_deaths and new_vaccinations by month. each row corresponding to a month, and columns corresponding to the worldwide number of new cases, deaths or vaccinations in this month.

```r
wor_time= select(df, location,date, new_cases,new_deaths,new_vaccinations)
wor_time$date = as.Date(wor_time$date)

wor_time[is.na(wor_time)] = 0

date_frame = strftime(wor_time$date,"%Y-%m")
cases_agg = aggregate(new_cases~date_frame, wor_time, sum)
cases_deaths = aggregate(new_deaths~date_frame, wor_time, sum)
cases_vac = aggregate(new_vaccinations~date_frame, wor_time, sum)
monthly = data.frame(cases_agg,cases_deaths, cases_vac)

gg_by_cases = ggplot(monthly, aes(x = date_frame, y = new_cases)) + geom_bar(stat = "identity", fill = "#CC79A7") +
  labs(y = "number of cases", x = "date by month") +
  ggtitle("new cases per month") + theme(axis.text.x = element_text(angle=40, hjust=1))

gg_by_death = ggplot(monthly, aes(x = date_frame, y = new_deaths)) + geom_bar(stat = "identity",fill="#FC4E07") +
  labs(y = "number of deaths", x = "date by month") +
  ggtitle("new deaths per month") + theme(axis.text.x = element_text(angle=40, hjust=1))


gg_by_vac = ggplot(monthly, aes(x = date_frame, y = new_vaccinations)) + geom_bar(stat = "identity",fill="#00AFBB") +
  labs(y = "number of vaccinations", x = "date by month") +
  ggtitle("new vaccinations per month") + theme(axis.text.x = element_text(angle=40, hjust=1))

gg_by_cases
```

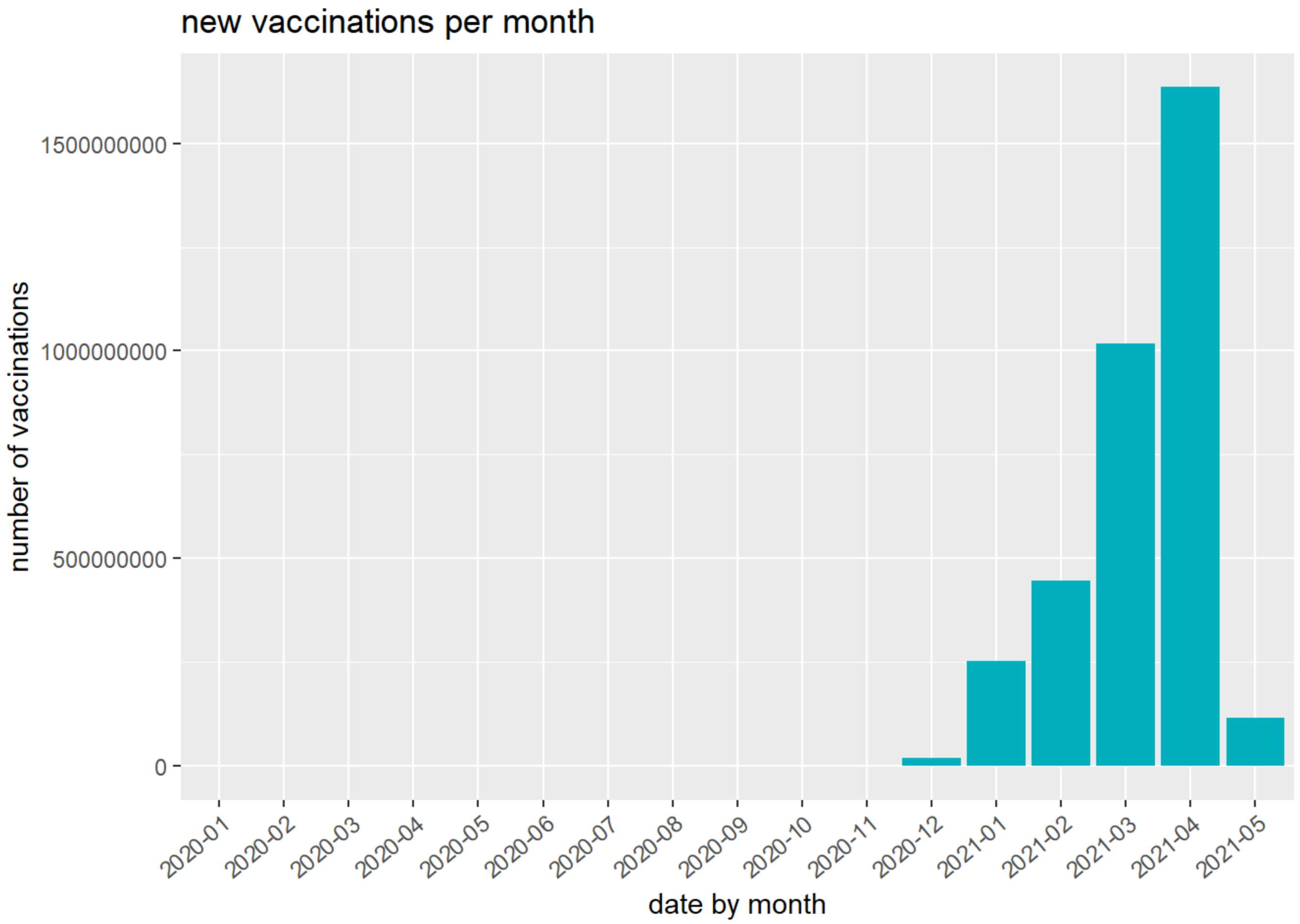


Loading [MathJax]/jax/output/HTML-CSS/jax.js

```
gg_by_death
```



```
gg_by_vac
```

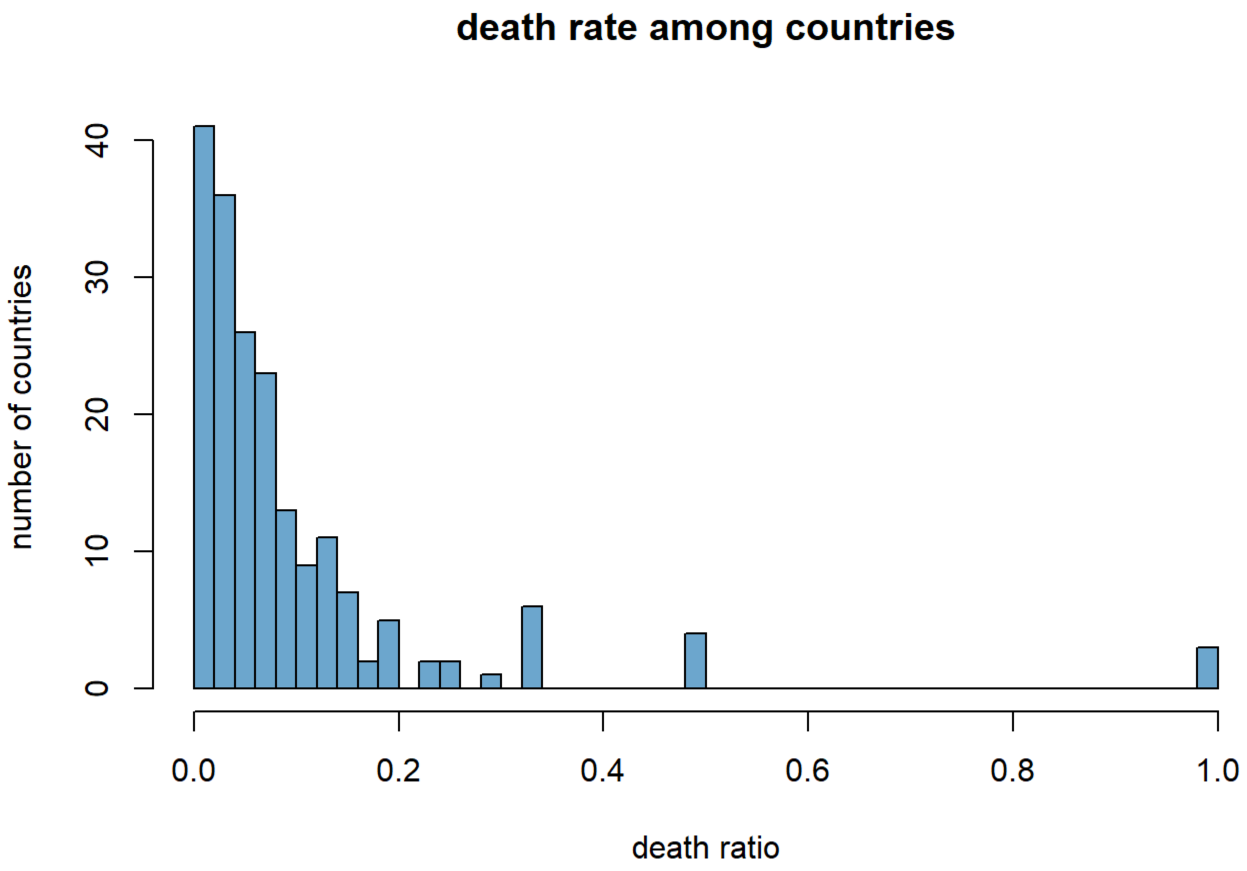


first, We changed all the vaccine cells who has NA to zero. we were supposed to sum all the world wide number of the three columns by months. we had at first some NA values so we changed them to 0, because zero value doesn’t change the sum.

7.

Add to the covid data-frame a new column called death_rate, defined for location and date as the number of total_deaths divided by the number of total_cases.

```
df$death_rate = df$total_deaths/df$total_cases

agg_deaths_loc = aggregate(total_deaths~location, df, max)
agg_cases_loc = aggregate(total_cases~location, df, max)
agg_deaths_date = aggregate(total_deaths~date, df, max)
agg_cases_date = aggregate(total_cases~date, df, max)

comb_deaths = full_join(agg_deaths_loc, agg_deaths_date, by = "total_deaths")
comb_cases = full_join(agg_cases_loc, agg_cases_date, by = "total_cases")

countries_death_rate = aggregate(death_rate~location, df, max)

hist(countries_death_rate$death_rate, col = 'skyblue3',breaks = 50,
     main = "death rate among countries",xlab = "death ratio",ylab = "number of countries")
```



return the top 3 countries having the highest death rate:

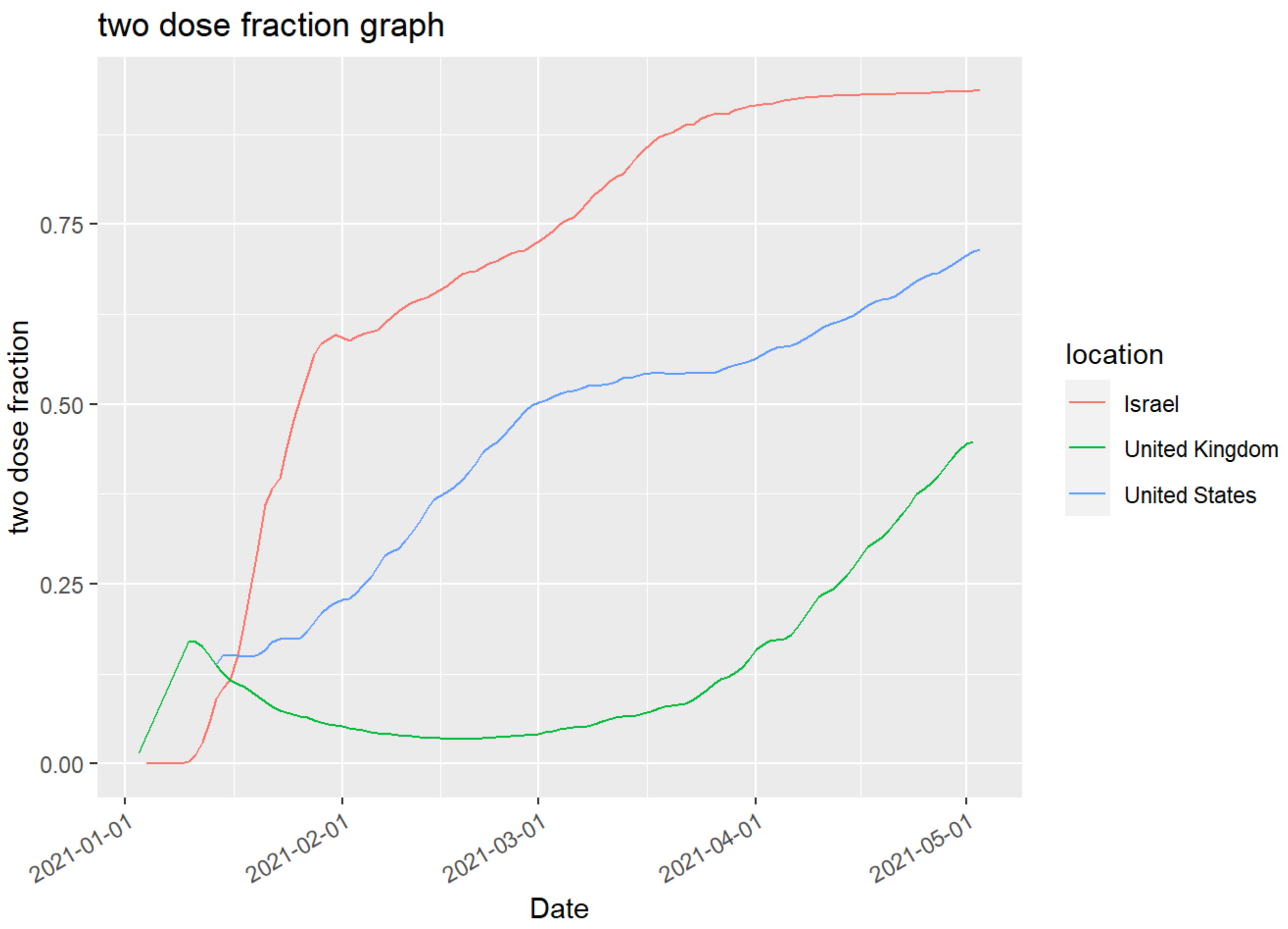
```
head(countries_death_rate%>%
       arrange(desc(countries_death_rate$death_rate))
     %>% filter(death_rate == max(death_rate))
     %>% select(location,death_rate),3)
```

```
##   location death_rate
## 1   Guyana          1
## 2     Iran          1
## 3    Sudan          1
```

8.

Given that most vaccinations are given in two-doses, we want to investigate whether different countries employ different vaccination strategies. While some countries vaccinate only individuals for which there are two doses of the vaccine given at proximity in time (usually less than one month apart), other countries first use the available vaccines to vaccinate as many people as possible using one dose, and may delay the second dose for these individuals.

```
df$two_dose_fraction = df$people_fully_vaccinated/df$people_vaccinated

order_data = aggregate(two_dose_fraction~location+date,df,sum)

country_3 = filter(order_data,(location == "Israel" | location == "United Kingdom" | location ==
"United States"))

plot8 =   ggplot(country_3, aes(x = as.Date(date), y = two_dose_fraction, color = location))+
  geom_line() + labs(y = "two dose fraction",x = "Date")+ scale_x_date(date_breaks = "1 months")+
  ggtitle("two dose fraction graph") + theme(axis.text.x = element_text(angle=30, hjust=1))
plot8
```



We can see Israel gives two doses all the way. USA did the same like Israel and UK in the beginning gave only one dose, but after a while they changed and switched to two doses.

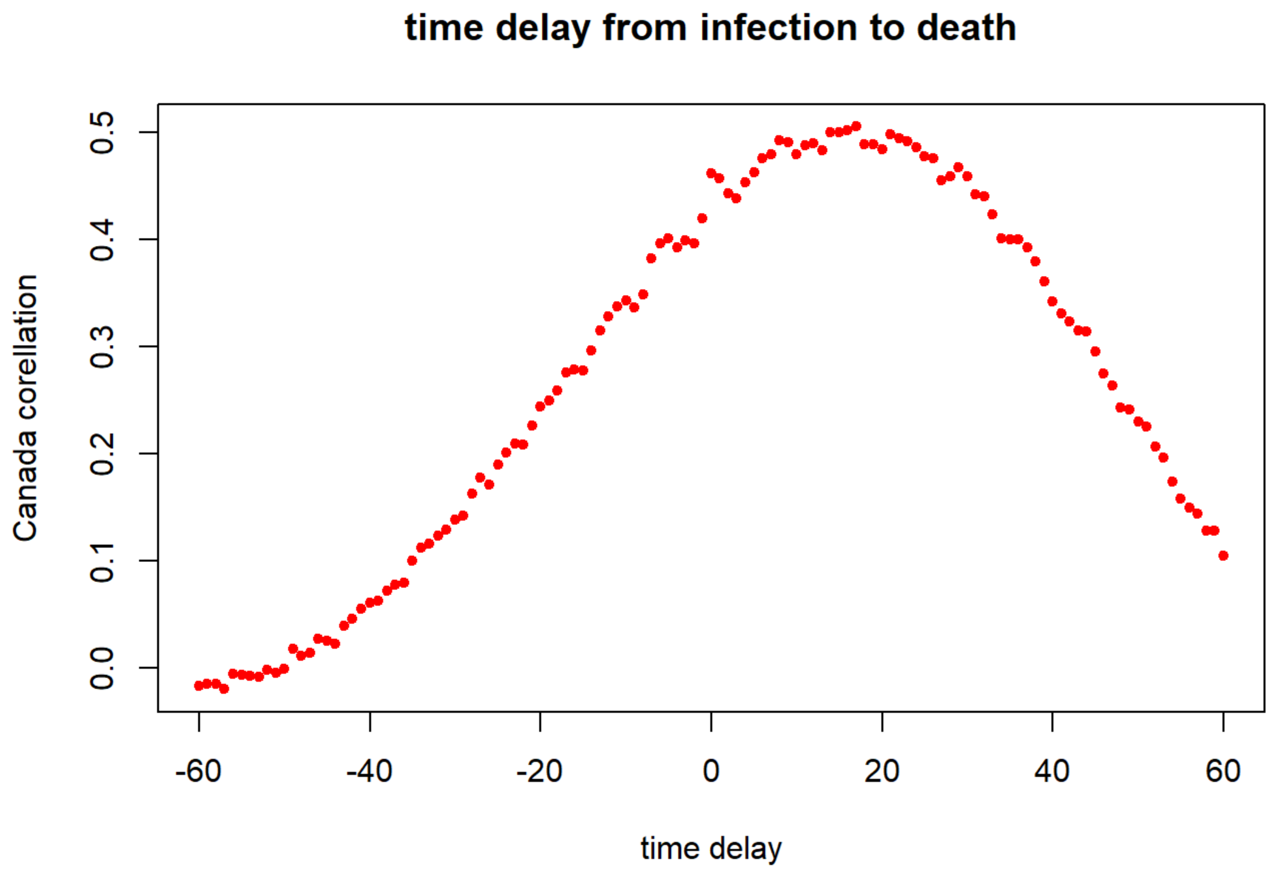
9.

a

```
cor_func = function(data, country, col_x, col_y){
  data_cor = subset(df, df$location==country)
  len_rows = nrow(data_cor)
  vector_cor = c()
  return_vec = c()

  for (row in -60:60){
    if ( row >= 0) {
      x_vec = data_cor[col_x][1:(len_rows - row),] %>% replace_na(0)
      y_vec = data_cor[col_y][(row + 1): len_rows,] %>% replace_na(0)
      return_vec = c(return_vec, cor(x_vec, y_vec))
    }
    else {
      x_vec = data_cor[col_x][(1-row):len_rows, ] %>% replace_na(0)
      y_vec = data_cor[col_y][1:(len_rows+row), ] %>% replace_na(0)
      return_vec = c(return_vec, cor(x_vec, y_vec))

    }
  }

  return(return_vec)}

canada_cor = cor_func(df,"Canada","new_cases", "new_deaths")
```

b

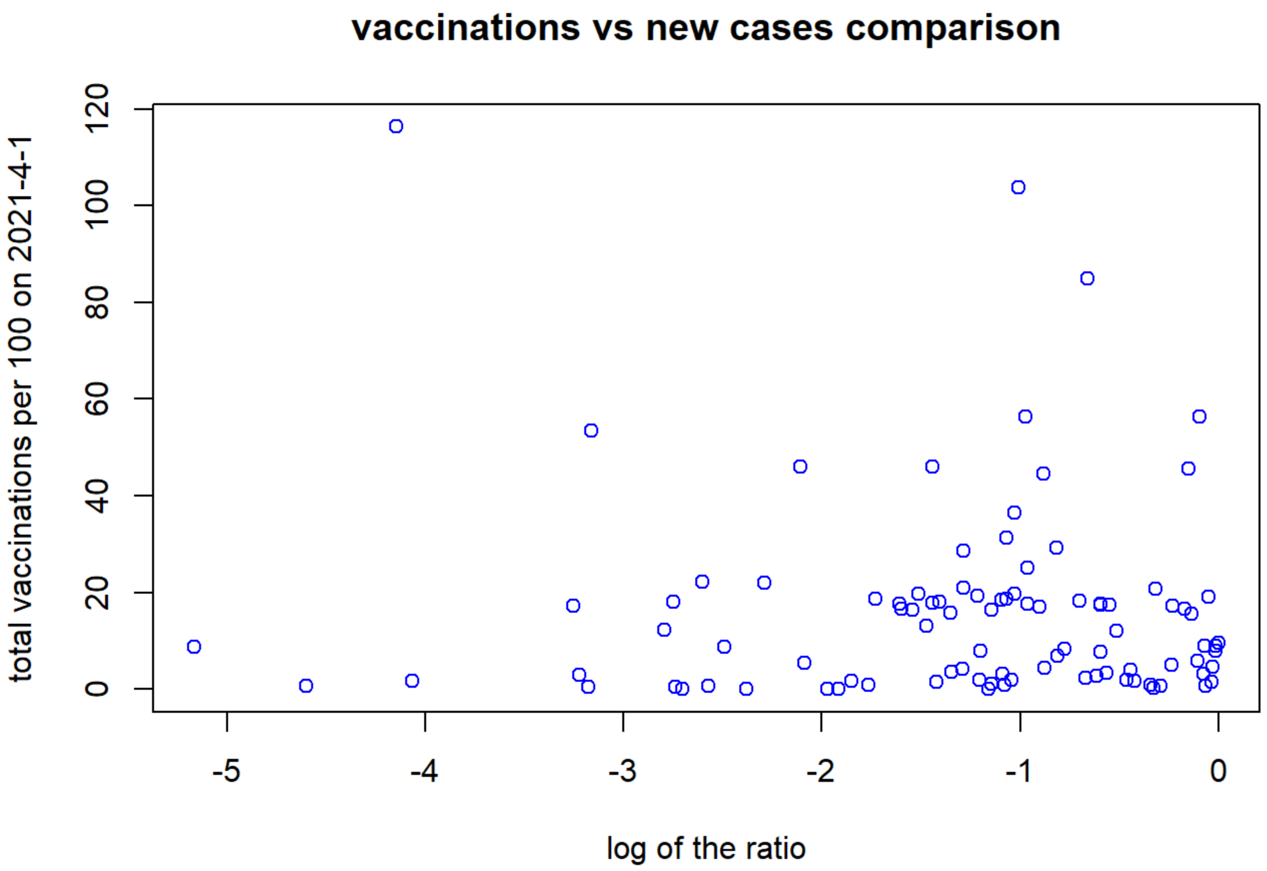
```
delta_t = seq(-60,60)

plot(delta_t,canada_cor,xlab = "time delay",
     ylab = "Canada corellation",
     main = "time delay from infection to death", col = "red",
     pch = 20)
```



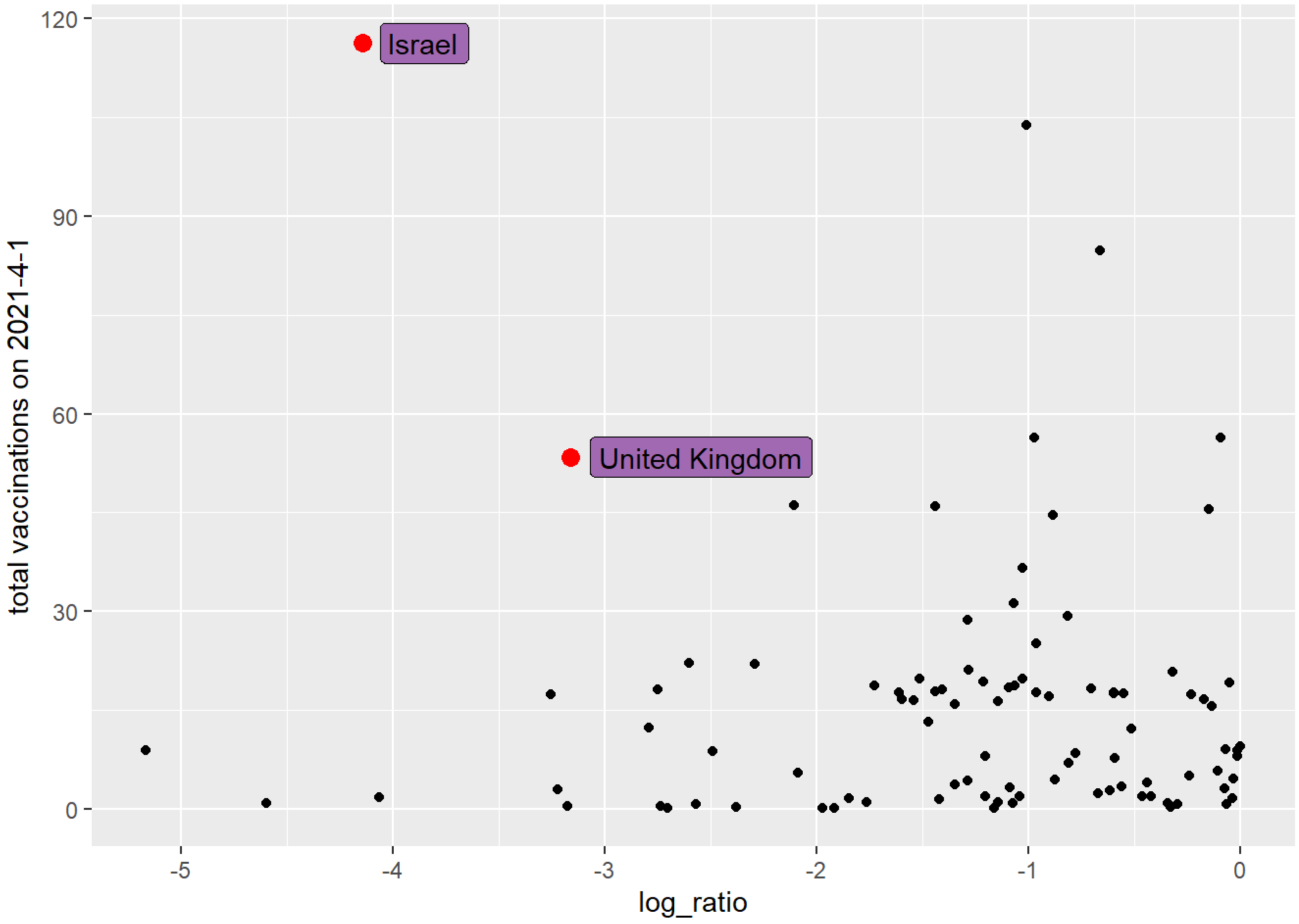
We will look at the numbers in absolute value and see that there is a peak after 20 days. From this it can be understood that after 20 days of illness the same person will die. the time delay, which the cross correlation maximized is on the value -19 approximately

10.

```
agg_temp = aggregate(new_cases_smoothed~location+date, df, sum)
agg10 = aggregate(new_cases_smoothed~location, df, max)
names(agg10)[names(agg10) == "new_cases_smoothed"] = "max_new_cases_smoothed"
agg11 = filter(agg_temp,(date == "2021-04-23"))
agg11$max_new_cases_smoothed = agg10$max_new_cases_smoothed
agg11$ratio = agg11$new_cases_smoothed/agg11$max_new_cases_smoothed
agg11 = na.omit(agg11)
agg_temp2 = aggregate(total_vaccinations_per_hundred~location+date, df, sum)
agg12 = filter(agg_temp2,(date == "2021-04-01"))
agg12$date = NULL
agg12 = na.omit(agg12)

total_join = full_join(agg11,agg12, by = "location", copy = T)
total_join = na.omit(total_join)
names(total_join)[names(total_join) == "total_vaccinations_per_hundred"] = "total vaccinations on 
2021-4-1"
total_join$log_ratio = log(total_join$ratio)

plot10 = plot(total_join$log_ratio, total_join$`total vaccinations on 2021-4-1`,
              ylab = "total vaccinations per 100 on 2021-4-1",
              xlab = "log of the ratio",
              main = "vaccinations vs new cases comparison",
              col = "blue")
```



```
gg = ggplot(total_join,aes(log_ratio,`total vaccinations on 2021-4-1`))+
geom_label(data=total_join %>% filter(location == "Israel"),
           aes(label="Israel"),fill = "#a269b3",hjust = -0.2)+
  geom_label(data=total_join %>% filter(location == "United Kingdom"),
             aes(label="United Kingdom"),
             fill = "#a269b3",hjust = -0.09) + geom_point() +
  geom_point(data=total_join %>% filter(location == "United Kingdom"),color = "red",size = 3)+
  geom_point(data=total_join %>% filter(location == "Israel"),color = "red",size = 3)
gg
```



we can see that for Israel and United Kingdom, the uses of two doses of vaccines reduce the ratio between new cases and max new cases. nevertheless, there are some countries that use two dose of vaccines and the ratio is still high. we can think about one explanation is that the vaccine comes from a different company than Pfizer and Moderna, which they less effective(less than 95%). we can see that Israel is really high, comparing to the other countries in the use of two dose vaccines, and those vaccines are really effective (low ratio).