covid-19

Alexey Sokolov

2022-07-18

Data source and overview

Data was downloaded from this website - https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series Data represents daily time series of COVID-19 cases including confirmed, recovered and deaths

Import data

here we just use read_csv data function that takes URL as a parameter

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names <- c("time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_global.csv",
                "time_series_covid19_confirmed_US.csv",
                "time_series_covid19_deaths_US.csv")
urls <- str_c(url_in, file_names)

global_cases <- read_csv(urls[1])
```

```
## Rows: 285 Columns: 925
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (923): 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 <- read_csv(urls[2])
```

```
## Rows: 285 Columns: 925
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (923): 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.
```

```
us_cases <- read_csv(urls[3])
```

```
## Rows: 3342 Columns: 932
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (926): 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 <- read_csv(urls[4])
```

```
## Rows: 3342 Columns: 933
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (927): 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.
```

Tidy data

here we "lengthen" data (decrease the number of columns and increase the number of rows). Previously we had column for every possible date and now we put this information in rows using pivot_longer function. Also we remove Lat and Long as we won't use it in further analysis. And finally we convert date to date format using mdy function

```
global_cases <- global_cases %>%
  pivot_longer(cols = -c(`Province/State`, `Country/Region`, Lat, Long),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(Lat, Long))

global_deaths <- global_deaths %>%
  pivot_longer(cols = -c(`Province/State`, `Country/Region`, Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat, Long))


us_cases <- us_cases %>%
  pivot_longer(cols = - (UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))

us_deaths <- us_deaths %>%
```

```
  pivot_longer(cols = -(UID:Population),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))
```

Transform data

Here we joint global_cases and global_deaths datasets using full_join column and filter all rows that don't have any cases using filter function. We also add information about population size using uid variable. For US data we just join us_cases with us_deaths and store new dataset in us variable

```
uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/
uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

```
## Rows: 4317 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): iso2, iso3, FIPS, Admin2, Province_State, Country_Region, Combined_Key
## dbl (5): UID, code3, Lat, Long_, Population
##
## 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 <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = `Country/Region`,
         Province_State = `Province/State`) %>%
  mutate(date = mdy(date))
```

```
## Joining, by = c("Province/State", "Country/Region", "date")
```

```
global <- global %>% filter(cases > 0)
global <- global %>%
  unite("Combined_Key",
        c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date,
         cases, deaths, Population,
         Combined_Key)

us <- us_cases %>%
  full_join(us_deaths)
```

```
## Joining, by = c("Admin2", "Province_State", "Country_Region", "Combined_Key",
## "date")
```

Summary of data

here we show summary of data using summary function

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:242772      Length:242772      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-10-14   1st Qu.:     857
##  Mode  :character   Mode  :character   Median :2021-05-24   Median :   12600
##                                        Mean   :2021-05-20   Mean   :  722044
##                                        3rd Qu.:2021-12-27   3rd Qu.:  183605
##                                        Max.   :2022-07-30   Max.   :91309159
##
##      deaths          Population        Combined_Key
##  Min.   :      0   Min.   :8.090e+02   Length:242772
##  1st Qu.:      6   1st Qu.:7.892e+05   Class :character
##  Median :    150   Median :7.133e+06   Mode  :character
##  Mean   :  12050   Mean   :2.923e+07
##  3rd Qu.:   2756   3rd Qu.:2.914e+07
##  Max.   :1029925   Max.   :1.380e+09
##                    NA's   :4953
```

```
summary(us)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3077982     Length:3077982     Length:3077982     Length:3077982
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases           Population          deaths
##  Min.   :2020-01-22   Min.   :  -3073   Min.   :       0   Min.   :  -82.0
##  1st Qu.:2020-09-08   1st Qu.:    165   1st Qu.:    9917   1st Qu.:    2.0
##  Median :2021-04-26   Median :   1543   Median :   24892   Median :   26.0
##  Mean   :2021-04-26   Mean   :  10376   Mean   :   99604   Mean   :  154.1
##  3rd Qu.:2021-12-12   3rd Qu.:   5841   3rd Qu.:   64979   3rd Qu.:   95.0
##  Max.   :2022-07-30   Max.   :3292692   Max.   :10039107   Max.   :32708.0
```

Visualizing data

for this, let's summarize number of cases, deaths, population and introduce new column - deaths per million. After that let's plot number of cases and deaths in USA and New York. From the plots bellow it's clear that these two characteristics are correlated.

```
us_by_state <- us %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
```

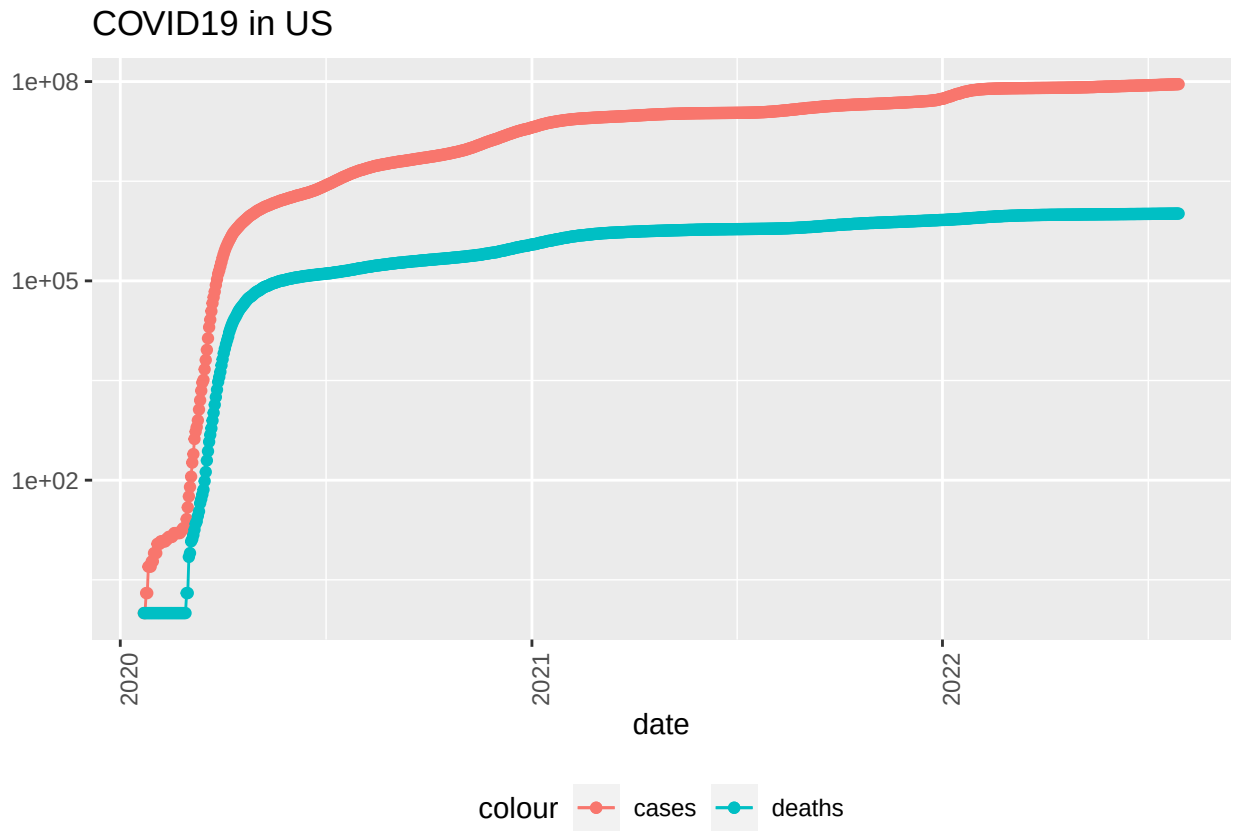
```
  select(Province_State, Country_Region, date, cases, deaths, deaths_per_mill,
         Population) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can
## override using the `.groups` argument.
```

```
us_totals <- us_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

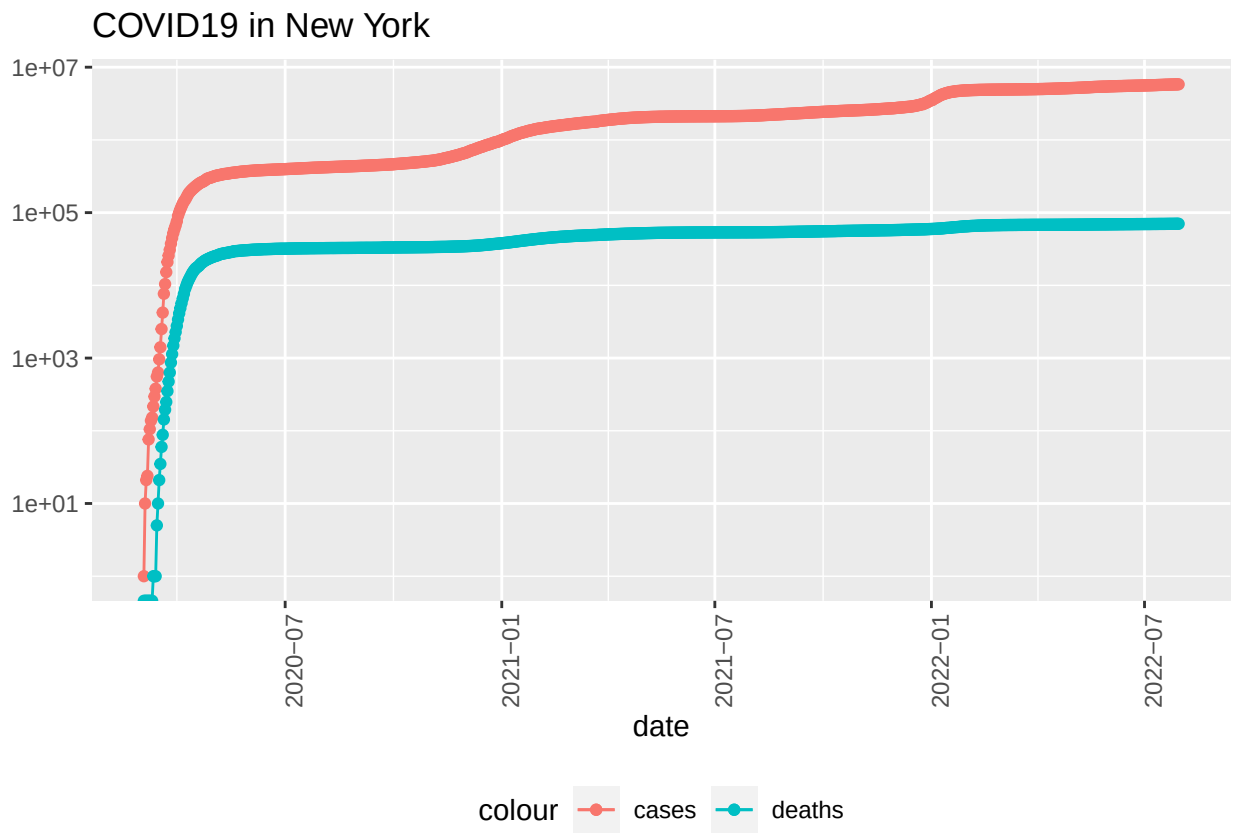
```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

```
us_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```




```
state <- "New York"
us_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y = NULL)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```




```
max(us_totals$date)
```

```
## [1] "2022-07-30"
```

```
max(us_totals$deaths)
```

```
## [1] 1029925
```

Analyzing data

Next let's introduce two new variables: new_cases and new_deaths and plot them in USA and New York.

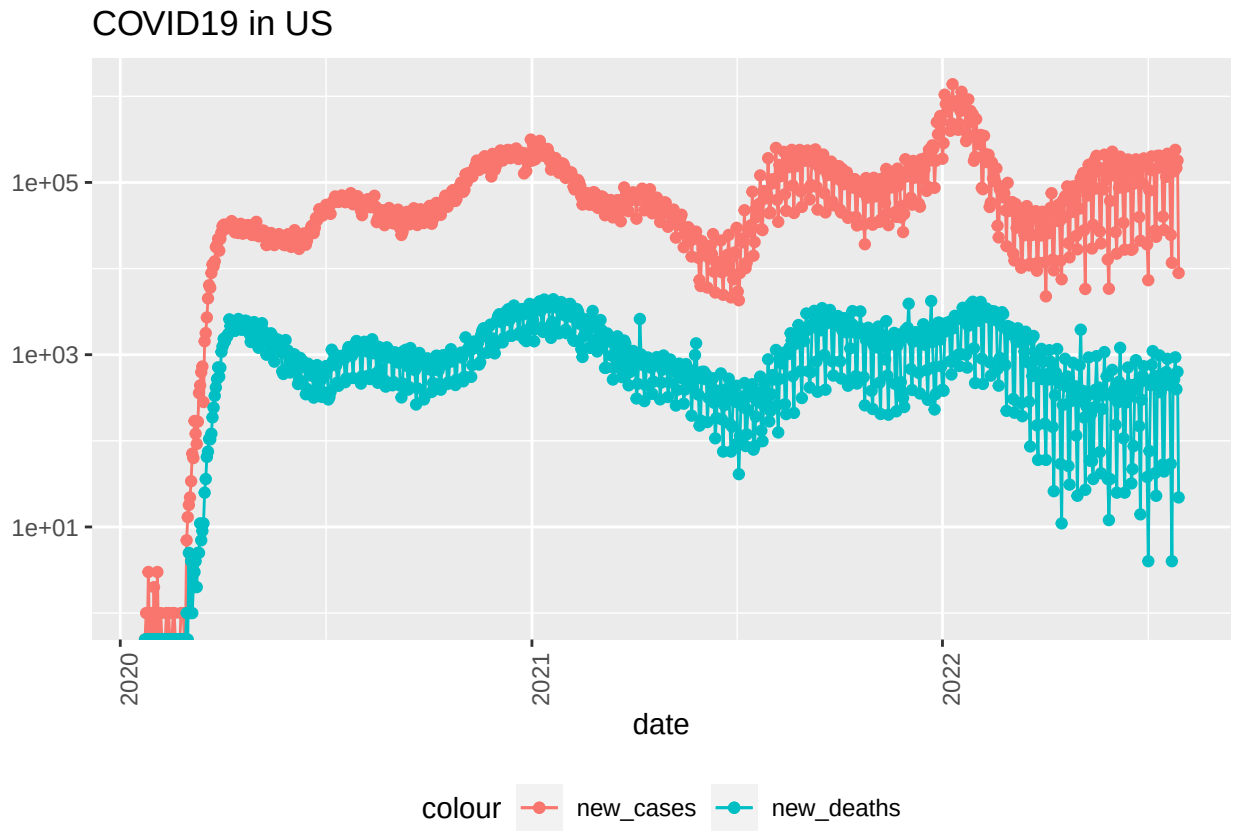
```
us_by_state <- us_by_state %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))

us_totals <- us_totals %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))
tail(us_totals %>% select(new_cases, new_deaths, everything()))
```

```
## # A tibble: 6 x 8
##   new_cases new_deaths Country_Region date          cases deaths deaths_per_mill
```

```
##       <dbl>      <dbl> <chr>          <date>        <dbl>  <dbl>           <dbl>
## 1    188900        418 US             2022-07-25 90598955 1.03e6           3087.
## 2    134725        517 US             2022-07-26 90733680 1.03e6           3088.
## 3    239176        933 US             2022-07-27 90972856 1.03e6           3091.
## 4    147362        397 US             2022-07-28 91120218 1.03e6           3092.
## 5    180053        633 US             2022-07-29 91300271 1.03e6           3094.
## 6      8888         22 US             2022-07-30 91309159 1.03e6           3094.
## # ... with 1 more variable: Population <dbl>
```

```
us_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```

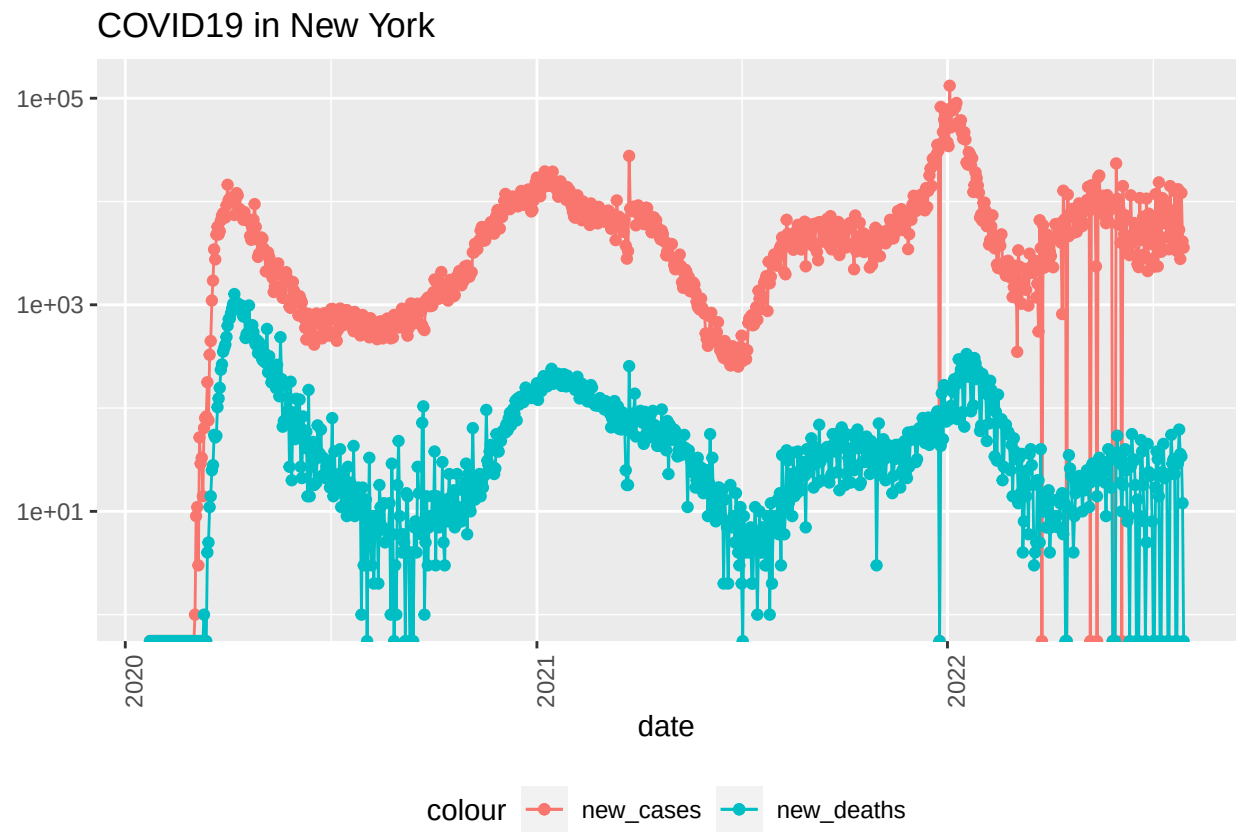
```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 2 rows containing missing values (geom_point).
```




```
state <- "New York"
us_by_state %>%
  filter(Province_State == state) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y = NULL)
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 6 rows containing missing values (geom_point).
```




```
us_state_totals <- us_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population),
            cases_per_thou = 1000 * cases / population,
            deaths_per_thou = 1000 * deaths / population) %>%
  filter(cases > 0, population > 0)

us_state_totals %>%
  slice_min(deaths_per_thou, n = 10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State        deaths  cases population
##              <dbl>          <dbl> <chr>                 <dbl>  <dbl>      <dbl>
##  1           0.593           134. American Samoa           33 7.47e3      55641
##  2           0.671           228. Northern Mariana Isl~    37 1.26e4      55144
##  3           1.11            230. Hawaii                 1571 3.26e5    1415872
##  4           1.11            221. Vermont                 693 1.38e5     623989
##  5           1.12            203. Virgin Islands          120 2.18e4     107268
##  6           1.27            227. Puerto Rico            4767 8.51e5    3754939
##  7           1.53            315. Utah                   4900 1.01e6    3205958
##  8           1.77            383. Alaska                 1309 2.84e5     740995
##  9           1.79            227. Washington            13604 1.73e6    7614893
## 10           1.84            205. Maine                  2467 2.76e5    1344212
```

```
us_state_totals %>%
  slice_max(deaths_per_thou, n = 10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths   cases population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.25           294. Mississippi     12648  875132    2976149
##  2            4.23           302. Arizona         30768 2196429    7278717
##  3            4.11           283. Oklahoma        16252 1120934    3956971
##  4            4.06           291. Alabama         19891 1424411    4903185
##  5            3.99           310. West Virginia    7156  555107    1792147
##  6            3.95           322. Tennessee       27006 2198946    6829174
##  7            3.93           282. New Mexico       8246  591041    2096829
##  8            3.88           299. Arkansas        11719  903555    3017804
##  9            3.86           291. New Jersey      34253 2582493    8882190
## 10            3.78           296. Louisiana       17571 1377666    4648794
```

Model data in the USA

Here we will use linear model to build a model of a dependency between deaths per thousand and cases per thousand. When we have model, we can then plot it against real data that we have.

```
mod <- lm(deaths_per_thou ~ cases_per_thou, data = us_state_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = us_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3267 -0.5683  0.1173  0.7168  1.1550
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.213727   0.688778  -0.310    0.758
## cases_per_thou  0.011252   0.002491   4.518 3.44e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.834 on 54 degrees of freedom
## Multiple R-squared:  0.2743, Adjusted R-squared:  0.2608
## F-statistic: 20.41 on 1 and 54 DF,  p-value: 3.443e-05
```

```
us_state_totals %>% slice_min(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1 American Samoa     33  7471      55641           134.           0.593
```

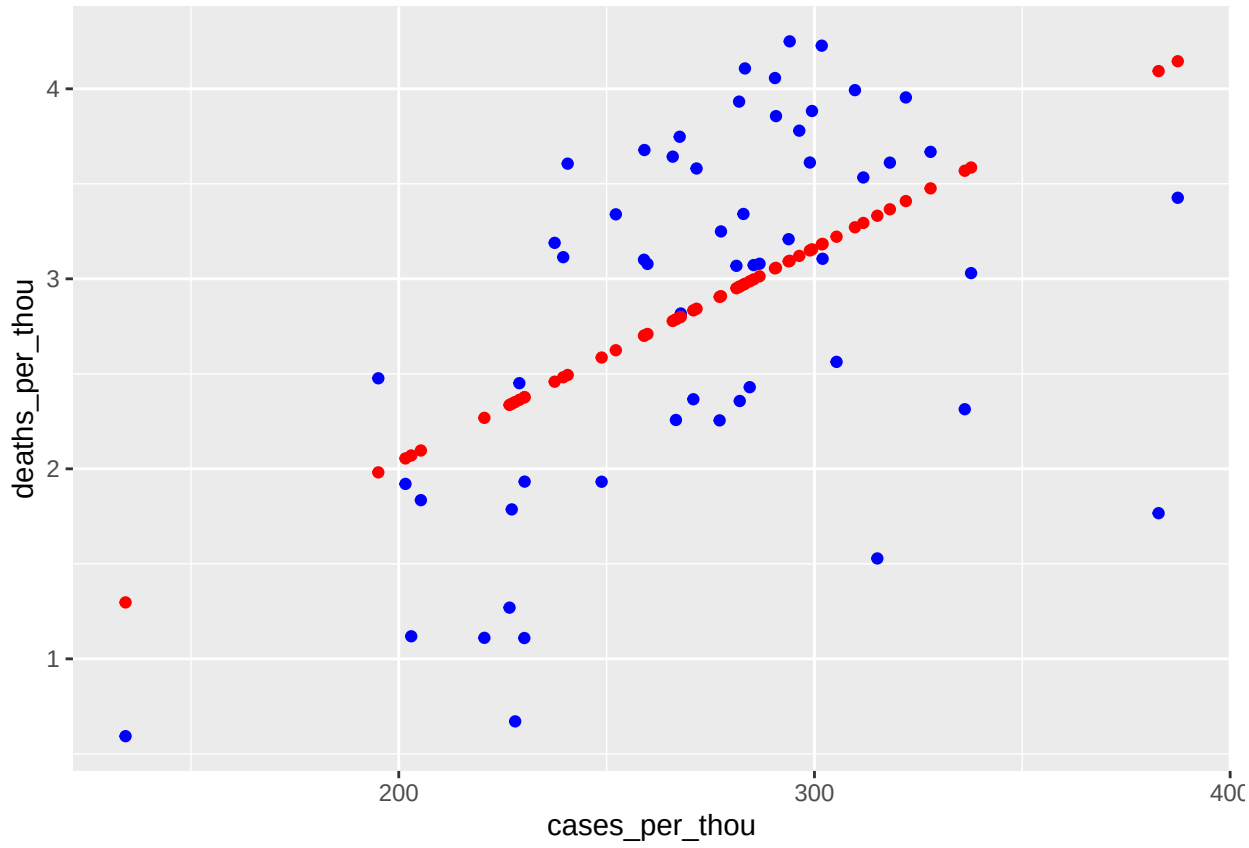
```
us_state_totals %>% slice_max(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths  cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 Rhode Island     3630 410399    1059361           387.            3.43
```

```
us_state_totals %>% mutate(pred = predict(mod))
```

```
## # A tibble: 56 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          19891 1.42e6    4903185           291.            4.06   3.05
##  2 Alaska            1309 2.84e5     740995           383.            1.77   4.09
##  3 American Samoa      33 7.47e3      55641           134.            0.593  1.30
##  4 Arizona          30768 2.20e6    7278717           302.            4.23   3.18
##  5 Arkansas         11719 9.04e5    3017804           299.            3.88   3.16
##  6 California       93491 1.07e7   39512223           271.            2.37   2.83
##  7 Colorado         12986 1.60e6    5758736           277.            2.26   2.91
##  8 Connecticut      11102 8.54e5    3565287           240.            3.11   2.48
##  9 Delaware          3024 2.94e5     973764           302.            3.11   3.18
## 10 District of Co~   1364 1.63e5     705749           230.            1.93   2.38
## # ... with 46 more rows
```

```
us_tot_w_pred <- us_state_totals %>% mutate(pred = predict(mod))
us_tot_w_pred %>% ggplot() +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") +
  geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```

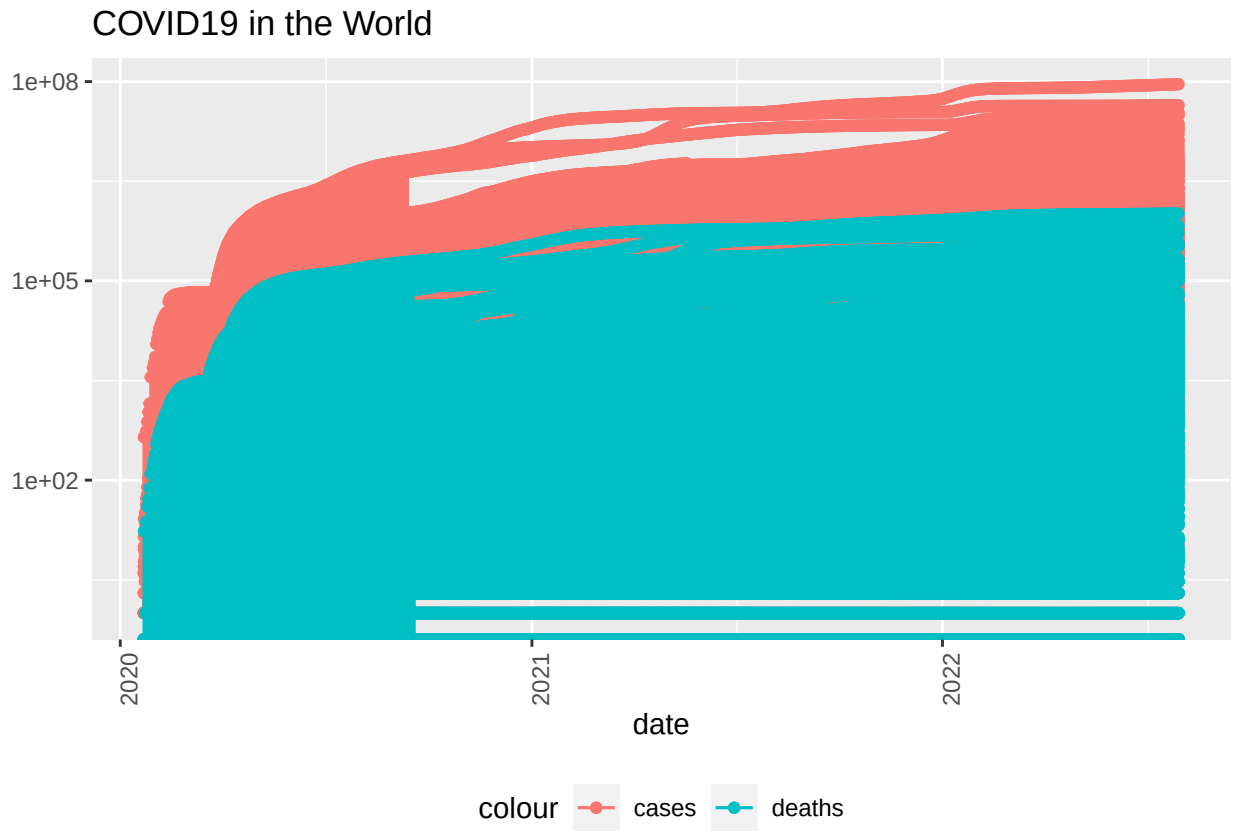


Analyze worldwide data

let's plot a graph of global deaths and global cases and then a graph of cases and deaths in China. For China I can clearly see huge increase in number of cases for 2022.

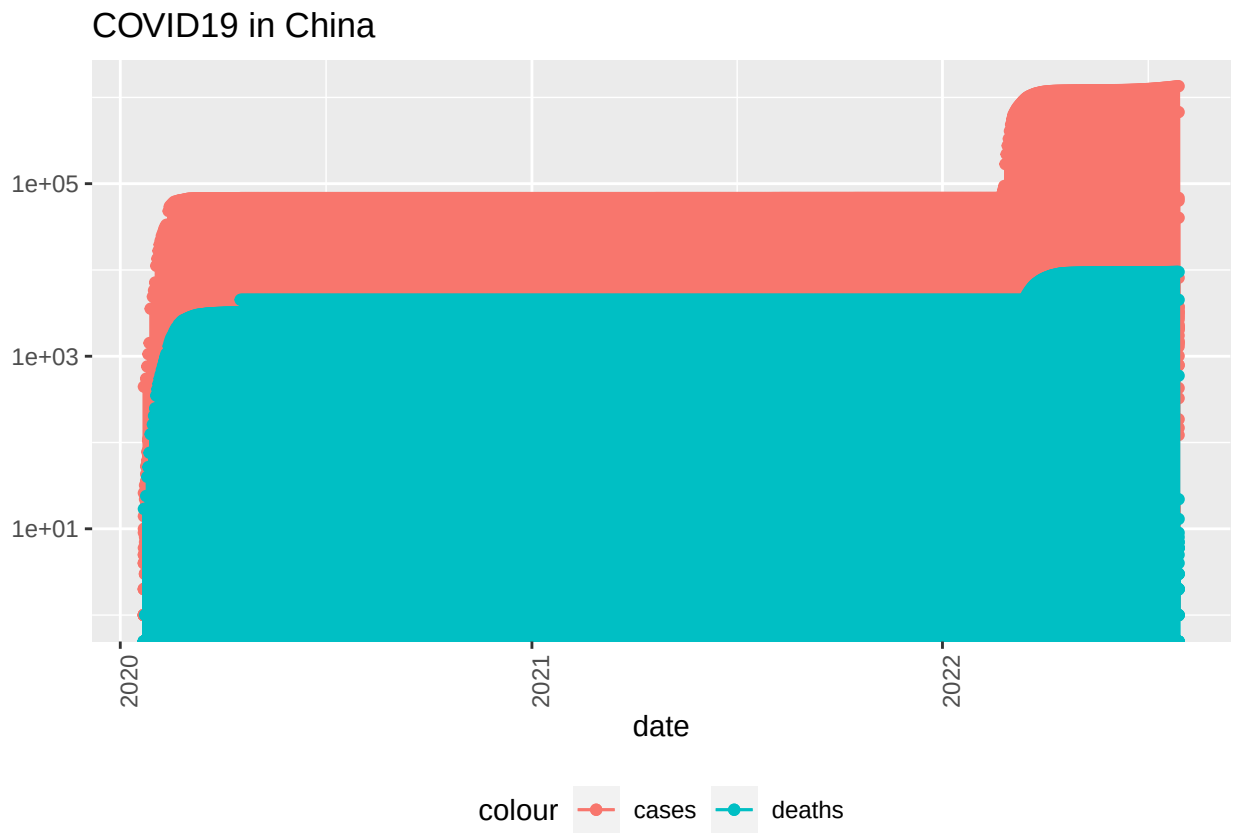
```
global %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in the World", y = NULL)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```




```
country <- "China"
global %>%
  filter(Country_Region == country) %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", country), y = NULL)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```



Linear model for global data

Next let's build linear model for global data and try to find a model for a number of cases and number of deaths dependency.

```
mod <- lm(cases ~ deaths, data = global)
summary(mod)
```

```
##
## Call:
## lm(formula = cases ~ deaths, data = global)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -14998550       -94      3339     11430  29321953
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -3.207e+03  3.378e+03   -0.949    0.342
## deaths       6.019e+01  5.747e-02 1047.234   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1629000 on 242770 degrees of freedom
## Multiple R-squared:  0.8188, Adjusted R-squared:  0.8188
```

```
## F-statistic: 1.097e+06 on 1 and 242770 DF,  p-value: < 2.2e-16
```

Bias analysis

As any human I have lots of biases. For example I can have a biases towards some specific countries. I think there are two possible solutions to mitigate that:

1. Analyze data in all possible combinations with all possible types of visualizations. That's ideal solution but quite often it's just not feasible
2. Work only with unpersonalized data, for example instead of Categorical country_region columns it's possible to use some numbers instead (for example 0 - Afghanistan and 1 - China, etc. . .)