Covid 19 analysis

Jie Shen

2023-09-24

File and Data

This is a R Markdown document for **COVID 19 project for China**. The data used in this project can be found at "https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series". Please visit the site for detailed data description.

Project goal

The project is to discover patterns and trends from Covid data in China. I want to explore things like the Covid cases and deaths trends over the years, and what states are best and worst.

Packages needed

Be sure the following packages are installed first:

- tidyverse
- ggplot2

Load Packages

```r
library(tidyverse)
library(ggplot2)
library(forcats)
library(lubridate)
```

Import Data and clean up

```r
#Import data from webnsite
url_in<-"https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
file_names<-c("time_series_covid19_confirmed_global.csv","time_series_covid19_deaths_global.csv")

urls=str_c(url_in, file_names)
global_cases<-read_csv(urls[1])
```

```
## 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<-read_csv(urls[2])
```

```
## 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.
```

Now let’s take a look and do some clean up

```
# Take a look
head(global_cases)
```

```
## # A tibble: 6 x 1,147
##   ‘Province/State‘ ‘Country/Region‘   Lat  Long ‘1/22/20‘ ‘1/23/20‘ ‘1/24/20‘
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # i 1,140 more variables: ‘1/25/20‘ <dbl>, ‘1/26/20‘ <dbl>, ‘1/27/20‘ <dbl>,
## #   ‘1/28/20‘ <dbl>, ‘1/29/20‘ <dbl>, ‘1/30/20‘ <dbl>, ‘1/31/20‘ <dbl>,
## #   ‘2/1/20‘ <dbl>, ‘2/2/20‘ <dbl>, ‘2/3/20‘ <dbl>, ‘2/4/20‘ <dbl>,
## #   ‘2/5/20‘ <dbl>, ‘2/6/20‘ <dbl>, ‘2/7/20‘ <dbl>, ‘2/8/20‘ <dbl>,
## #   ‘2/9/20‘ <dbl>, ‘2/10/20‘ <dbl>, ‘2/11/20‘ <dbl>, ‘2/12/20‘ <dbl>,
## #   ‘2/13/20‘ <dbl>, ‘2/14/20‘ <dbl>, ‘2/15/20‘ <dbl>, ‘2/16/20‘ <dbl>,
## #   ‘2/17/20‘ <dbl>, ‘2/18/20‘ <dbl>, ‘2/19/20‘ <dbl>, ‘2/20/20‘ <dbl>, ...
```

```
head(global_deaths)
```

```
## # A tibble: 6 x 1,147
##   ‘Province/State‘ ‘Country/Region‘   Lat  Long ‘1/22/20‘ ‘1/23/20‘ ‘1/24/20‘
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # i 1,140 more variables: ‘1/25/20‘ <dbl>, ‘1/26/20‘ <dbl>, ‘1/27/20‘ <dbl>,
## #   ‘1/28/20‘ <dbl>, ‘1/29/20‘ <dbl>, ‘1/30/20‘ <dbl>, ‘1/31/20‘ <dbl>,
## #   ‘2/1/20‘ <dbl>, ‘2/2/20‘ <dbl>, ‘2/3/20‘ <dbl>, ‘2/4/20‘ <dbl>,
## #   ‘2/5/20‘ <dbl>, ‘2/6/20‘ <dbl>, ‘2/7/20‘ <dbl>, ‘2/8/20‘ <dbl>,
## #   ‘2/9/20‘ <dbl>, ‘2/10/20‘ <dbl>, ‘2/11/20‘ <dbl>, ‘2/12/20‘ <dbl>,
## #   ‘2/13/20‘ <dbl>, ‘2/14/20‘ <dbl>, ‘2/15/20‘ <dbl>, ‘2/16/20‘ <dbl>,
## #   ‘2/17/20‘ <dbl>, ‘2/18/20‘ <dbl>, ‘2/19/20‘ <dbl>, ‘2/20/20‘ <dbl>, ...
```

```
# Need to pivot dates to rows
global_cases<-global_cases %>%
  pivot_longer(cols= -c("Province/State", "Country/Region", Lat, Long),
                       names_to="date",
                       values_to="cases")
head(global_cases)
```

```
## # A tibble: 6 x 6
##   ‘Province/State‘ ‘Country/Region‘   Lat  Long date    cases
##   <chr>            <chr>            <dbl> <dbl> <chr>   <dbl>
## 1 <NA>             Afghanistan       33.9  67.7 1/22/20     0
## 2 <NA>             Afghanistan       33.9  67.7 1/23/20     0
## 3 <NA>             Afghanistan       33.9  67.7 1/24/20     0
## 4 <NA>             Afghanistan       33.9  67.7 1/25/20     0
## 5 <NA>             Afghanistan       33.9  67.7 1/26/20     0
## 6 <NA>             Afghanistan       33.9  67.7 1/27/20     0
```

```
# Do similar things to global deaths
global_deaths<-global_deaths %>%
  pivot_longer(cols= -c("Province/State", "Country/Region", Lat, Long),
                       names_to="date",
                       values_to="deaths")

# Combine global cases and deaths
global<- global_cases %>%
        full_join(global_deaths) %>%
        mutate(date=mdy(date)) %>%
        rename(Country_Region='Country/Region',
               Province_State ='Province/State')
```

```
## Joining with ‘by = join_by(‘Province/State‘, ‘Country/Region‘, Lat, Long,
## date)‘
```

```
# Take a look again
head(global)
```

```
## # A tibble: 6 x 7
##   Province_State Country_Region   Lat  Long date       cases deaths
##   <chr>          <chr>          <dbl> <dbl> <date>     <dbl>  <dbl>
## 1 <NA>           Afghanistan     33.9  67.7 2020-01-22     0      0
## 2 <NA>           Afghanistan     33.9  67.7 2020-01-23     0      0
## 3 <NA>           Afghanistan     33.9  67.7 2020-01-24     0      0
## 4 <NA>           Afghanistan     33.9  67.7 2020-01-25     0      0
## 5 <NA>           Afghanistan     33.9  67.7 2020-01-26     0      0
## 6 <NA>           Afghanistan     33.9  67.7 2020-01-27     0      0
```

```
# US data has "Combined_Key". Add this to global data too.
global<-global%>%
  unite("Combined_Key",
       c("Province_State", "Country_Region"),
       sep=", ",
       na.rm=TRUE,
       remove=FALSE
  )
```

```
# US data has "Combined_Key". Add this to global data too.
global<-global%>%
  unite("Combined_Key",
       c("Province_State", "Country_Region"),
       sep=", ",
       na.rm=TRUE,
       remove=FALSE
  )
```

```
# Take another look
head(global)
```

```
## # A tibble: 6 x 8
##   Combined_Key Province_State Country_Region   Lat  Long date       cases deaths
##   <chr>        <chr>          <chr>          <dbl> <dbl> <date>     <dbl>  <dbl>
## 1 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-22     0      0
## 2 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-23     0      0
## 3 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-24     0      0
## 4 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-25     0      0
## 5 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-26     0      0
## 6 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-27     0      0
```

```
# Summary statistics
summary(global)
```

```
##  Combined_Key       Province_State     Country_Region          Lat
##  Length:330327      Length:330327      Length:330327      Min.   :-71.950
##  Class :character   Class :character   Class :character   1st Qu.:  3.934
##  Mode  :character   Mode  :character   Mode  :character   Median : 21.513
##                                                           Mean   : 19.719
##                                                           3rd Qu.: 40.464
##                                                           Max.   : 71.707
##                                                           NA’s   :2286
##       Long              date                cases               deaths
##  Min.   :-178.12   Min.   :2020-01-22   Min.   :        0   Min.   :      0
##  1st Qu.: -42.60   1st Qu.:2020-11-02   1st Qu.:      680   1st Qu.:      3
##  Median :  20.94   Median :2021-08-15   Median :    14429   Median :    150
##  Mean   :  22.18   Mean   :2021-08-15   Mean   :   959384   Mean   :  13380
##  3rd Qu.:  90.36   3rd Qu.:2022-05-28   3rd Qu.:   228517   3rd Qu.:   3032
##  Max.   : 178.06   Max.   :2023-03-09   Max.   :103802702   Max.   :1123836
##  NA’s   :2286
```

We can see the earliest date is 2020-01-22 and the latest is 2023-03-09.

Since tt's unfair to compare the numbers from big population state to a small state, I also want to see cases and deaths per populations. I found the population data set on the same github website.

```
# Import population data
uid_lookup_url="https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_
uid=read_csv(uid_lookup_url)
```

```
## Rows: 4321 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.
```

```
# After looking through the columns, exclude unwanted columns %>%
uid<-uid%>% select(-c(Lat, Long_, Combined_Key, iso2,  iso3,  code3,Admin2, UID, FIPS) )

# Add population column to global data
global<-global%>%
  full_join(uid, by=c("Province_State", "Country_Region"))

# Take another look
head(global)
```

```
## # A tibble: 6 x 9
##   Combined_Key Province_State Country_Region   Lat  Long date       cases deaths
##   <chr>        <chr>          <chr>          <dbl> <dbl> <date>     <dbl>  <dbl>
## 1 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-22     0      0
## 2 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-23     0      0
## 3 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-24     0      0
## 4 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-25     0      0
## 5 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-26     0      0
## 6 Afghanistan  <NA>           Afghanistan     33.9  67.7 2020-01-27     0      0
## # i 1 more variable: Population <dbl>
```

Analysis

Get per state and total Country numbers

```
# Get a China data frame
CN<-global%>%filter(Country_Region=="China")

# China by state total cases, deaths, and death per million population
CN_by_state<-CN%>%
  group_by( Country_Region,Province_State, date) %>%
  summarise(cases=sum(cases), deaths=sum(deaths), Population = sum(Population)) %>%
  mutate(death_per_mill = deaths/Population*1000000) %>%
  ungroup()
```

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

```
#Take a look
tail(CN_by_state)
```

```
## # A tibble: 6 x 7
##   Country_Region Province_State date       cases deaths Population
##   <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
## 1 China          Zhejiang       2023-03-05 11848      1   64567588
## 2 China          Zhejiang       2023-03-06 11848      1   64567588
## 3 China          Zhejiang       2023-03-07 11848      1   64567588
## 4 China          Zhejiang       2023-03-08 11848      1   64567588
## 5 China          Zhejiang       2023-03-09 11848      1   64567588
## 6 China          <NA>           NA            NA     NA 1411778724
## # i 1 more variable: death_per_mill <dbl>
```

```
# China Totals
CN_totals<- CN%>%
  group_by( Country_Region, date) %>%
  summarise(cases=sum(cases), deaths=sum(deaths), Population = sum(Population)) %>%
  mutate(death_per_mill = deaths/Population*1000000) %>%
  arrange(death_per_mill) %>%
 ungroup()
```

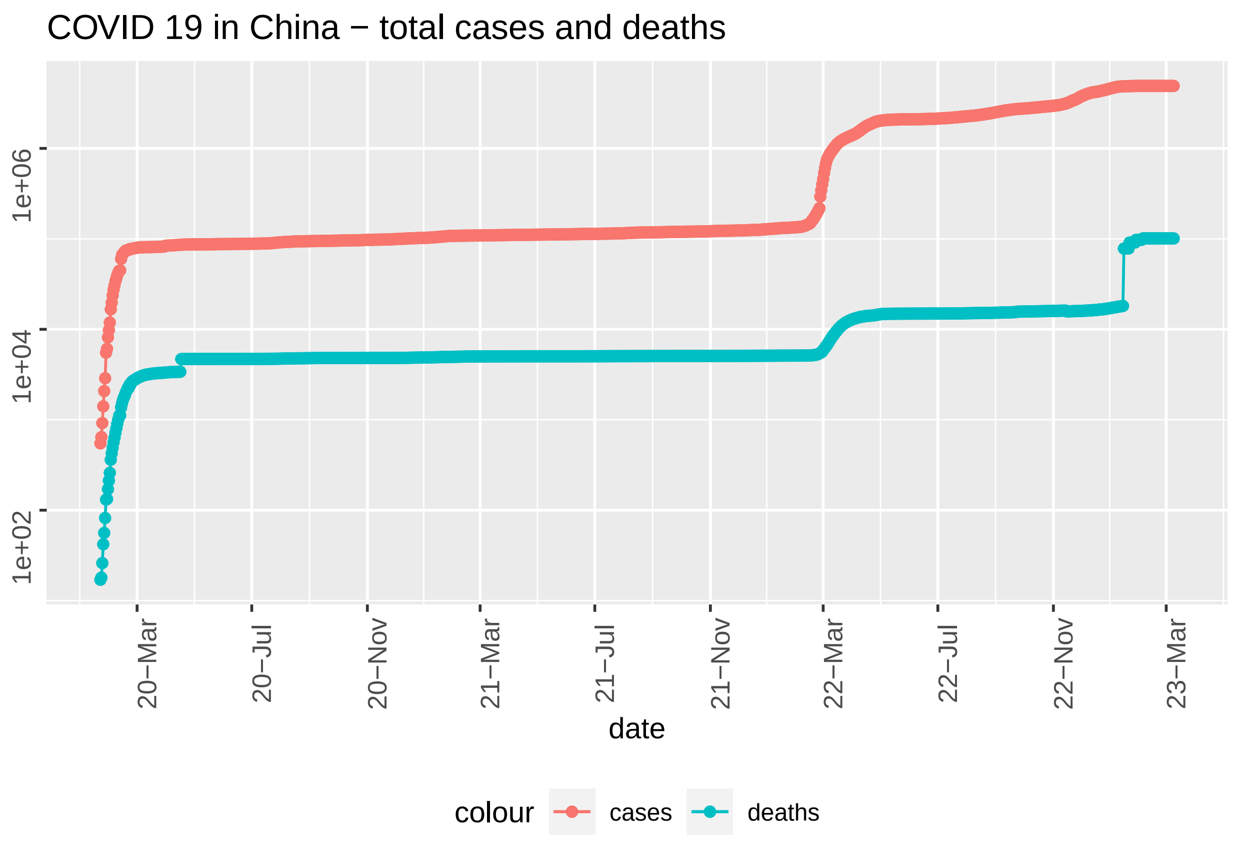
```
## ‘summarise()‘ has grouped output by ’Country_Region’. You can override using
## the ‘.groups‘ argument.
```

```
#Take a look
tail(CN_totals)
```

```
## # A tibble: 6 x 6
##   Country_Region date         cases deaths Population death_per_mill
##   <chr>          <date>       <dbl>  <dbl>      <dbl>          <dbl>
## 1 China          2023-03-05 4903524 101054         NA             NA
## 2 China          2023-03-06 4903524 101055         NA             NA
## 3 China          2023-03-07 4903524 101055         NA             NA
## 4 China          2023-03-08 4903524 101055         NA             NA
## 5 China          2023-03-09 4903524 101056         NA             NA
## 6 China          NA              NA     NA 1411778724             NA
```

Visualization CN totals

```
# Visualize CN totals
options(repr.plot.width=30, repr.plot.height=10)
CN_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() +
   scale_x_date(date_labels = "%y-%b", date_breaks = "4 month") +
  theme(legend.position='bottom', axis.text=element_text(angle=90, size=10)) +
  labs(title="COVID 19 in China - total cases and deaths", y=NULL)
```



How about new cases and new deaths?

When looking at trends, it's good to see how many new cases and new deaths. Let's add those columns

```
# Add new cases columns to China data
CN_by_state<- CN_by_state%>% arrange(Country_Region, Province_State, date) %>%
      mutate(new_cases=cases-lag(cases), new_deaths=deaths-lag(deaths))

CN_totals<- CN_totals%>% arrange(Country_Region, date) %>%
      mutate(new_cases=cases-lag(cases), new_deaths=deaths-lag(deaths))

# Take a look
tail(CN_by_state)
```

```
## # A tibble: 6 x 9
##   Country_Region Province_State date       cases deaths Population
##   <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
## 1 China          Zhejiang       2023-03-05 11848      1   64567588
## 2 China          Zhejiang       2023-03-06 11848      1   64567588
## 3 China          Zhejiang       2023-03-07 11848      1   64567588
## 4 China          Zhejiang       2023-03-08 11848      1   64567588
## 5 China          Zhejiang       2023-03-09 11848      1   64567588
## 6 China          <NA>           NA            NA     NA 1411778724
## # i 3 more variables: death_per_mill <dbl>, new_cases <dbl>, new_deaths <dbl>
```

```
tail(CN_totals)
```

```
## # A tibble: 6 x 8
##   Country_Region date          cases deaths Population death_per_mill new_cases
##   <chr>          <date>        <dbl>  <dbl>      <dbl>          <dbl>     <dbl>
## 1 China          2023-03-05 4903524 101054         NA             NA         0
## 2 China          2023-03-06 4903524 101055         NA             NA         0
## 3 China          2023-03-07 4903524 101055         NA             NA         0
## 4 China          2023-03-08 4903524 101055         NA             NA         0
## 5 China          2023-03-09 4903524 101056         NA             NA         0
## 6 China          NA              NA     NA 1411778724             NA        NA
## # i 1 more variable: new_deaths <dbl>
```

Visualize new cases and deaths in China

```
# Visualize China totals
options(repr.plot.width=30, repr.plot.height=10)
CN_totals %>%
  filter(cases>0) %>%
  ggplot(aes(x=date, y=new_cases))  +
  geom_line(aes(color="new_cases")) +
  geom_point(aes(color="new_cases")) +
  geom_line(aes(y=deaths, color="new_deaths")) +
  geom_point(aes(y=deaths, color="new_deaths")) +
  scale_y_log10() +
  scale_x_date(date_labels = "%y-%b", date_breaks = "4 month") +
  theme(legend.position='bottom', axis.text=element_text(angle=90, size=10)) +
  labs(title="COVID 19 in China - new cases and deaths", y=NULL)
```

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

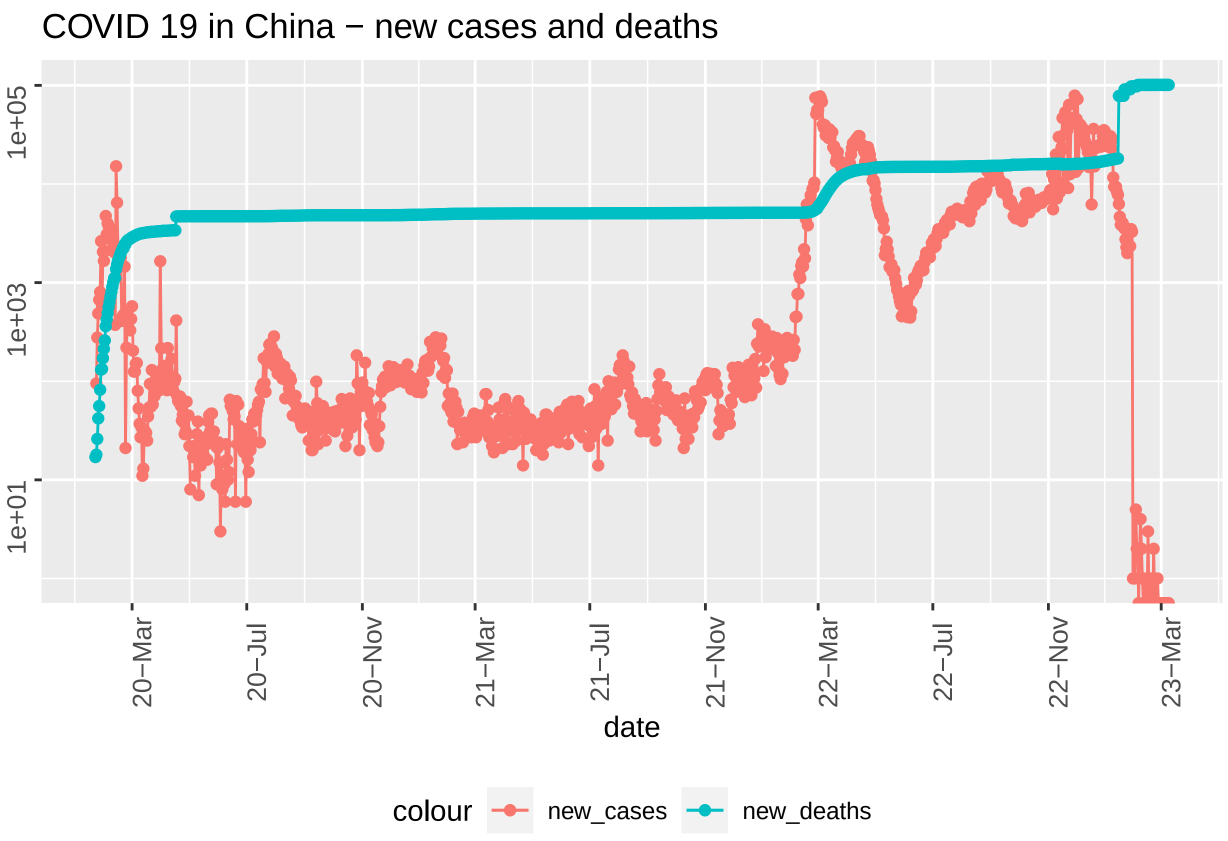
```
## Warning: Transformation introduced infinite values in continuous y-axis
```

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

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

```
## Warning: Removed 1 row containing missing values (‘geom_line()‘).
```

```
## Warning: Removed 3 rows containing missing values (‘geom_point()‘).
```



## What are the worst and best states in China?

CN by states

Let's see which states are best/worst (in term of death/population)

```
CN_state_totals <- CN_by_state %>%
   group_by(Province_State) %>%
   summarize(cases=max(cases),
       deaths= max(deaths),
       Population=max(Population),
       cases_per_thou=1000*cases/Population,
       deaths_per_thou=1000*deaths/Population)
CN_state_totals %>% slice_min(deaths_per_thou,n=10)
```

```
## # A tibble: 10 x 6
##    Province_State cases deaths Population cases_per_thou deaths_per_thou
##    <chr>          <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
##  1 Jiangsu         5075      0   84748016         0.0599       0
##  2 Ningxia         1276      0    7202654         0.177        0
##  3 Qinghai          782      0    5923957         0.132        0
##  4 Tibet           1647      0    3648100         0.451        0
##  5 Zhejiang       11848      1   64567588         0.183        0.0000155
##  6 Shanxi          7167      1   34915616         0.205        0.0000286
##  7 Guangxi        13371      2   50126804         0.267        0.0000399
##  8 Inner Mongolia  8847      1   24049155         0.368        0.0000416
##  9 Jiangxi          3423      2   45188635         0.0757       0.0000443
## 10 Liaoning         3547      2   42591407         0.0833       0.0000470
```

```
CN_state_totals %>% slice_max(deaths_per_thou,n=10)
```

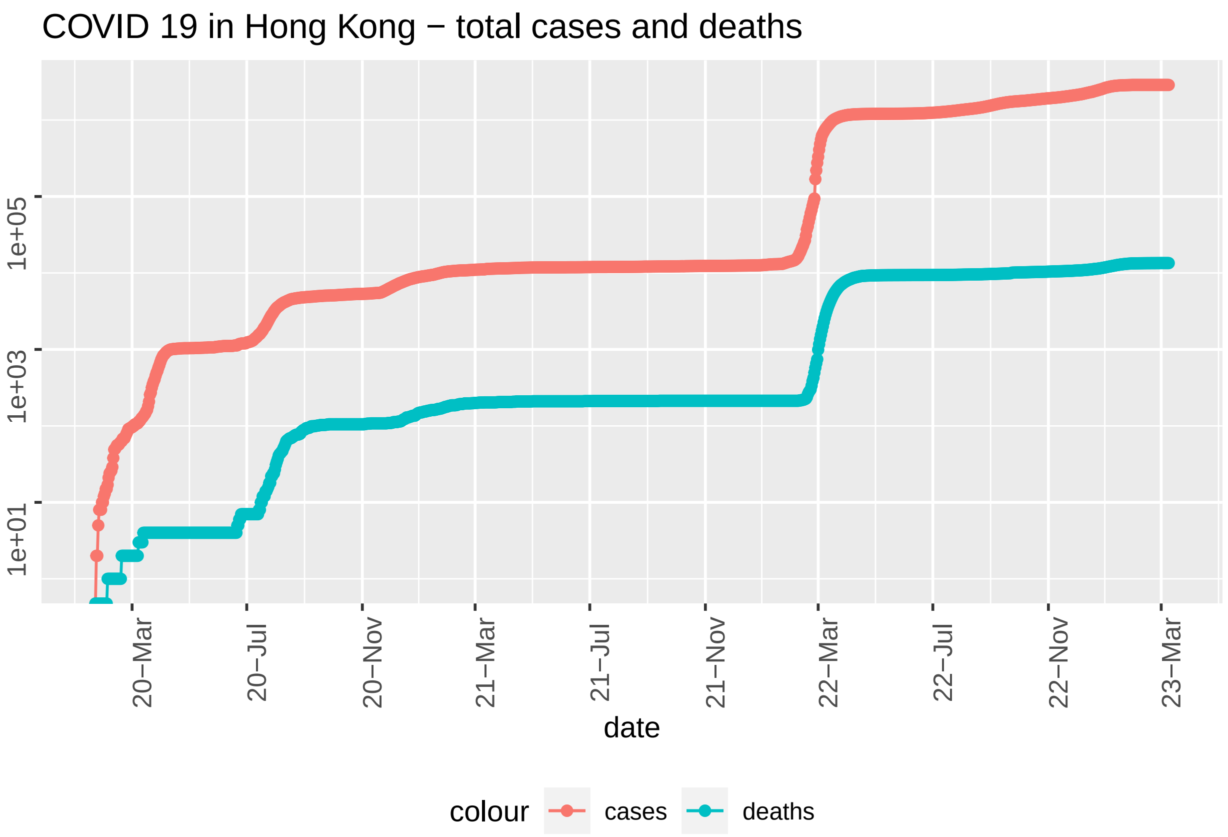
```
## # A tibble: 10 x 6
##    Province_State   cases deaths Population cases_per_thou deaths_per_thou
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
##  1 Hong Kong      2876106  13467    7496988        384.           1.80
##  2 Macau             3547    121     649342          5.46         0.186
##  3 Hubei            72131   4515   57752557          1.25         0.0782
##  4 Shanghai         67040    595   24870895          2.70         0.0239
##  5 Beijing          40774     20   21893095          1.86         0.000914
##  6 Hainan           10483      6   10081232          1.04         0.000595
##  7 Heilongjiang      6603     18   31850088          0.207        0.000565
##  8 Chongqing        14715     11   32054159          0.459        0.000343
##  9 Henan             9948     23   99365519          0.100        0.000231
## 10 Tianjin           4392      3   13866009          0.317        0.000216
```

visualize state of interest

I want to visualize the top 3 worst states

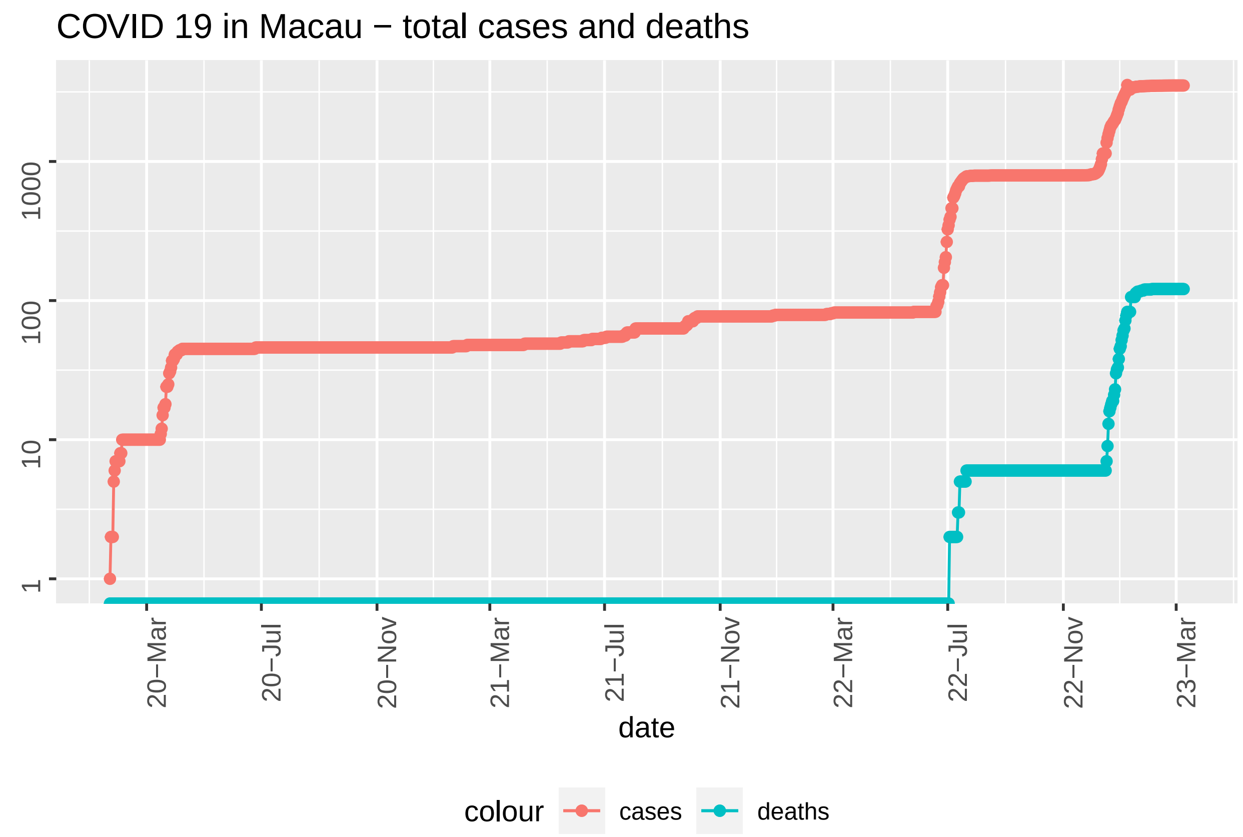
```
state<- "Hong Kong"
CN_by_state %>%
  filter(Province_State==state) %>%
  ggplot(aes(x=date, y=cases))  +
  geom_line(aes(color="cases")) +
  geom_point(aes(color="cases")) +
  geom_line(aes(y=deaths, color="deaths")) +
  geom_point(aes(y=deaths, color="deaths")) +
  scale_y_log10() +
  scale_x_date(date_labels = "%y-%b", date_breaks = "4 month") +
  theme(legend.position='bottom', axis.text=element_text(angle=90, size=10)) +
  labs(title=str_c("COVID 19 in ", state," - total cases and deaths"), y=NULL)
```

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

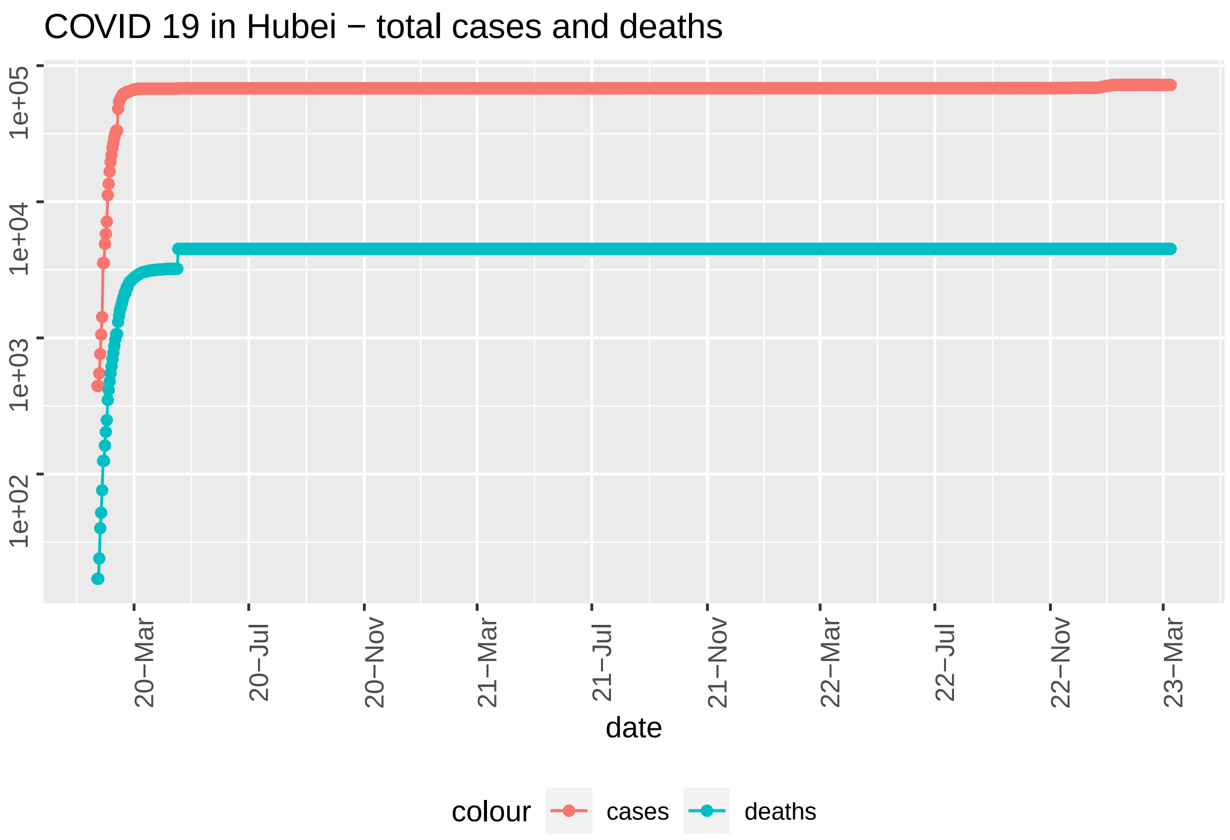


```r
state<- "Macau"
CN_by_state %>%
  filter(Province_State==state) %>%
  ggplot(aes(x=date, y=cases))  +
  geom_line(aes(color="cases")) +
  geom_point(aes(color="cases")) +
  geom_line(aes(y=deaths, color="deaths")) +
  geom_point(aes(y=deaths, color="deaths")) +
  scale_y_log10() +
  scale_x_date(date_labels = "%y-%b", date_breaks = "4 month") +
  theme(legend.position='bottom', axis.text=element_text(angle=90, size=10)) +
  labs(title=str_c("COVID 19 in ", state," - total cases and deaths"), y=NULL)
```

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



```
state<- "Hubei"
CN_by_state %>%
  filter(Province_State==state) %>%
  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() +
  scale_x_date(date_labels = "%y-%b", date_breaks = "4 month") +
  theme(legend.position='bottom', axis.text=element_text(angle=90, size=10)) +
  labs(title=str_c("COVID 19 in ", state," - total cases and deaths"), y=NULL)
```


COVID 19 in Hubei − total cases and deaths
1e+05
1e+04
1e+03
1e+02
20−Mar
20−Jul
20−Nov
21−Mar
21−Jul
21−Nov
22−Mar
22−Jul
22−Nov
23−Mar
date
colour
cases
deaths