# Covid 19 Data

12/2021

## Data Source and biases

The data used in this exploration was obtained from a data repository that is managed by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE). The university aggregates data from many credible sources, such as the World Health Organization (WHO), European Centre for Disease Prevention and Control (ECDC), and the US CDC. The repository contains data not only on the United States, but also on most countries around the world. They used dozens of sources to get covid data on as many countries as possible.

The data sets in the repository track Covid19 statistics , such number of cases, deaths, and recoveries for many countries (and province/state if required).

Since the data is gathered by a very reputable university, I trust that the data is reliable. However, it may still contain bias. Johns Hopkins University is not the ones who are personally collecting data. Instead, they are only aggregating data that is reported by each country. If the original data contains bias, then that bias will also transfer to the aggregated data.

Covid19 data is difficult to gather accurately, because it mainly comes from health care facilities that test people to see if they are infected with the disease. The problem with Covid19 is that symptoms take up to two weeks to develop, and there are also people who can get infected but show no symptoms. These people should be counted together with the sick, because they are also capable of spreading the disease. But since they don't get checked, it becomes impossible to include this subset of people as part of the data. Covid19 testing is also not 100% effective. False Positive and False Negative tests are possible, so the data that is generated is not completely reliable. Furthermore, there are some countries who are not fully transparent with the world about their current Covid19 statistics. They may want to hide, or at least reduce, the number of cases and/or deaths related to Covid19 for political reasons, such as appearing to the public that they are handling to situation well. These are some of the reasons why the data in this repository cannot be viewed as completely accurate. Instead, they should be seen as a good approximation of the true Covid19 statistics of each country.

## Import libraries

Below are the libraries that I will use in this exploration.

```r
library(tidyverse)
library(ggplot2)
library(lubridate) # For use in converting date/time variables
```

## Import data

Import all the datasets from the Johns Hopkins Github repository:

```r
## Import the csv files. Note that they all begin the same way

url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names <- c("time_series_covid19_confirmed_US.csv",
                "time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_US.csv",
                "time_series_covid19_deaths_global.csv",
                "time_series_covid19_recovered_global.csv")
urls <- str_c(url_in, file_names)

us_cases <- read_csv(urls[1])
global_cases <- read_csv(urls[2])
us_deaths <- read_csv(urls[3])
global_deaths <- read_csv(urls[4])
global_recovered <- read_csv(urls[5])
```

## Tidy data

Tidy the data by re-arranging the table in a more meaningful way using a pivot table. I won't be using the "Province/State", "Lat", or "Long" data columns, so I will be removing them here as well.

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

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

# Tidy global_recovered:
global_recovered <- global_recovered %>%
  pivot_longer(cols = -c('Province/State', 'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "recovery") %>%
  select(-c('Province/State', Lat, Long))

# Note that us_deaths contains a population column that is not present in us_cases. Do a full join to c
#US <- us_cases %>%
#  full_join(us_deaths)
```

I live in South Africa, so for this analysis I want to filter out data that is related to my country.

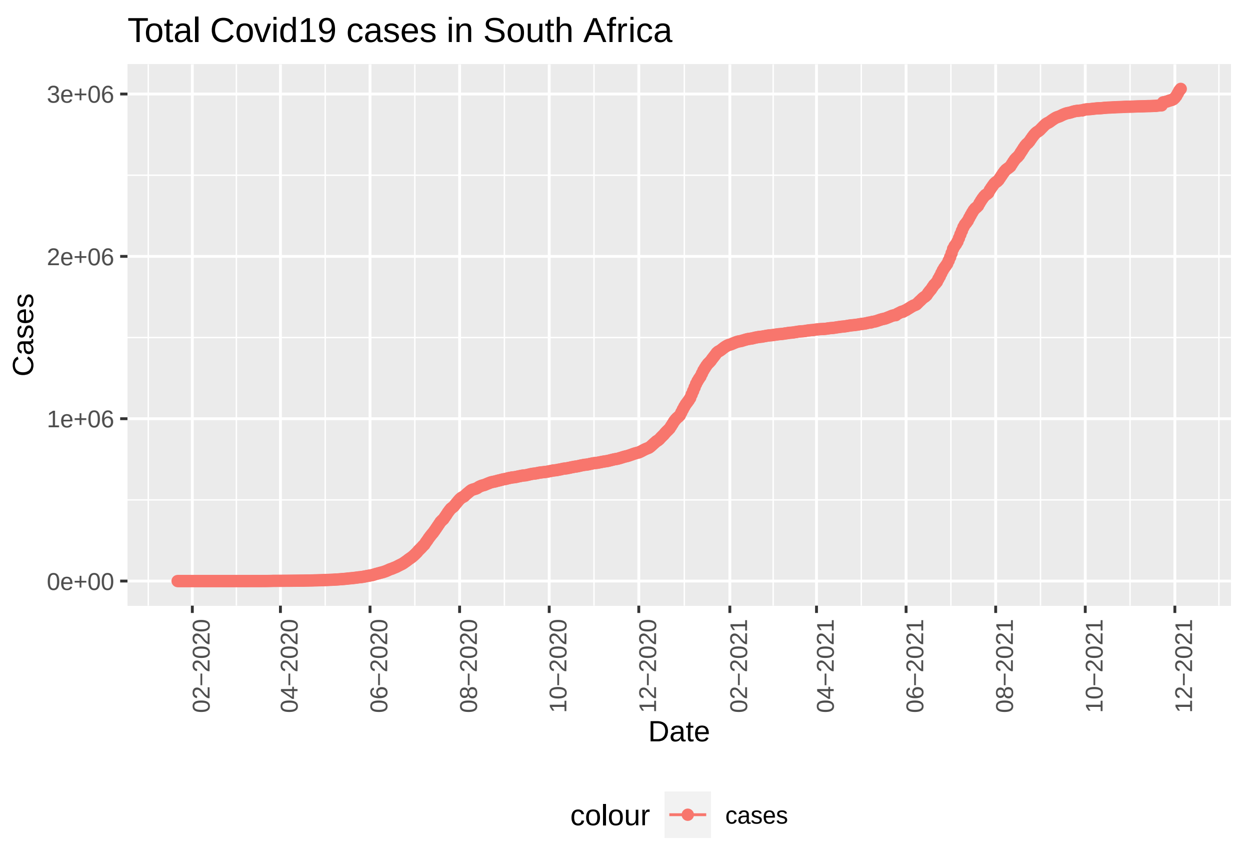
```
SA_cases <- global_cases %>%
  filter(`Country/Region` == 'South Africa') %>%
  mutate(date = mdy(date)) # Change date into date object

SA_deaths <- global_deaths %>%
  filter(`Country/Region` == 'South Africa')  %>%
  mutate(date = mdy(date)) # Change date into date object

SA_recovered <- global_recovered %>%
  filter(`Country/Region` == 'South Africa')  %>%
  mutate(date = mdy(date)) # Change date into date object

SA_data <- SA_cases %>% # Combine the above datasets into a single larger table
  full_join(SA_deaths) %>%
  full_join(SA_recovered)

SA_data
```

```
## # A tibble: 684 x 5
##    `Country/Region` date       cases deaths recovery
##    <chr>            <date>     <dbl>  <dbl>    <dbl>
##  1 South Africa     2020-01-22     0      0        0
##  2 South Africa     2020-01-23     0      0        0
##  3 South Africa     2020-01-24     0      0        0
##  4 South Africa     2020-01-25     0      0        0
##  5 South Africa     2020-01-26     0      0        0
##  6 South Africa     2020-01-27     0      0        0
##  7 South Africa     2020-01-28     0      0        0
##  8 South Africa     2020-01-29     0      0        0
##  9 South Africa     2020-01-30     0      0        0
## 10 South Africa     2020-01-31     0      0        0
## # ... with 674 more rows
```

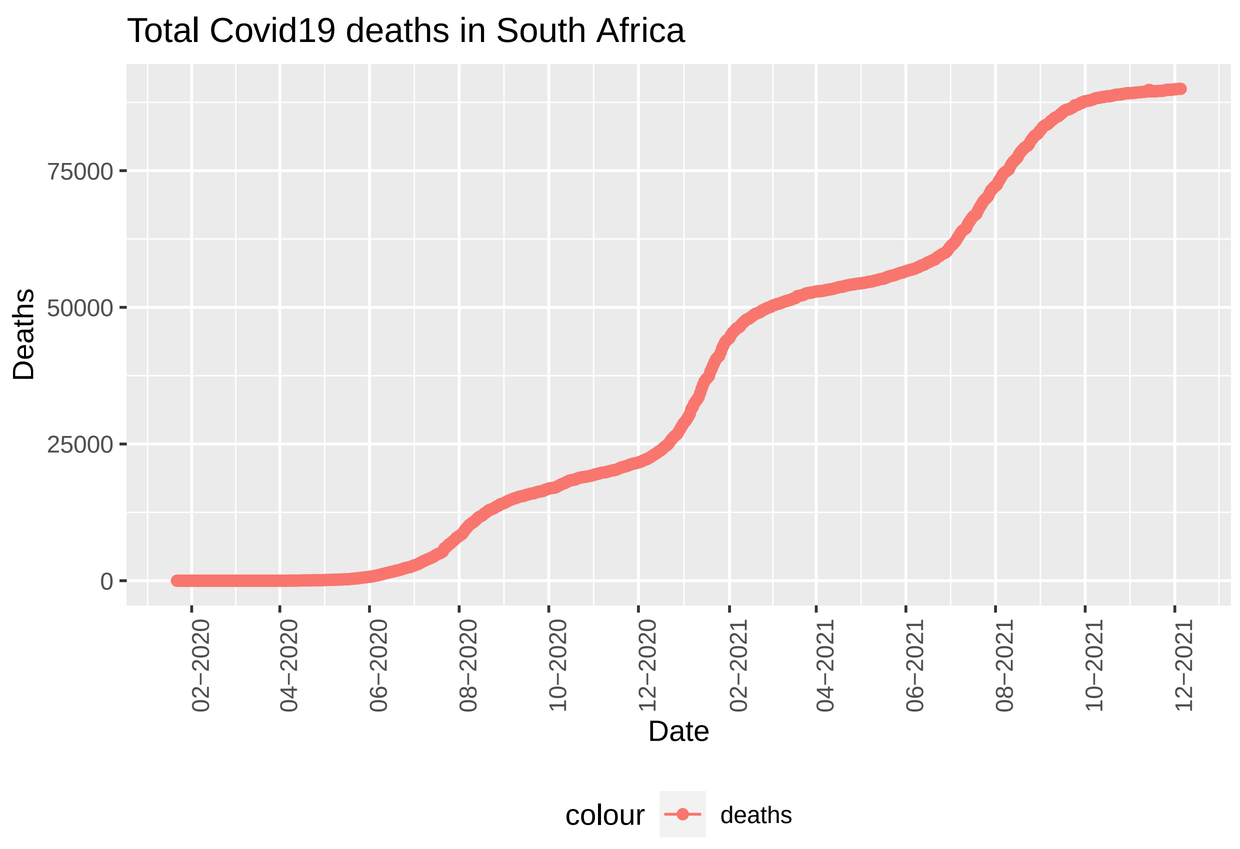
## Visualize the data

Below are the plots of the cases, deaths, and recoveries for South Africa:

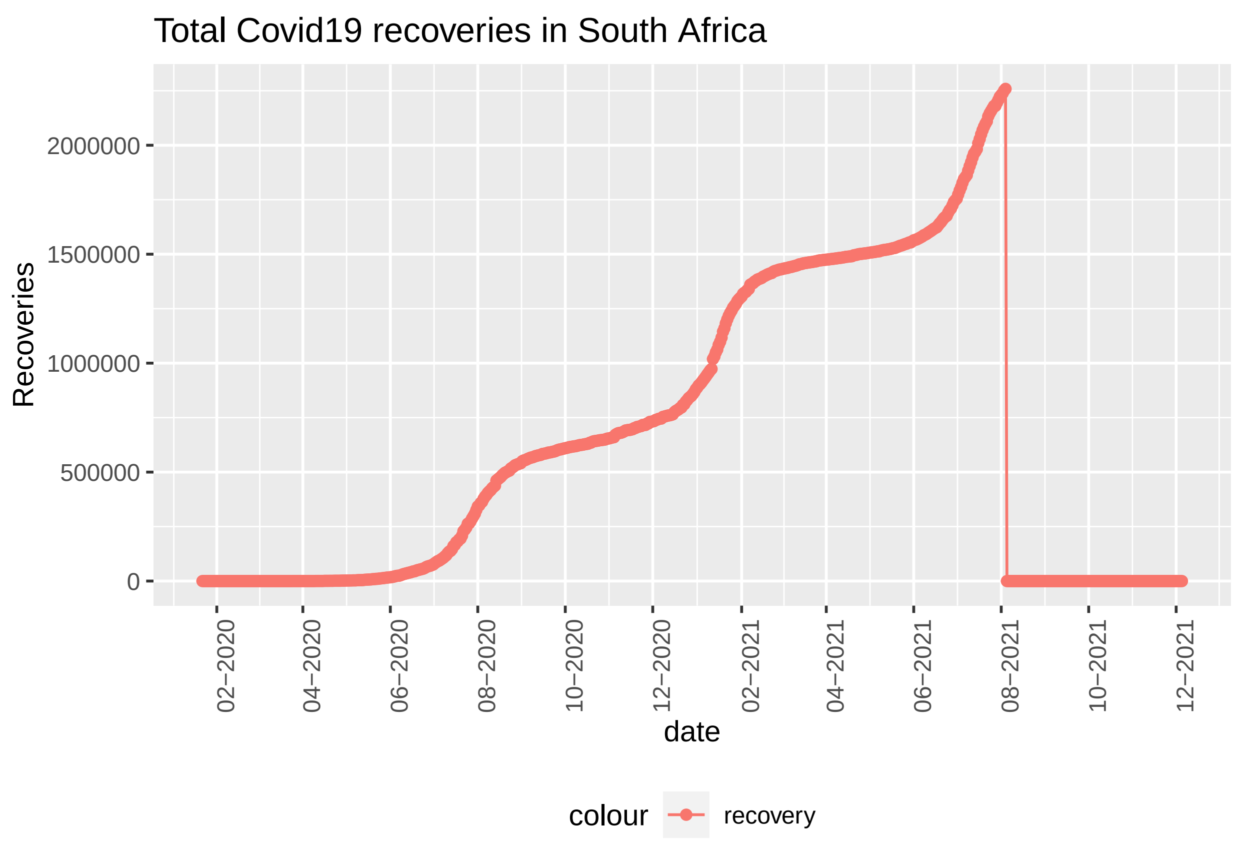
```
SA_data %>%
  #filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  scale_x_date(date_labels = "%m-%Y", date_breaks = '2 month') +
  labs(title = "Total Covid19 cases in South Africa", x='Date', y='Cases')
```



```
SA_data %>%
  #filter(cases > 0) %>%
  ggplot(aes(x = date, y = deaths)) +
  geom_line(aes(color = "deaths")) +
  geom_point(aes(color = "deaths")) +
  scale_x_date(date_labels = "%m-%Y", date_breaks = '2 month') +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "Total Covid19 deaths in South Africa", x='Date', y='Deaths')
```



```
SA_data %>%
  #filter(cases > 0) %>%
  ggplot(aes(x = date, y = recovery)) +
  geom_line(aes(color = "recovery")) +
  geom_point(aes(color = "recovery")) +
  scale_x_date(date_labels = "%m-%Y", date_breaks = '2 month') +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "Total Covid19 recoveries in South Africa", y='Recoveries')
```



## Observations

I found it interesting that all three plots (cases, deaths, and recoveries) show the same general shape. They start of level, then rises exponentially for two months and then stays level for another four months, before repeating again. In South Africa, there appears to have been three waves of high-volume infections.

There seems to be an anomaly in the recovery data set. The data showed that there were no recoveries past August 2021. This doesn't make sense, so my personal theory is that the data sources that Johns Hopkins used no longer reported recovery data.

This observation makes sense, because as more people get infected, the death rate would also rise roughly proportionally. As more people get infected, more people can also recover from it. It makes sense that the shape of all three graphs correspond with each other.

## Monthly cases

I want to see how the data looks when I aggregate the cases for each month.

```
SA_data_dates <- SA_data %>%
  filter(cases > 0) %>%
  mutate(month = month(date)) %>%
  mutate(year = year(date)) %>%
  group_by(month, year) %>%
  summarize(total = sum(cases))
```

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

```
unique(SA_data_dates$month)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

```
unique(SA_data_dates$year)
```

```
## [1] 2021 2020
```

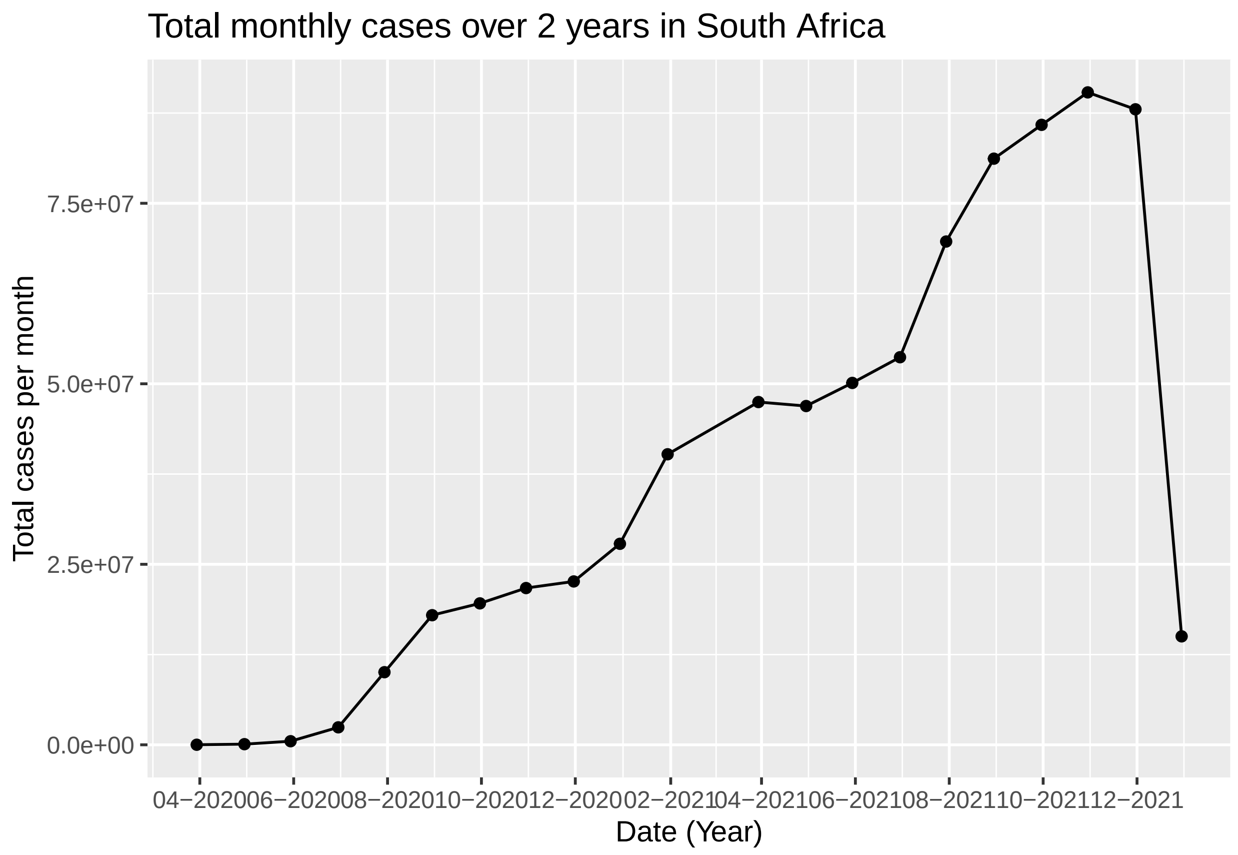
```
length(SA_data_dates$total)
```

```
## [1] 22
```

```
SA_data_dates
```

```
## # A tibble: 22 x 3
## # Groups:   month [12]
##    month  year    total
##    <dbl> <dbl>    <dbl>
##  1     1  2021 40240702
##  2     2  2021 41711243
##  3     3  2020    10173
##  4     3  2021 47472251
##  5     4  2020    88044
##  6     4  2021 46922791
##  7     5  2020   497374
##  8     5  2021 50121593
##  9     6  2020  2421471
## 10     6  2021 53680896
## # ... with 12 more rows
```

```
SA_data_dates %>%
  mutate(MonthYear = as.Date(paste0("30-", month,"-", year),"%d-%m-%Y")) %>%
  ggplot(aes(x=MonthYear, y=total, group = 1)) +
  geom_line() +
  geom_point() +
  scale_x_date(date_labels = "%m-%Y", date_breaks = '2 months') +
  labs(x = "Date (Year)", y = "Total cases per month", title = "Total monthly cases over 2 years in Sou
```



```
SA_data_dates
```

```
## # A tibble: 22 x 3
## # Groups:   month [12]
##    month  year    total
##    <dbl> <dbl>    <dbl>
##  1     1  2021 40240702
##  2     2  2021 41711243
##  3     3  2020    10173
##  4     3  2021 47472251
##  5     4  2020    88044
##  6     4  2021 46922791
##  7     5  2020   497374
##  8     5  2021 50121593
##  9     6  2020  2421471
## 10     6  2021 53680896
## # ... with 12 more rows
```

## Model

Looking at the shape of the monthly cases, it appears to increase almost linearly with time. For that reason, I want to correlate the total number of monthly cases against time. Therefore, I created a linear model that is a function of both the month and the year:

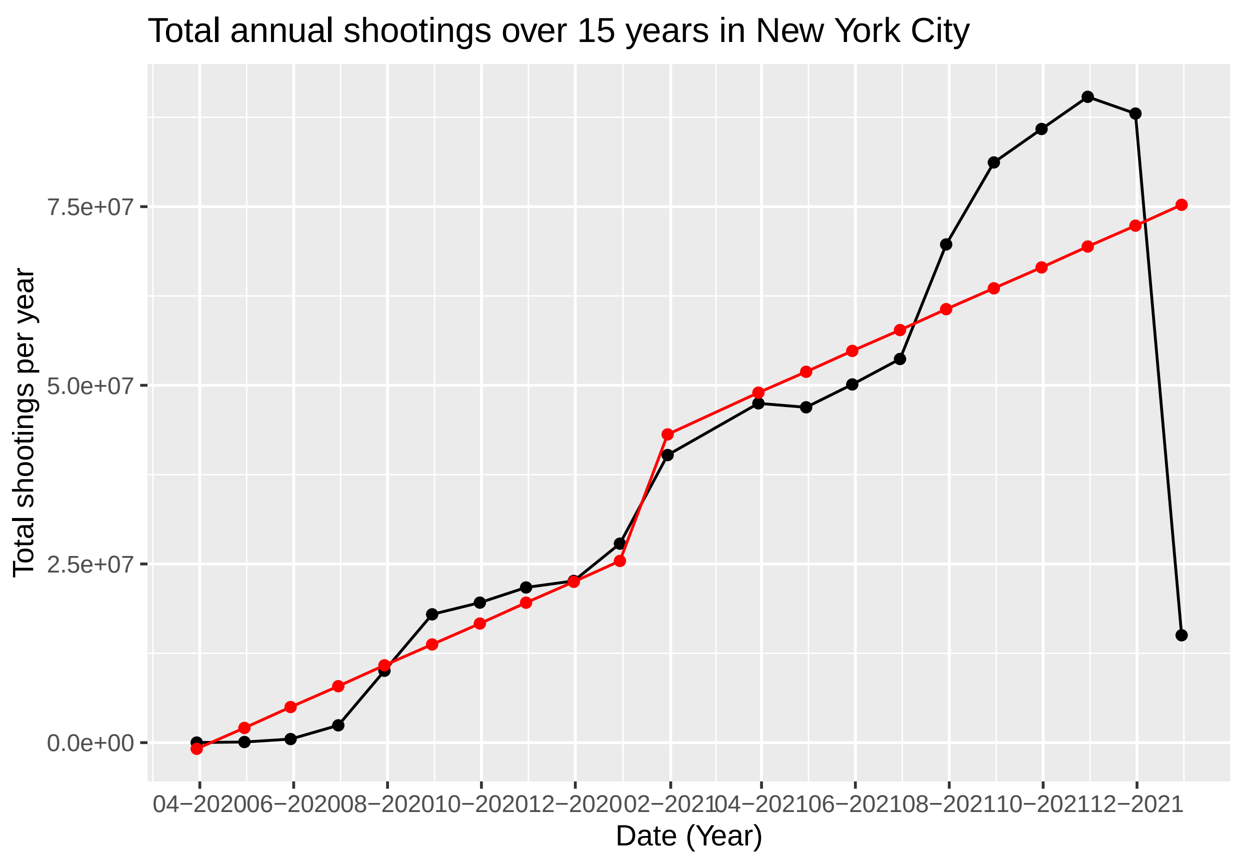
```
mod <- lm(total ~ year + month, data = SA_data_dates)
summary(mod)
```

```
##
## Call:
## lm(formula = total ~ year + month, data = SA_data_dates)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -60234265  -4266060   -318279   3891353  20947064
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.007e+11  1.458e+10  -6.907 1.38e-06 ***
## year         4.983e+07  7.213e+06   6.909 1.38e-06 ***
## month        2.921e+06  1.109e+06   2.635   0.0163 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16650000 on 19 degrees of freedom
## Multiple R-squared:  0.7257, Adjusted R-squared:  0.6968
## F-statistic: 25.13 on 2 and 19 DF,  p-value: 4.606e-06
```

```
SA_data_dates$pred <- predict(mod)  ## Add predictions to a new column
```

```
SA_data_dates %>%
  mutate(MonthYear = as.Date(paste0("30-", month,"-", year),"%d-%m-%Y")) %>%
  ggplot() +
  geom_line(aes(x=MonthYear, y=total)) +
  geom_point(aes(x=MonthYear, y=total)) +
  geom_line(aes(x=MonthYear, y=pred), color = 'red') +
  geom_point(aes(x=MonthYear, y=pred), color = 'red') +
  scale_x_date(date_labels = "%m-%Y", date_breaks = '2 months') +
  labs(x = "Date (Year)", y = "Total shootings per year", title = "Total annual shootings over 15 years
```

```
## 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 1 rows containing missing values (geom_point).
```



## Conclusions

The model performed fairly well during the first 18 months, but failed to predict the recent exponential increase in cases after July 2021. Furthermore, the final data point at the end of 2021 is unexpectedly low. It appears to be the only outlier in the data.

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_South Africa.1252  LC_CTYPE=English_South Africa.1252
## [3] LC_MONETARY=English_South Africa.1252 LC_NUMERIC=C
## [5] LC_TIME=English_South Africa.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.7.10 forcats_0.5.1    stringr_1.4.0    dplyr_1.0.7
##  [5] purrr_0.3.4      readr_2.0.2      tidyr_1.1.4      tibble_3.1.4
##  [9] ggplot2_3.3.5    tidyverse_1.3.1
```

```
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7       assertthat_0.2.1 digest_0.6.27    utf8_1.2.2
##  [5] R6_2.5.1         cellranger_1.1.0 backports_1.2.1  reprex_2.0.1
##  [9] evaluate_0.14    highr_0.9        httr_1.4.2       pillar_1.6.3
## [13] rlang_0.4.11     curl_4.3.2       readxl_1.3.1     rstudioapi_0.13
## [17] rmarkdown_2.11   labeling_0.4.2   bit_4.0.4        munsell_0.5.0
## [21] broom_0.7.9      compiler_4.1.1   modelr_0.1.8     xfun_0.26
## [25] pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.1 fansi_0.5.0
## [29] crayon_1.4.1     tzdb_0.1.2       dbplyr_2.1.1     withr_2.4.2
## [33] grid_4.1.1       jsonlite_1.7.2   gtable_0.3.0     lifecycle_1.0.1
## [37] DBI_1.1.1        magrittr_2.0.1   scales_1.1.1     cli_3.0.1
## [41] stringi_1.7.4    vroom_1.5.5      farver_2.1.0     fs_1.5.0
## [45] xml2_1.3.2       ellipsis_0.3.2   generics_0.1.0   vctrs_0.3.8
## [49] tools_4.1.1      bit64_4.0.5      glue_1.4.2       hms_1.1.1
## [53] parallel_4.1.1   fastmap_1.1.0    yaml_2.2.1       colorspace_2.0-2
## [57] rvest_1.0.1      knitr_1.34       haven_2.4.3
```