# COVID-19_Analysis

Solomon

2024-08-10

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

##In this report, R is used to import, tidy, transform, visualize, and model COVID-19 data. The details of the process are given in the steps below.

```r
library(plotly)
```

### Step 1 Import R libraries and set up environment

```
## Warning: package 'plotly' was built under R version 4.3.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(tidyverse)
```

```
## Warning: package ’tidyverse’ was built under R version 4.3.3

## Warning: package ’tidyr’ was built under R version 4.3.3

## Warning: package ’readr’ was built under R version 4.3.3

## Warning: package ’dplyr’ was built under R version 4.3.3

## Warning: package ’stringr’ was built under R version 4.3.3

## -- 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 lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks plotly::filter(), 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
```

```r
library(lubridate)
options(warn=-1)
options(dplyr.summarise.inform = FALSE)
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_knit$set(root.dir = getwd())
```

#### Step 2 Download and import COVID-19 source data files. Tidyverse package used to Read the CSV directly from the data sources.

```r
url_in <-
  'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti
filenames <- c('time_series_covid19_confirmed_global.csv', 'time_series_covid19_deaths_global.csv', 'ti

confirmed_global <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_cov
```

```
## 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.
```

```r
deaths_global <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
```

```
## 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.
```

```r
confirmed_us <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_1
```

```
## 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.
```

```r
deaths_us <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_da
```

```
## 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.
```

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

```
## 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.
```

```r
urls <- str_c(url_in, filenames)
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.
```

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

```
## 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 <- read_csv(urls[4])
```

```
## 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.
```

```
# Read .csv file for US COVID-19 vaccinations
url_in <-
  'https://covid.ourworldindata.org/data/vaccinations/us_state_vaccinations.csv'
US_vaccinations <- read_csv(url_in) %>%
  select(-c(total_vaccinations, total_distributed, people_vaccinated, people_fully_vaccinated,
            daily_vaccinations_raw, daily_vaccinations, daily_vaccinations_per_million,
            share_doses_used, total_boosters)) %>%
  rename(Province_State = 'location')
```

```
## Rows: 54628 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): location
## dbl  (14): total_vaccinations, total_distributed, people_vaccinated, people_...
## date  (1): date
##
## 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.
```

```
head(confirmed_global)
```

```
## # 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>, ...
```

```
sort(colnames(confirmed_global), decreasing = TRUE)
```

```
##    [1] "Province/State" "Long"           "Lat"            "Country/Region"
##    [5] "9/9/22"         "9/9/21"         "9/9/20"         "9/8/22"
##    [9] "9/8/21"         "9/8/20"         "9/7/22"         "9/7/21"
##   [13] "9/7/20"         "9/6/22"         "9/6/21"         "9/6/20"
##   [17] "9/5/22"         "9/5/21"         "9/5/20"         "9/4/22"
##   [21] "9/4/21"         "9/4/20"         "9/30/22"        "9/30/21"
##   [25] "9/30/20"        "9/3/22"         "9/3/21"         "9/3/20"
##   [29] "9/29/22"        "9/29/21"        "9/29/20"        "9/28/22"
##   [33] "9/28/21"        "9/28/20"        "9/27/22"        "9/27/21"
##   [37] "9/27/20"        "9/26/22"        "9/26/21"        "9/26/20"
##   [41] "9/25/22"        "9/25/21"        "9/25/20"        "9/24/22"
##   [45] "9/24/21"        "9/24/20"        "9/23/22"        "9/23/21"
##   [49] "9/23/20"        "9/22/22"        "9/22/21"        "9/22/20"
##   [53] "9/21/22"        "9/21/21"        "9/21/20"        "9/20/22"
##   [57] "9/20/21"        "9/20/20"        "9/2/22"         "9/2/21"
##   [61] "9/2/20"         "9/19/22"        "9/19/21"        "9/19/20"
##   [65] "9/18/22"        "9/18/21"        "9/18/20"        "9/17/22"
##   [69] "9/17/21"        "9/17/20"        "9/16/22"        "9/16/21"
##   [73] "9/16/20"        "9/15/22"        "9/15/21"        "9/15/20"
##   [77] "9/14/22"        "9/14/21"        "9/14/20"        "9/13/22"
##   [81] "9/13/21"        "9/13/20"        "9/12/22"        "9/12/21"
##   [85] "9/12/20"        "9/11/22"        "9/11/21"        "9/11/20"
##   [89] "9/10/22"        "9/10/21"        "9/10/20"        "9/1/22"
##   [93] "9/1/21"         "9/1/20"         "8/9/22"         "8/9/21"
##   [97] "8/9/20"         "8/8/22"         "8/8/21"         "8/8/20"
##  [101] "8/7/22"         "8/7/21"         "8/7/20"         "8/6/22"
##  [105] "8/6/21"         "8/6/20"         "8/5/22"         "8/5/21"
##  [109] "8/5/20"         "8/4/22"         "8/4/21"         "8/4/20"
##  [113] "8/31/22"        "8/31/21"        "8/31/20"        "8/30/22"
##  [117] "8/30/21"        "8/30/20"        "8/3/22"         "8/3/21"
##  [121] "8/3/20"         "8/29/22"        "8/29/21"        "8/29/20"
##  [125] "8/28/22"        "8/28/21"        "8/28/20"        "8/27/22"
##  [129] "8/27/21"        "8/27/20"        "8/26/22"        "8/26/21"
##  [133] "8/26/20"        "8/25/22"        "8/25/21"        "8/25/20"
##  [137] "8/24/22"        "8/24/21"        "8/24/20"        "8/23/22"
##  [141] "8/23/21"        "8/23/20"        "8/22/22"        "8/22/21"
##  [145] "8/22/20"        "8/21/22"        "8/21/21"        "8/21/20"
##  [149] "8/20/22"        "8/20/21"        "8/20/20"        "8/2/22"
##  [153] "8/2/21"         "8/2/20"         "8/19/22"        "8/19/21"
##  [157] "8/19/20"        "8/18/22"        "8/18/21"        "8/18/20"
##  [161] "8/17/22"        "8/17/21"        "8/17/20"        "8/16/22"
##  [165] "8/16/21"        "8/16/20"        "8/15/22"        "8/15/21"
##  [169] "8/15/20"        "8/14/22"        "8/14/21"        "8/14/20"
##  [173] "8/13/22"        "8/13/21"        "8/13/20"        "8/12/22"
##  [177] "8/12/21"        "8/12/20"        "8/11/22"        "8/11/21"
##  [181] "8/11/20"        "8/10/22"        "8/10/21"        "8/10/20"
##  [185] "8/1/22"         "8/1/21"         "8/1/20"         "7/9/22"
##  [189] "7/9/21"         "7/9/20"         "7/8/22"         "7/8/21"
##  [193] "7/8/20"         "7/7/22"         "7/7/21"         "7/7/20"
##  [197] "7/6/22"         "7/6/21"         "7/6/20"         "7/5/22"
##  [201] "7/5/21"         "7/5/20"         "7/4/22"         "7/4/21"
##  [205] "7/4/20"         "7/31/22"        "7/31/21"        "7/31/20"
##  [209] "7/30/22"        "7/30/21"        "7/30/20"        "7/3/22"
##  [213] "7/3/21"         "7/3/20"         "7/29/22"        "7/29/21"
##  [217] "7/29/20"        "7/28/22"        "7/28/21"        "7/28/20"
##  [221] "7/27/22"        "7/27/21"        "7/27/20"        "7/26/22"
##  [225] "7/26/21"        "7/26/20"        "7/25/22"        "7/25/21"
##  [229] "7/25/20"        "7/24/22"        "7/24/21"        "7/24/20"
##  [233] "7/23/22"        "7/23/21"        "7/23/20"        "7/22/22"
##  [237] "7/22/21"        "7/22/20"        "7/21/22"        "7/21/21"
##  [241] "7/21/20"        "7/20/22"        "7/20/21"        "7/20/20"
##  [245] "7/2/22"         "7/2/21"         "7/2/20"         "7/19/22"
##  [249] "7/19/21"        "7/19/20"        "7/18/22"        "7/18/21"
##  [253] "7/18/20"        "7/17/22"        "7/17/21"        "7/17/20"
##  [257] "7/16/22"        "7/16/21"        "7/16/20"        "7/15/22"
##  [261] "7/15/21"        "7/15/20"        "7/14/22"        "7/14/21"
##  [265] "7/14/20"        "7/13/22"        "7/13/21"        "7/13/20"
##  [269] "7/12/22"        "7/12/21"        "7/12/20"        "7/11/22"
##  [273] "7/11/21"        "7/11/20"        "7/10/22"        "7/10/21"
##  [277] "7/10/20"        "7/1/22"         "7/1/21"         "7/1/20"
##  [281] "6/9/22"         "6/9/21"         "6/9/20"         "6/8/22"
##  [285] "6/8/21"         "6/8/20"         "6/7/22"         "6/7/21"
##  [289] "6/7/20"         "6/6/22"         "6/6/21"         "6/6/20"
##  [293] "6/5/22"         "6/5/21"         "6/5/20"         "6/4/22"
##  [297] "6/4/21"         "6/4/20"         "6/30/22"        "6/30/21"
##  [301] "6/30/20"        "6/3/22"         "6/3/21"         "6/3/20"
##  [305] "6/29/22"         "6/29/21"         "6/29/20"         "6/28/22"
##  [309] "6/28/21"         "6/28/20"         "6/27/22"         "6/27/21"
##  [313] "6/27/20"         "6/26/22"         "6/26/21"         "6/26/20"
##  [317] "6/25/22"         "6/25/21"         "6/25/20"         "6/24/22"
##  [321] "6/24/21"         "6/24/20"         "6/23/22"         "6/23/21"
##  [325] "6/23/20"         "6/22/22"         "6/22/21"         "6/22/20"
##  [329] "6/21/22"         "6/21/21"         "6/21/20"         "6/20/22"
##  [333] "6/20/21"         "6/20/20"         "6/2/22"          "6/2/21"
##  [337] "6/2/20"          "6/19/22"         "6/19/21"         "6/19/20"
##  [341] "6/18/22"         "6/18/21"         "6/18/20"         "6/17/22"
##  [345] "6/17/21"         "6/17/20"         "6/16/22"         "6/16/21"
##  [349] "6/16/20"         "6/15/22"         "6/15/21"         "6/15/20"
##  [353] "6/14/22"         "6/14/21"         "6/14/20"         "6/13/22"
##  [357] "6/13/21"         "6/13/20"         "6/12/22"         "6/12/21"
##  [361] "6/12/20"         "6/11/22"         "6/11/21"         "6/11/20"
##  [365] "6/10/22"         "6/10/21"         "6/10/20"         "6/1/22"
##  [369] "6/1/21"          "6/1/20"          "5/9/22"          "5/9/21"
##  [373] "5/9/20"          "5/8/22"          "5/8/21"          "5/8/20"
##  [377] "5/7/22"          "5/7/21"          "5/7/20"          "5/6/22"
##  [381] "5/6/21"          "5/6/20"          "5/5/22"          "5/5/21"
##  [385] "5/5/20"          "5/4/22"          "5/4/21"          "5/4/20"
##  [389] "5/31/22"         "5/31/21"         "5/31/20"         "5/30/22"
##  [393] "5/30/21"         "5/30/20"         "5/3/22"          "5/3/21"
##  [397] "5/3/20"          "5/29/22"         "5/29/21"         "5/29/20"
##  [401] "5/28/22"         "5/28/21"         "5/28/20"         "5/27/22"
##  [405] "5/27/21"         "5/27/20"         "5/26/22"         "5/26/21"
##  [409] "5/26/20"         "5/25/22"         "5/25/21"         "5/25/20"
##  [413] "5/24/22"         "5/24/21"         "5/24/20"         "5/23/22"
##  [417] "5/23/21"         "5/23/20"         "5/22/22"         "5/22/21"
##  [421] "5/22/20"         "5/21/22"         "5/21/21"         "5/21/20"
##  [425] "5/20/22"         "5/20/21"         "5/20/20"         "5/2/22"
##  [429] "5/2/21"          "5/2/20"          "5/19/22"         "5/19/21"
##  [433] "5/19/20"         "5/18/22"         "5/18/21"         "5/18/20"
##  [437] "5/17/22"         "5/17/21"         "5/17/20"         "5/16/22"
##  [441] "5/16/21"         "5/16/20"         "5/15/22"         "5/15/21"
##  [445] "5/15/20"         "5/14/22"         "5/14/21"         "5/14/20"
##  [449] "5/13/22"         "5/13/21"         "5/13/20"         "5/12/22"
##  [453] "5/12/21"         "5/12/20"         "5/11/22"         "5/11/21"
##  [457] "5/11/20"         "5/10/22"         "5/10/21"         "5/10/20"
##  [461] "5/1/22"          "5/1/21"          "5/1/20"          "4/9/22"
##  [465] "4/9/21"          "4/9/20"          "4/8/22"          "4/8/21"
##  [469] "4/8/20"          "4/7/22"          "4/7/21"          "4/7/20"
##  [473] "4/6/22"          "4/6/21"          "4/6/20"          "4/5/22"
##  [477] "4/5/21"          "4/5/20"          "4/4/22"          "4/4/21"
##  [481] "4/4/20"          "4/30/22"         "4/30/21"         "4/30/20"
##  [485] "4/3/22"          "4/3/21"          "4/3/20"          "4/29/22"
##  [489] "4/29/21"         "4/29/20"         "4/28/22"         "4/28/21"
##  [493] "4/28/20"         "4/27/22"         "4/27/21"         "4/27/20"
##  [497] "4/26/22"         "4/26/21"         "4/26/20"         "4/25/22"
##  [501] "4/25/21"         "4/25/20"         "4/24/22"         "4/24/21"
##  [505] "4/24/20"         "4/23/22"         "4/23/21"         "4/23/20"
##  [509] "4/22/22"         "4/22/21"         "4/22/20"         "4/21/22"
##  [513] "4/21/21"         "4/21/20"         "4/20/22"         "4/20/21"
##  [517] "4/20/20"         "4/2/22"          "4/2/21"          "4/2/20"
##  [521] "4/19/22"        "4/19/21"        "4/19/20"        "4/18/22"
##  [525] "4/18/21"        "4/18/20"        "4/17/22"        "4/17/21"
##  [529] "4/17/20"        "4/16/22"        "4/16/21"        "4/16/20"
##  [533] "4/15/22"        "4/15/21"        "4/15/20"        "4/14/22"
##  [537] "4/14/21"        "4/14/20"        "4/13/22"        "4/13/21"
##  [541] "4/13/20"        "4/12/22"        "4/12/21"        "4/12/20"
##  [545] "4/11/22"        "4/11/21"        "4/11/20"        "4/10/22"
##  [549] "4/10/21"        "4/10/20"        "4/1/22"         "4/1/21"
##  [553] "4/1/20"         "3/9/23"         "3/9/22"         "3/9/21"
##  [557] "3/9/20"         "3/8/23"         "3/8/22"         "3/8/21"
##  [561] "3/8/20"         "3/7/23"         "3/7/22"         "3/7/21"
##  [565] "3/7/20"         "3/6/23"         "3/6/22"         "3/6/21"
##  [569] "3/6/20"         "3/5/23"         "3/5/22"         "3/5/21"
##  [573] "3/5/20"         "3/4/23"         "3/4/22"         "3/4/21"
##  [577] "3/4/20"         "3/31/22"        "3/31/21"        "3/31/20"
##  [581] "3/30/22"        "3/30/21"        "3/30/20"        "3/3/23"
##  [585] "3/3/22"         "3/3/21"         "3/3/20"         "3/29/22"
##  [589] "3/29/21"        "3/29/20"        "3/28/22"        "3/28/21"
##  [593] "3/28/20"        "3/27/22"        "3/27/21"        "3/27/20"
##  [597] "3/26/22"        "3/26/21"        "3/26/20"        "3/25/22"
##  [601] "3/25/21"        "3/25/20"        "3/24/22"        "3/24/21"
##  [605] "3/24/20"        "3/23/22"        "3/23/21"        "3/23/20"
##  [609] "3/22/22"        "3/22/21"        "3/22/20"        "3/21/22"
##  [613] "3/21/21"        "3/21/20"        "3/20/22"        "3/20/21"
##  [617] "3/20/20"        "3/2/23"         "3/2/22"         "3/2/21"
##  [621] "3/2/20"         "3/19/22"        "3/19/21"        "3/19/20"
##  [625] "3/18/22"        "3/18/21"        "3/18/20"        "3/17/22"
##  [629] "3/17/21"        "3/17/20"        "3/16/22"        "3/16/21"
##  [633] "3/16/20"        "3/15/22"        "3/15/21"        "3/15/20"
##  [637] "3/14/22"        "3/14/21"        "3/14/20"        "3/13/22"
##  [641] "3/13/21"        "3/13/20"        "3/12/22"        "3/12/21"
##  [645] "3/12/20"        "3/11/22"        "3/11/21"        "3/11/20"
##  [649] "3/10/22"        "3/10/21"        "3/10/20"        "3/1/23"
##  [653] "3/1/22"         "3/1/21"         "3/1/20"         "2/9/23"
##  [657] "2/9/22"         "2/9/21"         "2/9/20"         "2/8/23"
##  [661] "2/8/22"         "2/8/21"         "2/8/20"         "2/7/23"
##  [665] "2/7/22"         "2/7/21"         "2/7/20"         "2/6/23"
##  [669] "2/6/22"         "2/6/21"         "2/6/20"         "2/5/23"
##  [673] "2/5/22"         "2/5/21"         "2/5/20"         "2/4/23"
##  [677] "2/4/22"         "2/4/21"         "2/4/20"         "2/3/23"
##  [681] "2/3/22"         "2/3/21"         "2/3/20"         "2/29/20"
##  [685] "2/28/23"        "2/28/22"        "2/28/21"        "2/28/20"
##  [689] "2/27/23"        "2/27/22"        "2/27/21"        "2/27/20"
##  [693] "2/26/23"        "2/26/22"        "2/26/21"        "2/26/20"
##  [697] "2/25/23"        "2/25/22"        "2/25/21"        "2/25/20"
##  [701] "2/24/23"        "2/24/22"        "2/24/21"        "2/24/20"
##  [705] "2/23/23"        "2/23/22"        "2/23/21"        "2/23/20"
##  [709] "2/22/23"        "2/22/22"        "2/22/21"        "2/22/20"
##  [713] "2/21/23"        "2/21/22"        "2/21/21"        "2/21/20"
##  [717] "2/20/23"        "2/20/22"        "2/20/21"        "2/20/20"
##  [721] "2/2/23"         "2/2/22"         "2/2/21"         "2/2/20"
##  [725] "2/19/23"        "2/19/22"        "2/19/21"        "2/19/20"
##  [729] "2/18/23"        "2/18/22"        "2/18/21"        "2/18/20"
##  [733] "2/17/23"        "2/17/22"        "2/17/21"        "2/17/20"
##  [737] "2/16/23"         "2/16/22"         "2/16/21"         "2/16/20"
##  [741] "2/15/23"         "2/15/22"         "2/15/21"         "2/15/20"
##  [745] "2/14/23"         "2/14/22"         "2/14/21"         "2/14/20"
##  [749] "2/13/23"         "2/13/22"         "2/13/21"         "2/13/20"
##  [753] "2/12/23"         "2/12/22"         "2/12/21"         "2/12/20"
##  [757] "2/11/23"         "2/11/22"         "2/11/21"         "2/11/20"
##  [761] "2/10/23"         "2/10/22"         "2/10/21"         "2/10/20"
##  [765] "2/1/23"          "2/1/22"          "2/1/21"          "2/1/20"
##  [769] "12/9/22"         "12/9/21"         "12/9/20"         "12/8/22"
##  [773] "12/8/21"         "12/8/20"         "12/7/22"         "12/7/21"
##  [777] "12/7/20"         "12/6/22"         "12/6/21"         "12/6/20"
##  [781] "12/5/22"         "12/5/21"         "12/5/20"         "12/4/22"
##  [785] "12/4/21"         "12/4/20"         "12/31/22"        "12/31/21"
##  [789] "12/31/20"        "12/30/22"        "12/30/21"        "12/30/20"
##  [793] "12/3/22"         "12/3/21"         "12/3/20"         "12/29/22"
##  [797] "12/29/21"        "12/29/20"        "12/28/22"        "12/28/21"
##  [801] "12/28/20"        "12/27/22"        "12/27/21"        "12/27/20"
##  [805] "12/26/22"        "12/26/21"        "12/26/20"        "12/25/22"
##  [809] "12/25/21"        "12/25/20"        "12/24/22"        "12/24/21"
##  [813] "12/24/20"        "12/23/22"        "12/23/21"        "12/23/20"
##  [817] "12/22/22"        "12/22/21"        "12/22/20"        "12/21/22"
##  [821] "12/21/21"        "12/21/20"        "12/20/22"        "12/20/21"
##  [825] "12/20/20"        "12/2/22"         "12/2/21"         "12/2/20"
##  [829] "12/19/22"        "12/19/21"        "12/19/20"        "12/18/22"
##  [833] "12/18/21"        "12/18/20"        "12/17/22"        "12/17/21"
##  [837] "12/17/20"        "12/16/22"        "12/16/21"        "12/16/20"
##  [841] "12/15/22"        "12/15/21"        "12/15/20"        "12/14/22"
##  [845] "12/14/21"        "12/14/20"        "12/13/22"        "12/13/21"
##  [849] "12/13/20"        "12/12/22"        "12/12/21"        "12/12/20"
##  [853] "12/11/22"        "12/11/21"        "12/11/20"        "12/10/22"
##  [857] "12/10/21"        "12/10/20"        "12/1/22"         "12/1/21"
##  [861] "12/1/20"         "11/9/22"         "11/9/21"         "11/9/20"
##  [865] "11/8/22"         "11/8/21"         "11/8/20"         "11/7/22"
##  [869] "11/7/21"         "11/7/20"         "11/6/22"         "11/6/21"
##  [873] "11/6/20"         "11/5/22"         "11/5/21"         "11/5/20"
##  [877] "11/4/22"         "11/4/21"         "11/4/20"         "11/30/22"
##  [881] "11/30/21"        "11/30/20"        "11/3/22"         "11/3/21"
##  [885] "11/3/20"         "11/29/22"        "11/29/21"        "11/29/20"
##  [889] "11/28/22"        "11/28/21"        "11/28/20"        "11/27/22"
##  [893] "11/27/21"        "11/27/20"        "11/26/22"        "11/26/21"
##  [897] "11/26/20"        "11/25/22"        "11/25/21"        "11/25/20"
##  [901] "11/24/22"        "11/24/21"        "11/24/20"        "11/23/22"
##  [905] "11/23/21"        "11/23/20"        "11/22/22"        "11/22/21"
##  [909] "11/22/20"        "11/21/22"        "11/21/21"        "11/21/20"
##  [913] "11/20/22"        "11/20/21"        "11/20/20"        "11/2/22"
##  [917] "11/2/21"         "11/2/20"         "11/19/22"        "11/19/21"
##  [921] "11/19/20"        "11/18/22"        "11/18/21"        "11/18/20"
##  [925] "11/17/22"        "11/17/21"        "11/17/20"        "11/16/22"
##  [929] "11/16/21"        "11/16/20"        "11/15/22"        "11/15/21"
##  [933] "11/15/20"        "11/14/22"        "11/14/21"        "11/14/20"
##  [937] "11/13/22"        "11/13/21"        "11/13/20"        "11/12/22"
##  [941] "11/12/21"        "11/12/20"        "11/11/22"        "11/11/21"
##  [945] "11/11/20"        "11/10/22"        "11/10/21"        "11/10/20"
##  [949] "11/1/22"         "11/1/21"         "11/1/20"         "10/9/22"
##  [953] "10/9/21"        "10/9/20"        "10/8/22"        "10/8/21"
##  [957] "10/8/20"        "10/7/22"        "10/7/21"        "10/7/20"
##  [961] "10/6/22"        "10/6/21"        "10/6/20"        "10/5/22"
##  [965] "10/5/21"        "10/5/20"        "10/4/22"        "10/4/21"
##  [969] "10/4/20"        "10/31/22"       "10/31/21"       "10/31/20"
##  [973] "10/30/22"       "10/30/21"       "10/30/20"       "10/3/22"
##  [977] "10/3/21"        "10/3/20"        "10/29/22"       "10/29/21"
##  [981] "10/29/20"       "10/28/22"       "10/28/21"       "10/28/20"
##  [985] "10/27/22"       "10/27/21"       "10/27/20"       "10/26/22"
##  [989] "10/26/21"       "10/26/20"       "10/25/22"       "10/25/21"
##  [993] "10/25/20"       "10/24/22"       "10/24/21"       "10/24/20"
##  [997] "10/23/22"       "10/23/21"       "10/23/20"       "10/22/22"
## [1001] "10/22/21"       "10/22/20"       "10/21/22"       "10/21/21"
## [1005] "10/21/20"       "10/20/22"       "10/20/21"       "10/20/20"
## [1009] "10/2/22"        "10/2/21"        "10/2/20"        "10/19/22"
## [1013] "10/19/21"       "10/19/20"       "10/18/22"       "10/18/21"
## [1017] "10/18/20"       "10/17/22"       "10/17/21"       "10/17/20"
## [1021] "10/16/22"       "10/16/21"       "10/16/20"       "10/15/22"
## [1025] "10/15/21"       "10/15/20"       "10/14/22"       "10/14/21"
## [1029] "10/14/20"       "10/13/22"       "10/13/21"       "10/13/20"
## [1033] "10/12/22"       "10/12/21"       "10/12/20"       "10/11/22"
## [1037] "10/11/21"       "10/11/20"       "10/10/22"       "10/10/21"
## [1041] "10/10/20"       "10/1/22"        "10/1/21"        "10/1/20"
## [1045] "1/9/23"         "1/9/22"         "1/9/21"         "1/8/23"
## [1049] "1/8/22"         "1/8/21"         "1/7/23"         "1/7/22"
## [1053] "1/7/21"         "1/6/23"         "1/6/22"         "1/6/21"
## [1057] "1/5/23"         "1/5/22"         "1/5/21"         "1/4/23"
## [1061] "1/4/22"         "1/4/21"         "1/31/23"        "1/31/22"
## [1065] "1/31/21"        "1/31/20"        "1/30/23"        "1/30/22"
## [1069] "1/30/21"        "1/30/20"        "1/3/23"         "1/3/22"
## [1073] "1/3/21"         "1/29/23"        "1/29/22"        "1/29/21"
## [1077] "1/29/20"        "1/28/23"        "1/28/22"        "1/28/21"
## [1081] "1/28/20"        "1/27/23"        "1/27/22"        "1/27/21"
## [1085] "1/27/20"        "1/26/23"        "1/26/22"        "1/26/21"
## [1089] "1/26/20"        "1/25/23"        "1/25/22"        "1/25/21"
## [1093] "1/25/20"        "1/24/23"        "1/24/22"        "1/24/21"
## [1097] "1/24/20"        "1/23/23"        "1/23/22"        "1/23/21"
## [1101] "1/23/20"        "1/22/23"        "1/22/22"        "1/22/21"
## [1105] "1/22/20"        "1/21/23"        "1/21/22"        "1/21/21"
## [1109] "1/20/23"        "1/20/22"        "1/20/21"        "1/2/23"
## [1113] "1/2/22"         "1/2/21"         "1/19/23"        "1/19/22"
## [1117] "1/19/21"        "1/18/23"        "1/18/22"        "1/18/21"
## [1121] "1/17/23"        "1/17/22"        "1/17/21"        "1/16/23"
## [1125] "1/16/22"        "1/16/21"        "1/15/23"        "1/15/22"
## [1129] "1/15/21"        "1/14/23"        "1/14/22"        "1/14/21"
## [1133] "1/13/23"        "1/13/22"        "1/13/21"        "1/12/23"
## [1137] "1/12/22"        "1/12/21"        "1/11/23"        "1/11/22"
## [1141] "1/11/21"        "1/10/23"        "1/10/22"        "1/10/21"
## [1145] "1/1/23"         "1/1/22"         "1/1/21"
```

```
head(deaths_global)
```

```
## # 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>, ...
```

```
sort(colnames(deaths_global), decreasing = TRUE)
```

```
##    [1] "Province/State" "Long"           "Lat"            "Country/Region"
##    [5] "9/9/22"         "9/9/21"         "9/9/20"         "9/8/22"
##    [9] "9/8/21"         "9/8/20"         "9/7/22"         "9/7/21"
##   [13] "9/7/20"         "9/6/22"         "9/6/21"         "9/6/20"
##   [17] "9/5/22"         "9/5/21"         "9/5/20"         "9/4/22"
##   [21] "9/4/21"         "9/4/20"         "9/30/22"        "9/30/21"
##   [25] "9/30/20"        "9/3/22"         "9/3/21"         "9/3/20"
##   [29] "9/29/22"        "9/29/21"        "9/29/20"        "9/28/22"
##   [33] "9/28/21"        "9/28/20"        "9/27/22"        "9/27/21"
##   [37] "9/27/20"        "9/26/22"        "9/26/21"        "9/26/20"
##   [41] "9/25/22"        "9/25/21"        "9/25/20"        "9/24/22"
##   [45] "9/24/21"        "9/24/20"        "9/23/22"        "9/23/21"
##   [49] "9/23/20"        "9/22/22"        "9/22/21"        "9/22/20"
##   [53] "9/21/22"        "9/21/21"        "9/21/20"        "9/20/22"
##   [57] "9/20/21"        "9/20/20"        "9/2/22"         "9/2/21"
##   [61] "9/2/20"         "9/19/22"        "9/19/21"        "9/19/20"
##   [65] "9/18/22"        "9/18/21"        "9/18/20"        "9/17/22"
##   [69] "9/17/21"        "9/17/20"        "9/16/22"        "9/16/21"
##   [73] "9/16/20"        "9/15/22"        "9/15/21"        "9/15/20"
##   [77] "9/14/22"        "9/14/21"        "9/14/20"        "9/13/22"
##   [81] "9/13/21"        "9/13/20"        "9/12/22"        "9/12/21"
##   [85] "9/12/20"        "9/11/22"        "9/11/21"        "9/11/20"
##   [89] "9/10/22"        "9/10/21"        "9/10/20"        "9/1/22"
##   [93] "9/1/21"         "9/1/20"         "8/9/22"         "8/9/21"
##   [97] "8/9/20"         "8/8/22"         "8/8/21"         "8/8/20"
##  [101] "8/7/22"         "8/7/21"         "8/7/20"         "8/6/22"
##  [105] "8/6/21"         "8/6/20"         "8/5/22"         "8/5/21"
##  [109] "8/5/20"         "8/4/22"         "8/4/21"         "8/4/20"
##  [113] "8/31/22"        "8/31/21"        "8/31/20"        "8/30/22"
##  [117] "8/30/21"        "8/30/20"        "8/3/22"         "8/3/21"
##  [121] "8/3/20"         "8/29/22"        "8/29/21"        "8/29/20"
##  [125] "8/28/22"        "8/28/21"        "8/28/20"        "8/27/22"
##  [129] "8/27/21"        "8/27/20"        "8/26/22"        "8/26/21"
##  [133] "8/26/20"        "8/25/22"        "8/25/21"        "8/25/20"
##  [137] "8/24/22"        "8/24/21"        "8/24/20"        "8/23/22"
##  [141] "8/23/21"        "8/23/20"        "8/22/22"        "8/22/21"
##  [145] "8/22/20"        "8/21/22"        "8/21/21"        "8/21/20"
##  [149] "8/20/22"        "8/20/21"        "8/20/20"        "8/2/22"
##  [153] "8/2/21"         "8/2/20"         "8/19/22"        "8/19/21"
##  [157] "8/19/20"        "8/18/22"        "8/18/21"        "8/18/20"
##  [161] "8/17/22"        "8/17/21"        "8/17/20"        "8/16/22"
##  [165] "8/16/21"        "8/16/20"        "8/15/22"        "8/15/21"
##  [169] "8/15/20"        "8/14/22"        "8/14/21"        "8/14/20"
##  [173] "8/13/22"        "8/13/21"        "8/13/20"        "8/12/22"
##  [177] "8/12/21"        "8/12/20"        "8/11/22"        "8/11/21"
##  [181] "8/11/20"        "8/10/22"        "8/10/21"        "8/10/20"
##  [185] "8/1/22"         "8/1/21"         "8/1/20"         "7/9/22"
##  [189] "7/9/21"         "7/9/20"         "7/8/22"         "7/8/21"
##  [193] "7/8/20"         "7/7/22"         "7/7/21"         "7/7/20"
##  [197] "7/6/22"         "7/6/21"         "7/6/20"         "7/5/22"
##  [201] "7/5/21"         "7/5/20"         "7/4/22"         "7/4/21"
##  [205] "7/4/20"         "7/31/22"        "7/31/21"        "7/31/20"
##  [209] "7/30/22"        "7/30/21"        "7/30/20"        "7/3/22"
##  [213] "7/3/21"         "7/3/20"         "7/29/22"        "7/29/21"
##  [217] "7/29/20"        "7/28/22"        "7/28/21"        "7/28/20"
##  [221] "7/27/22"        "7/27/21"        "7/27/20"        "7/26/22"
##  [225] "7/26/21"        "7/26/20"        "7/25/22"        "7/25/21"
##  [229] "7/25/20"        "7/24/22"        "7/24/21"        "7/24/20"
##  [233] "7/23/22"        "7/23/21"        "7/23/20"        "7/22/22"
##  [237] "7/22/21"        "7/22/20"        "7/21/22"        "7/21/21"
##  [241] "7/21/20"        "7/20/22"        "7/20/21"        "7/20/20"
##  [245] "7/2/22"         "7/2/21"         "7/2/20"         "7/19/22"
##  [249] "7/19/21"        "7/19/20"        "7/18/22"        "7/18/21"
##  [253] "7/18/20"        "7/17/22"        "7/17/21"        "7/17/20"
##  [257] "7/16/22"        "7/16/21"        "7/16/20"        "7/15/22"
##  [261] "7/15/21"        "7/15/20"        "7/14/22"        "7/14/21"
##  [265] "7/14/20"        "7/13/22"        "7/13/21"        "7/13/20"
##  [269] "7/12/22"        "7/12/21"        "7/12/20"        "7/11/22"
##  [273] "7/11/21"        "7/11/20"        "7/10/22"        "7/10/21"
##  [277] "7/10/20"        "7/1/22"         "7/1/21"         "7/1/20"
##  [281] "6/9/22"         "6/9/21"         "6/9/20"         "6/8/22"
##  [285] "6/8/21"         "6/8/20"         "6/7/22"         "6/7/21"
##  [289] "6/7/20"         "6/6/22"         "6/6/21"         "6/6/20"
##  [293] "6/5/22"         "6/5/21"         "6/5/20"         "6/4/22"
##  [297] "6/4/21"         "6/4/20"         "6/30/22"        "6/30/21"
##  [301] "6/30/20"        "6/3/22"         "6/3/21"         "6/3/20"
##  [305] "6/29/22"        "6/29/21"        "6/29/20"        "6/28/22"
##  [309] "6/28/21"        "6/28/20"        "6/27/22"        "6/27/21"
##  [313] "6/27/20"        "6/26/22"        "6/26/21"        "6/26/20"
##  [317] "6/25/22"        "6/25/21"        "6/25/20"        "6/24/22"
##  [321] "6/24/21"        "6/24/20"        "6/23/22"        "6/23/21"
##  [325] "6/23/20"        "6/22/22"        "6/22/21"        "6/22/20"
##  [329] "6/21/22"        "6/21/21"        "6/21/20"        "6/20/22"
##  [333] "6/20/21"        "6/20/20"        "6/2/22"         "6/2/21"
##  [337] "6/2/20"         "6/19/22"        "6/19/21"        "6/19/20"
##  [341] "6/18/22"        "6/18/21"        "6/18/20"        "6/17/22"
##  [345] "6/17/21"        "6/17/20"        "6/16/22"        "6/16/21"
##  [349] "6/16/20"        "6/15/22"        "6/15/21"        "6/15/20"
##  [353] "6/14/22"        "6/14/21"        "6/14/20"        "6/13/22"
##  [357] "6/13/21"        "6/13/20"        "6/12/22"        "6/12/21"
##  [361] "6/12/20"        "6/11/22"        "6/11/21"        "6/11/20"
##  [365] "6/10/22"        "6/10/21"        "6/10/20"        "6/1/22"
##  [369] "6/1/21"         "6/1/20"         "5/9/22"         "5/9/21"
##  [373] "5/9/20"         "5/8/22"         "5/8/21"         "5/8/20"
##  [377] "5/7/22"         "5/7/21"         "5/7/20"         "5/6/22"
##  [381] "5/6/21"         "5/6/20"         "5/5/22"         "5/5/21"
##  [385] "5/5/20"         "5/4/22"         "5/4/21"         "5/4/20"
##  [389] "5/31/22"        "5/31/21"        "5/31/20"        "5/30/22"
##  [393] "5/30/21"        "5/30/20"        "5/3/22"         "5/3/21"
##  [397] "5/3/20"         "5/29/22"        "5/29/21"        "5/29/20"
##  [401] "5/28/22"        "5/28/21"        "5/28/20"        "5/27/22"
##  [405] "5/27/21"        "5/27/20"        "5/26/22"        "5/26/21"
##  [409] "5/26/20"        "5/25/22"        "5/25/21"        "5/25/20"
##  [413] "5/24/22"        "5/24/21"        "5/24/20"        "5/23/22"
##  [417] "5/23/21"        "5/23/20"        "5/22/22"        "5/22/21"
##  [421] "5/22/20"        "5/21/22"        "5/21/21"        "5/21/20"
##  [425] "5/20/22"        "5/20/21"        "5/20/20"        "5/2/22"
##  [429] "5/2/21"         "5/2/20"         "5/19/22"        "5/19/21"
##  [433] "5/19/20"        "5/18/22"        "5/18/21"        "5/18/20"
##  [437] "5/17/22"        "5/17/21"        "5/17/20"        "5/16/22"
##  [441] "5/16/21"        "5/16/20"        "5/15/22"        "5/15/21"
##  [445] "5/15/20"        "5/14/22"        "5/14/21"        "5/14/20"
##  [449] "5/13/22"        "5/13/21"        "5/13/20"        "5/12/22"
##  [453] "5/12/21"        "5/12/20"        "5/11/22"        "5/11/21"
##  [457] "5/11/20"        "5/10/22"        "5/10/21"        "5/10/20"
##  [461] "5/1/22"         "5/1/21"         "5/1/20"         "4/9/22"
##  [465] "4/9/21"         "4/9/20"         "4/8/22"         "4/8/21"
##  [469] "4/8/20"         "4/7/22"         "4/7/21"         "4/7/20"
##  [473] "4/6/22"         "4/6/21"         "4/6/20"         "4/5/22"
##  [477] "4/5/21"         "4/5/20"         "4/4/22"         "4/4/21"
##  [481] "4/4/20"         "4/30/22"        "4/30/21"        "4/30/20"
##  [485] "4/3/22"         "4/3/21"         "4/3/20"         "4/29/22"
##  [489] "4/29/21"        "4/29/20"        "4/28/22"        "4/28/21"
##  [493] "4/28/20"        "4/27/22"        "4/27/21"        "4/27/20"
##  [497] "4/26/22"        "4/26/21"        "4/26/20"        "4/25/22"
##  [501] "4/25/21"        "4/25/20"        "4/24/22"        "4/24/21"
##  [505] "4/24/20"        "4/23/22"        "4/23/21"        "4/23/20"
##  [509] "4/22/22"        "4/22/21"        "4/22/20"        "4/21/22"
##  [513] "4/21/21"        "4/21/20"        "4/20/22"        "4/20/21"
##  [517] "4/20/20"        "4/2/22"         "4/2/21"         "4/2/20"
##  [521] "4/19/22"        "4/19/21"        "4/19/20"        "4/18/22"
##  [525] "4/18/21"        "4/18/20"        "4/17/22"        "4/17/21"
##  [529] "4/17/20"        "4/16/22"        "4/16/21"        "4/16/20"
##  [533] "4/15/22"        "4/15/21"        "4/15/20"        "4/14/22"
##  [537] "4/14/21"        "4/14/20"        "4/13/22"        "4/13/21"
##  [541] "4/13/20"        "4/12/22"        "4/12/21"        "4/12/20"
##  [545] "4/11/22"        "4/11/21"        "4/11/20"        "4/10/22"
##  [549] "4/10/21"        "4/10/20"        "4/1/22"         "4/1/21"
##  [553] "4/1/20"         "3/9/23"         "3/9/22"         "3/9/21"
##  [557] "3/9/20"         "3/8/23"         "3/8/22"         "3/8/21"
##  [561] "3/8/20"         "3/7/23"         "3/7/22"         "3/7/21"
##  [565] "3/7/20"         "3/6/23"         "3/6/22"         "3/6/21"
##  [569] "3/6/20"         "3/5/23"         "3/5/22"         "3/5/21"
##  [573] "3/5/20"         "3/4/23"         "3/4/22"         "3/4/21"
##  [577] "3/4/20"         "3/31/22"        "3/31/21"        "3/31/20"
## [581] "3/30/22"        "3/30/21"        "3/30/20"        "3/3/23"
## [585] "3/3/22"         "3/3/21"         "3/3/20"         "3/29/22"
## [589] "3/29/21"        "3/29/20"        "3/28/22"        "3/28/21"
## [593] "3/28/20"        "3/27/22"        "3/27/21"        "3/27/20"
## [597] "3/26/22"        "3/26/21"        "3/26/20"        "3/25/22"
## [601] "3/25/21"        "3/25/20"        "3/24/22"        "3/24/21"
## [605] "3/24/20"        "3/23/22"        "3/23/21"        "3/23/20"
## [609] "3/22/22"        "3/22/21"        "3/22/20"        "3/21/22"
## [613] "3/21/21"        "3/21/20"        "3/20/22"        "3/20/21"
## [617] "3/20/20"        "3/2/23"         "3/2/22"         "3/2/21"
## [621] "3/2/20"         "3/19/22"        "3/19/21"        "3/19/20"
## [625] "3/18/22"        "3/18/21"        "3/18/20"        "3/17/22"
## [629] "3/17/21"        "3/17/20"        "3/16/22"        "3/16/21"
## [633] "3/16/20"        "3/15/22"        "3/15/21"        "3/15/20"
## [637] "3/14/22"        "3/14/21"        "3/14/20"        "3/13/22"
## [641] "3/13/21"        "3/13/20"        "3/12/22"        "3/12/21"
## [645] "3/12/20"        "3/11/22"        "3/11/21"        "3/11/20"
## [649] "3/10/22"        "3/10/21"        "3/10/20"        "3/1/23"
## [653] "3/1/22"         "3/1/21"         "3/1/20"         "2/9/23"
## [657] "2/9/22"         "2/9/21"         "2/9/20"         "2/8/23"
## [661] "2/8/22"         "2/8/21"         "2/8/20"         "2/7/23"
## [665] "2/7/22"         "2/7/21"         "2/7/20"         "2/6/23"
## [669] "2/6/22"         "2/6/21"         "2/6/20"         "2/5/23"
## [673] "2/5/22"         "2/5/21"         "2/5/20"         "2/4/23"
## [677] "2/4/22"         "2/4/21"         "2/4/20"         "2/3/23"
## [681] "2/3/22"         "2/3/21"         "2/3/20"         "2/29/20"
## [685] "2/28/23"        "2/28/22"        "2/28/21"        "2/28/20"
## [689] "2/27/23"        "2/27/22"        "2/27/21"        "2/27/20"
## [693] "2/26/23"        "2/26/22"        "2/26/21"        "2/26/20"
## [697] "2/25/23"        "2/25/22"        "2/25/21"        "2/25/20"
## [701] "2/24/23"        "2/24/22"        "2/24/21"        "2/24/20"
## [705] "2/23/23"        "2/23/22"        "2/23/21"        "2/23/20"
## [709] "2/22/23"        "2/22/22"        "2/22/21"        "2/22/20"
## [713] "2/21/23"        "2/21/22"        "2/21/21"        "2/21/20"
## [717] "2/20/23"        "2/20/22"        "2/20/21"        "2/20/20"
## [721] "2/2/23"         "2/2/22"         "2/2/21"         "2/2/20"
## [725] "2/19/23"        "2/19/22"        "2/19/21"        "2/19/20"
## [729] "2/18/23"        "2/18/22"        "2/18/21"        "2/18/20"
## [733] "2/17/23"        "2/17/22"        "2/17/21"        "2/17/20"
## [737] "2/16/23"        "2/16/22"        "2/16/21"        "2/16/20"
## [741] "2/15/23"        "2/15/22"        "2/15/21"        "2/15/20"
## [745] "2/14/23"        "2/14/22"        "2/14/21"        "2/14/20"
## [749] "2/13/23"        "2/13/22"        "2/13/21"        "2/13/20"
## [753] "2/12/23"        "2/12/22"        "2/12/21"        "2/12/20"
## [757] "2/11/23"        "2/11/22"        "2/11/21"        "2/11/20"
## [761] "2/10/23"        "2/10/22"        "2/10/21"        "2/10/20"
## [765] "2/1/23"         "2/1/22"         "2/1/21"         "2/1/20"
## [769] "12/9/22"        "12/9/21"        "12/9/20"        "12/8/22"
## [773] "12/8/21"        "12/8/20"        "12/7/22"        "12/7/21"
## [777] "12/7/20"        "12/6/22"        "12/6/21"        "12/6/20"
## [781] "12/5/22"        "12/5/21"        "12/5/20"        "12/4/22"
## [785] "12/4/21"        "12/4/20"        "12/31/22"       "12/31/21"
## [789] "12/31/20"       "12/30/22"       "12/30/21"       "12/30/20"
## [793] "12/3/22"        "12/3/21"        "12/3/20"        "12/29/22"
##  [797] "12/29/21"       "12/29/20"       "12/28/22"       "12/28/21"
##  [801] "12/28/20"       "12/27/22"       "12/27/21"       "12/27/20"
##  [805] "12/26/22"       "12/26/21"       "12/26/20"       "12/25/22"
##  [809] "12/25/21"       "12/25/20"       "12/24/22"       "12/24/21"
##  [813] "12/24/20"       "12/23/22"       "12/23/21"       "12/23/20"
##  [817] "12/22/22"       "12/22/21"       "12/22/20"       "12/21/22"
##  [821] "12/21/21"       "12/21/20"       "12/20/22"       "12/20/21"
##  [825] "12/20/20"       "12/2/22"        "12/2/21"        "12/2/20"
##  [829] "12/19/22"       "12/19/21"       "12/19/20"       "12/18/22"
##  [833] "12/18/21"       "12/18/20"       "12/17/22"       "12/17/21"
##  [837] "12/17/20"       "12/16/22"       "12/16/21"       "12/16/20"
##  [841] "12/15/22"       "12/15/21"       "12/15/20"       "12/14/22"
##  [845] "12/14/21"       "12/14/20"       "12/13/22"       "12/13/21"
##  [849] "12/13/20"       "12/12/22"       "12/12/21"       "12/12/20"
##  [853] "12/11/22"       "12/11/21"       "12/11/20"       "12/10/22"
##  [857] "12/10/21"       "12/10/20"       "12/1/22"        "12/1/21"
##  [861] "12/1/20"        "11/9/22"        "11/9/21"        "11/9/20"
##  [865] "11/8/22"        "11/8/21"        "11/8/20"        "11/7/22"
##  [869] "11/7/21"        "11/7/20"        "11/6/22"        "11/6/21"
##  [873] "11/6/20"        "11/5/22"        "11/5/21"        "11/5/20"
##  [877] "11/4/22"        "11/4/21"        "11/4/20"        "11/30/22"
##  [881] "11/30/21"       "11/30/20"       "11/3/22"        "11/3/21"
##  [885] "11/3/20"        "11/29/22"       "11/29/21"       "11/29/20"
##  [889] "11/28/22"       "11/28/21"       "11/28/20"       "11/27/22"
##  [893] "11/27/21"       "11/27/20"       "11/26/22"       "11/26/21"
##  [897] "11/26/20"       "11/25/22"       "11/25/21"       "11/25/20"
##  [901] "11/24/22"       "11/24/21"       "11/24/20"       "11/23/22"
##  [905] "11/23/21"       "11/23/20"       "11/22/22"       "11/22/21"
##  [909] "11/22/20"       "11/21/22"       "11/21/21"       "11/21/20"
##  [913] "11/20/22"       "11/20/21"       "11/20/20"       "11/2/22"
##  [917] "11/2/21"        "11/2/20"        "11/19/22"       "11/19/21"
##  [921] "11/19/20"       "11/18/22"       "11/18/21"       "11/18/20"
##  [925] "11/17/22"       "11/17/21"       "11/17/20"       "11/16/22"
##  [929] "11/16/21"       "11/16/20"       "11/15/22"       "11/15/21"
##  [933] "11/15/20"       "11/14/22"       "11/14/21"       "11/14/20"
##  [937] "11/13/22"       "11/13/21"       "11/13/20"       "11/12/22"
##  [941] "11/12/21"       "11/12/20"       "11/11/22"       "11/11/21"
##  [945] "11/11/20"       "11/10/22"       "11/10/21"       "11/10/20"
##  [949] "11/1/22"        "11/1/21"        "11/1/20"        "10/9/22"
##  [953] "10/9/21"        "10/9/20"        "10/8/22"        "10/8/21"
##  [957] "10/8/20"        "10/7/22"        "10/7/21"        "10/7/20"
##  [961] "10/6/22"        "10/6/21"        "10/6/20"        "10/5/22"
##  [965] "10/5/21"        "10/5/20"        "10/4/22"        "10/4/21"
##  [969] "10/4/20"        "10/31/22"       "10/31/21"       "10/31/20"
##  [973] "10/30/22"       "10/30/21"       "10/30/20"       "10/3/22"
##  [977] "10/3/21"        "10/3/20"        "10/29/22"       "10/29/21"
##  [981] "10/29/20"       "10/28/22"       "10/28/21"       "10/28/20"
##  [985] "10/27/22"       "10/27/21"       "10/27/20"       "10/26/22"
##  [989] "10/26/21"       "10/26/20"       "10/25/22"       "10/25/21"
##  [993] "10/25/20"       "10/24/22"       "10/24/21"       "10/24/20"
##  [997] "10/23/22"       "10/23/21"       "10/23/20"       "10/22/22"
## [1001] "10/22/21"       "10/22/20"       "10/21/22"       "10/21/21"
## [1005] "10/21/20"       "10/20/22"       "10/20/21"       "10/20/20"
## [1009] "10/2/22"        "10/2/21"        "10/2/20"        "10/19/22"
## [1013] "10/19/21"       "10/19/20"       "10/18/22"       "10/18/21"
## [1017] "10/18/20"       "10/17/22"       "10/17/21"       "10/17/20"
## [1021] "10/16/22"       "10/16/21"       "10/16/20"       "10/15/22"
## [1025] "10/15/21"       "10/15/20"       "10/14/22"       "10/14/21"
## [1029] "10/14/20"       "10/13/22"       "10/13/21"       "10/13/20"
## [1033] "10/12/22"       "10/12/21"       "10/12/20"       "10/11/22"
## [1037] "10/11/21"       "10/11/20"       "10/10/22"       "10/10/21"
## [1041] "10/10/20"       "10/1/22"        "10/1/21"        "10/1/20"
## [1045] "1/9/23"         "1/9/22"         "1/9/21"         "1/8/23"
## [1049] "1/8/22"         "1/8/21"         "1/7/23"         "1/7/22"
## [1053] "1/7/21"         "1/6/23"         "1/6/22"         "1/6/21"
## [1057] "1/5/23"         "1/5/22"         "1/5/21"         "1/4/23"
## [1061] "1/4/22"         "1/4/21"         "1/31/23"        "1/31/22"
## [1065] "1/31/21"        "1/31/20"        "1/30/23"        "1/30/22"
## [1069] "1/30/21"        "1/30/20"        "1/3/23"         "1/3/22"
## [1073] "1/3/21"         "1/29/23"        "1/29/22"        "1/29/21"
## [1077] "1/29/20"        "1/28/23"        "1/28/22"        "1/28/21"
## [1081] "1/28/20"        "1/27/23"        "1/27/22"        "1/27/21"
## [1085] "1/27/20"        "1/26/23"        "1/26/22"        "1/26/21"
## [1089] "1/26/20"        "1/25/23"        "1/25/22"        "1/25/21"
## [1093] "1/25/20"        "1/24/23"        "1/24/22"        "1/24/21"
## [1097] "1/24/20"        "1/23/23"        "1/23/22"        "1/23/21"
## [1101] "1/23/20"        "1/22/23"        "1/22/22"        "1/22/21"
## [1105] "1/22/20"        "1/21/23"        "1/21/22"        "1/21/21"
## [1109] "1/20/23"        "1/20/22"        "1/20/21"        "1/2/23"
## [1113] "1/2/22"         "1/2/21"         "1/19/23"        "1/19/22"
## [1117] "1/19/21"        "1/18/23"        "1/18/22"        "1/18/21"
## [1121] "1/17/23"        "1/17/22"        "1/17/21"        "1/16/23"
## [1125] "1/16/22"        "1/16/21"        "1/15/23"        "1/15/22"
## [1129] "1/15/21"        "1/14/23"        "1/14/22"        "1/14/21"
## [1133] "1/13/23"        "1/13/22"        "1/13/21"        "1/12/23"
## [1137] "1/12/22"        "1/12/21"        "1/11/23"        "1/11/22"
## [1141] "1/11/21"        "1/10/23"        "1/10/22"        "1/10/21"
## [1145] "1/1/23"         "1/1/22"         "1/1/21"
```

```
head(confirmed_us)
```

```
## # A tibble: 6 x 1,154
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 2 84001003 US    USA     840  1003 Baldwin Alabama        US              30.7
## 3 84001005 US    USA     840  1005 Barbour Alabama        US              31.9
## 4 84001007 US    USA     840  1007 Bibb    Alabama        US              33.0
## 5 84001009 US    USA     840  1009 Blount  Alabama        US              34.0
## 6 84001011 US    USA     840  1011 Bullock Alabama        US              32.1
## # i 1,145 more variables: Long_ <dbl>, Combined_Key <chr>, ‘1/22/20‘ <dbl>,
## #   ‘1/23/20‘ <dbl>, ‘1/24/20‘ <dbl>, ‘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>, ...
```

```r
head(deaths_us)
```

```
## # A tibble: 6 x 1,155
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 2 84001003 US    USA     840  1003 Baldwin Alabama        US              30.7
## 3 84001005 US    USA     840  1005 Barbour Alabama        US              31.9
## 4 84001007 US    USA     840  1007 Bibb    Alabama        US              33.0
## 5 84001009 US    USA     840  1009 Blount  Alabama        US              34.0
## 6 84001011 US    USA     840  1011 Bullock Alabama        US              32.1
## # i 1,146 more variables: Long_ <dbl>, Combined_Key <chr>, Population <dbl>,
## #   ‘1/22/20‘ <dbl>, ‘1/23/20‘ <dbl>, ‘1/24/20‘ <dbl>, ‘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>, ...
```

```r
cases <- confirmed_us %>%
  pivot_longer(cols = -c(UID:Combined_Key), names_to = "date", values_to = "Cases")%>%
  select(-c(iso2, iso3, code3, FIPS, UID, Country_Region))%>%
  mutate(date = mdy(date))

summary(cases)
```

```
##     Admin2           Province_State          Lat              Long_
##   Length:3819906     Length:3819906     Min.   :-14.27   Min.   :-174.16
##   Class :character   Class :character   1st Qu.: 33.90   1st Qu.: -97.81
##   Mode  :character   Mode  :character   Median : 38.01   Median : -89.49
##                                         Mean   : 36.72   Mean   : -88.64
##                                         3rd Qu.: 41.58   3rd Qu.: -82.31
##                                         Max.   : 69.31   Max.   : 145.67
##   Combined_Key            date                Cases
##   Length:3819906     Min.   :2020-01-22   Min.   :  -3073
##   Class :character   1st Qu.:2020-11-02   1st Qu.:    330
##   Mode  :character   Median :2021-08-15   Median :   2272
##                      Mean   :2021-08-15   Mean   :  14088
##                      3rd Qu.:2022-05-28   3rd Qu.:   8159
##                      Max.   :2023-03-09   Max.   :3710586
```

```r
deaths <- deaths_us %>%
  pivot_longer(cols = -c(UID:Population), names_to = "date", values_to = "deaths")%>%
  select(-c(iso2, iso3, code3, FIPS, UID, Country_Region))%>%
  mutate(date = mdy(date))

summary(deaths)
```

```
##     Admin2           Province_State          Lat              Long_
##   Length:3819906     Length:3819906     Min.   :-14.27   Min.   :-174.16
##   Class :character   Class :character   1st Qu.: 33.90   1st Qu.: -97.81
##   Mode  :character   Mode  :character   Median : 38.01   Median : -89.49
##                                         Mean   : 36.72   Mean   : -88.64
##                                         3rd Qu.: 41.58   3rd Qu.: -82.31
##                                         Max.   : 69.31   Max.   : 145.67
##  Combined_Key         Population          date                 deaths
##  Length:3819906     Min.   :       0   Min.   :2020-01-22   Min.   :  -82.0
##  Class :character   1st Qu.:    9917   1st Qu.:2020-11-02   1st Qu.:    4.0
##  Mode  :character   Median :   24892   Median :2021-08-15   Median :   37.0
##                     Mean   :   99604   Mean   :2021-08-15   Mean   :  186.9
##                     3rd Qu.:   64979   3rd Qu.:2022-05-28   3rd Qu.:  122.0
##                     Max.   :10039107   Max.   :2023-03-09   Max.   :35545.0
```

```
# For 'global_cases' df, make 'Province/State' and 'Country/Region' factors and pivot dates into rows
global_cases <- mutate_at(global_cases, vars('Province/State', 'Country/Region'), as.factor) %>%
    pivot_longer(cols = -c('Province/State', 'Country/Region', 'Lat', 'Long'),
                 names_to = 'Date',
                 values_to = 'Cases') %>%
    select(-c('Lat', 'Long'))

# For 'global_deaths' df, make 'Province/State' and 'Country/Region' factors and pivot dates into rows
global_deaths <- mutate_at(global_deaths, vars('Province/State', 'Country/Region'), as.factor) %>%
    pivot_longer(cols = -c('Province/State', 'Country/Region', 'Lat', 'Long'),
            names_to = 'Date',
            values_to = 'Deaths') %>%
    select(-c('Lat', 'Long'))

# Merge 'global_cases' df and 'global_deaths' df into 'global' df and rename columns
```

```
global <- global_cases %>%
    full_join(global_deaths) %>%
    rename(Country_Region = 'Country/Region',
            Province_State = 'Province/State') %>%
    mutate(Date = mdy(Date))
```

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

```
# Combine 'Province_State' and 'Country_Region' columns into one 'Combined_Key' column
global <- global %>%
    unite('Combined_Key',
        c(Province_State, Country_Region),
        sep = ', ',
        na.rm = TRUE,
        remove = FALSE)

# Join 'global' df with global population lookup table df and remove unneeded columns
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)
```

```
global_cases_per_hundred <- global %>%
  group_by(Country_Region, Population) %>%
  summarize(Cases = max(Cases), Population = max(Population, na.rm = T)) %>%
  mutate(Cases_per_hundred = (Cases/Population)*100) %>%
  arrange(desc(Cases_per_hundred)) %>%
  filter(Population > 0) %>%
  select(Country_Region, Population, Cases, Cases_per_hundred) %>%
  ungroup()
```

```
global_cases_per_hundred <- global_cases_per_hundred %>%
  group_by(Country_Region, Population) %>%
  summarize(Cases = sum(Cases), Population = sum(Population)) %>%
  summarize(Cases = max(Cases), Population = max(Population, na.rm = T)) %>%
  mutate(Cases_per_hundred = (Cases/Population)*100) %>%
  arrange(desc(Cases_per_hundred)) %>%
  filter(Population > 0) %>%
  select(Country_Region, Population, Cases, Cases_per_hundred) %>%
  ungroup()
```

```
global_deaths_per_hundred <- global %>%
  group_by(Country_Region, Population) %>%
  summarize(Deaths = max(Deaths), Population = max(Population, na.rm = T)) %>%
  mutate(Deaths_per_hundred = (Deaths/Population)*100) %>%
  arrange(desc(Deaths_per_hundred)) %>%
  filter(Population > 0) %>%
  select(Country_Region, Population, Deaths, Deaths_per_hundred) %>%
  ungroup()

# Combine populations and death totals for countries with 'State_Province' factors
global_deaths_per_hundred <- global_deaths_per_hundred %>%
  group_by(Country_Region, Population) %>%
  summarize(Deaths = sum(Deaths), Population = sum(Population)) %>%
  summarize(Deaths = max(Deaths), Population = max(Population, na.rm = T)) %>%
  mutate(Deaths_per_hundred = (Deaths/Population)*100) %>%
  arrange(desc(Deaths_per_hundred)) %>%
  filter(Population > 0) %>%
  select(Country_Region, Population, Deaths, Deaths_per_hundred) %>%
  ungroup()
```

```
US_cases <- mutate_at(US_cases, vars(Admin2, Province_State, Country_Region), as.factor) %>%
  rename(County = 'Admin2') %>%
    pivot_longer(cols = -(UID:Combined_Key),
        names_to = 'Date',
        values_to = 'Cases') %>%
  filter(Cases >= 0) %>%
    select(County:Cases) %>%
    mutate(Date = mdy(Date)) %>%
    select(-c(Lat, Long_))

# For 'US_deaths' df, create factors and pivot dates into rows, change 'Date' column to mdy
US_deaths <- mutate_at(US_deaths, vars(Admin2, Province_State, Country_Region), as.factor) %>%
  rename(County = 'Admin2') %>%
    pivot_longer(cols = -(UID:Population),
        names_to = 'Date',
        values_to = 'Deaths') %>%
```

```r
  filter(Deaths >= 0) %>%
    select(County:Deaths) %>%
    mutate(Date = mdy(Date)) %>%
    select(-c(Lat, Long_))

# Merge 'US_cases' df and 'US_deaths' df into 'US' df
US <- US_cases %>%
    full_join(US_deaths)
```

```
## Joining with `by = join_by(County, Province_State, Country_Region,
## Combined_Key, Date)`
```

```r
# For 'US_by_state' df, calculate sums of 'Cases', 'Deaths', and 'Population' variables by US state
US_by_state <- US %>%
    group_by(Province_State, Country_Region, Date) %>%
    summarize(Cases = sum(Cases), Deaths = sum(Deaths), Population = sum(Population)) %>%
    select(Province_State, Country_Region, Date, Cases, Deaths, Population) %>%
    ungroup()

# For 'US_by_state_cases_deaths_per_day' df, calculate 'New_Cases' and 'New_Deaths' variables
US_by_state_cases_deaths_per_day <- US_by_state %>%
  group_by(Province_State) %>%
  mutate(New_Cases = Cases - lag(Cases),
         New_Deaths = Deaths - lag(Deaths)) %>%
  select(Province_State, Country_Region, Date, Cases, Deaths, Population,
         New_Cases, New_Deaths) %>%
  ungroup()
```

```r
index1 <- which(US_by_state_cases_deaths_per_day$New_Cases >= 0)
US_by_state_cases_deaths_per_day <- US_by_state_cases_deaths_per_day[index1,]

# Remove negative 'New_Deaths' values from 'US_by_state_cases_deaths_per_day' df
index2 <- which(US_by_state_cases_deaths_per_day$New_Deaths >= 0)
US_by_state_cases_deaths_per_day <- US_by_state_cases_deaths_per_day[index2,]

# Group 'US_by_state_cases_deaths_per_day' df by 'Province_State' and filter rows with population > 0
US_by_state_cases_deaths_per_day <- US_by_state_cases_deaths_per_day %>%
  group_by(Province_State, Date) %>%
  select(Province_State, Country_Region, Date, Cases, Deaths, Population,
         New_Cases, New_Deaths) %>%
  filter(Population > 0) %>%
  ungroup()

# Group by 'Province_State', record max in 'Cases' variable, and calculate 'Cases_per_hundred' variable
US_by_state_cases_per_hundred <- US_by_state %>%
  group_by(Province_State, Population) %>%
  summarize(Cases = max(Cases)) %>%
  mutate(Cases_per_hundred = (Cases/Population)*100) %>%
  arrange(desc(Cases_per_hundred)) %>%
  filter(Population > 0) %>%
  select(Province_State, Population, Cases, Cases_per_hundred) %>%
  ungroup()
```

```r
# Group by 'Province_State', record max in 'Deaths' variable, and calculate 'Deaths_per_hundred' variab
US_by_state_deaths_per_hundred <- US_by_state %>%
  group_by(Province_State, Population) %>%
  summarize(Deaths = max(Deaths)) %>%
  mutate(Deaths_per_hundred = (Deaths/Population)*100) %>%
  arrange(desc(Deaths_per_hundred)) %>%
  filter(Population > 0) %>%
  select(Province_State, Population, Deaths, Deaths_per_hundred) %>%
  ungroup()
```

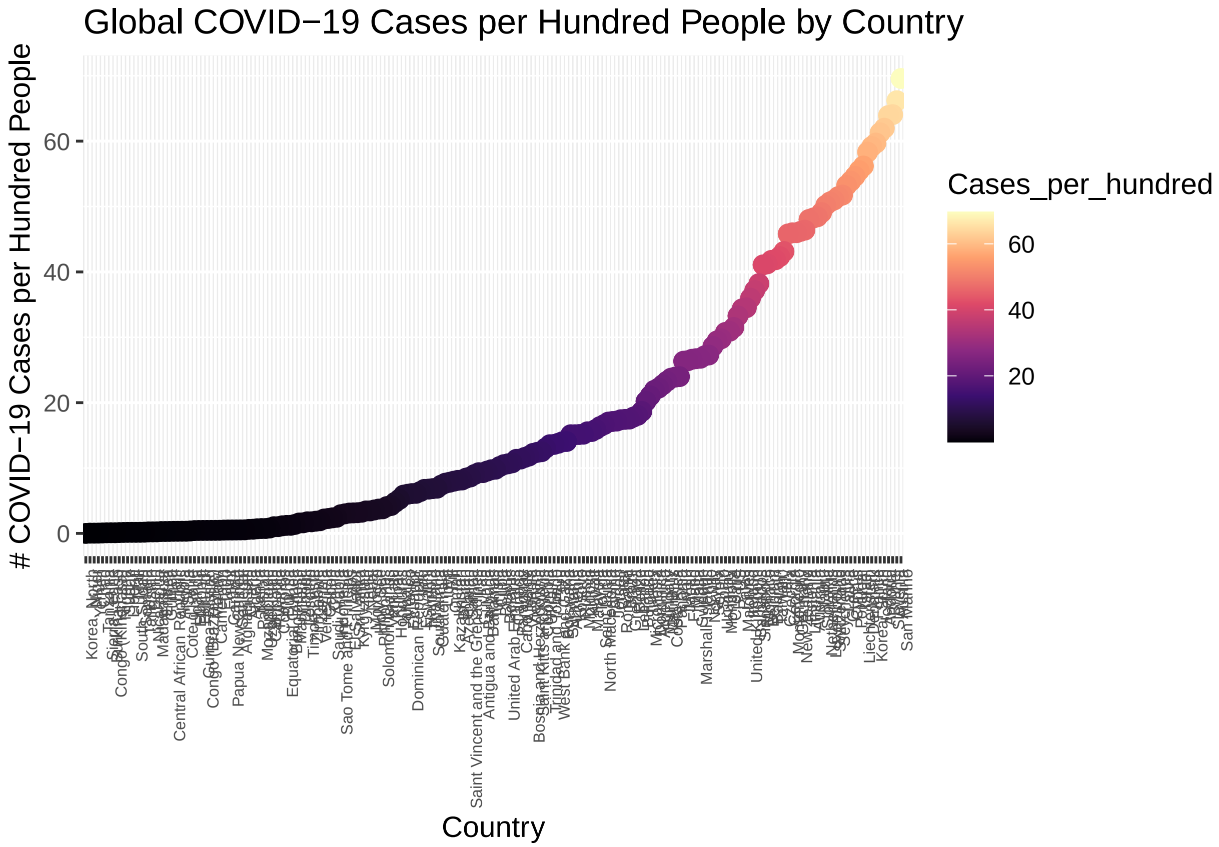
```r
US_vaccinations <- mutate_at(US_vaccinations, vars('Province_State'), as.factor)

# Create 'US_by_state_vaccinations_per_hundred' df holding max vaccination rates per US state
US_by_state_vaccinations_per_hundred <- US_vaccinations %>%
  group_by(Province_State) %>%
  mutate(Province_State = fct_recode(Province_State,
    "New York" = "New York State")) %>%
  summarize(people_fully_vaccinated_per_hundred = max(people_fully_vaccinated_per_hundred, na.rm = T),
            total_vaccinations_per_hundred = max(total_vaccinations_per_hundred, na.rm = T),
            people_vaccinated_per_hundred = max(people_vaccinated_per_hundred, na.rm = T),
            distributed_per_hundred = max(distributed_per_hundred, na.rm = T),
            total_boosters_per_hundred = max(total_boosters_per_hundred, na.rm = T))

# Merge 'US_by_state_deaths_per_hundred' df and 'US_by_state_vaccinations_per_hundred' df
US_by_state_deaths_vaccinations_per_hundred <- US_by_state_deaths_per_hundred %>%
    full_join(US_by_state_vaccinations_per_hundred) %>%
  filter(Population > 0)
```

```
## Joining with ‘by = join_by(Province_State)‘
```

### Step 3: Visualization

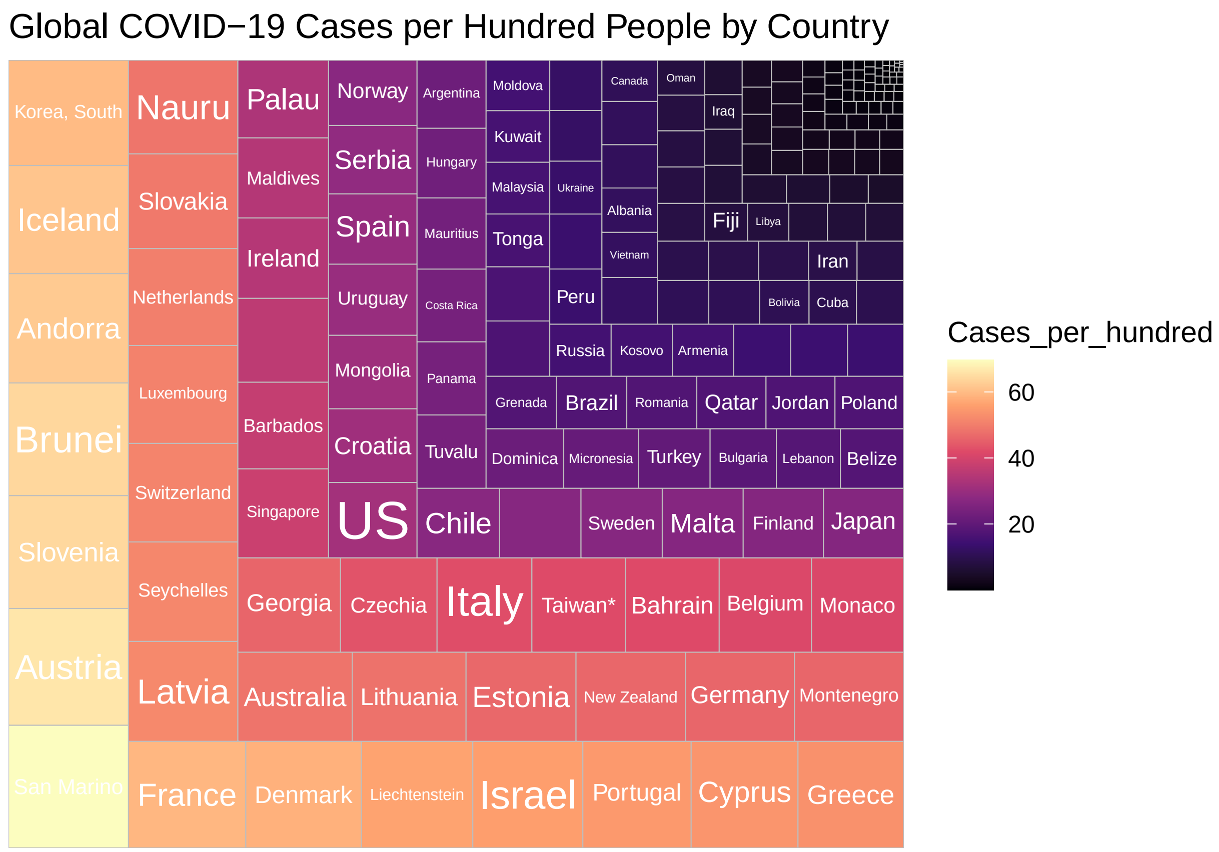
## On this plot i used a dot plot to reduce clutter and make it easier to compare values.

```r
ggplot(global_cases_per_hundred, aes(x = reorder(Country_Region, +Cases_per_hundred), y = Cases_per_hund
  geom_point(aes(color = Cases_per_hundred), size = 3) +
  scale_color_viridis_c(option = "magma") +
  labs(x = "Country", y = "# COVID-19 Cases per Hundred People", title = "Global COVID-19 Cases per Hund
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 6))
```

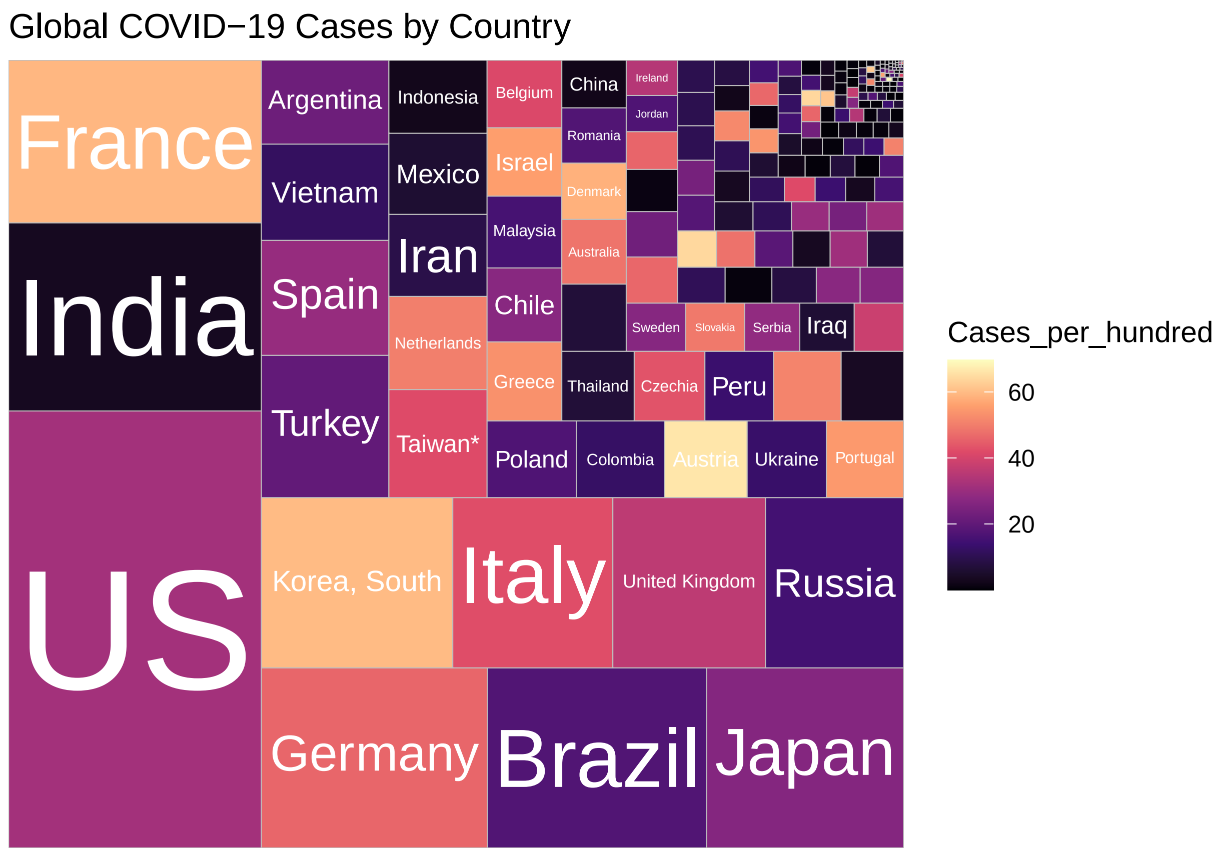


## Treemap can provide a visual representation of the data in a hierachical manner. A treemap could show each country as a rectangle, with the size of the rectangle representing the number of cases. Countries with more cases would have larger rectangles. If you add another layer, like continents, the treemap would first show rectangles for each continent, and within each continent, rectangles for each country. In this project cases per Hundred People it is a normalized metric that shows the number of COVID-19 cases per hundred people in the population. It helps to compare the impact of COVID-19 across countries with different population sizes.One might ask why Latvia smaller country has a bigger rectangle than the US? The answer is because the treemap is visualizing the normalized metric (cases per hundred people) rather than the absolute number of cases. I have added for your understanding the absolute number case plot right after this one.

```r
library(treemapify)
ggplot(global_cases_per_hundred, aes(area = Cases_per_hundred, fill = Cases_per_hundred, label = Countr
  geom_treemap() +
  geom_treemap_text(colour = "white", place = "centre", grow = TRUE) +
  scale_fill_viridis_c(option = "magma") +
  labs(title = "Global COVID-19 Cases per Hundred People by Country")
```

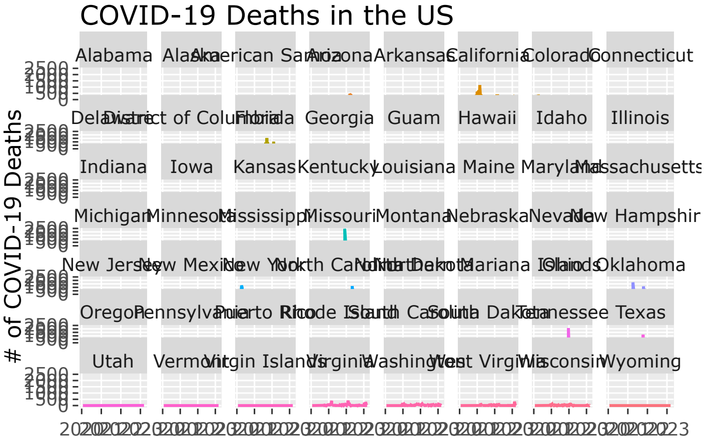


```r
ggplot(global_cases_per_hundred, aes(area = Cases, fill = Cases_per_hundred, label = Country_Region)) +
  geom_treemap() +
  geom_treemap_text(colour = "white", place = "centre", grow = TRUE) +
  scale_fill_viridis_c(option = "magma") +
  labs(title = "Global COVID-19 Cases by Country")
```



**Using plotly pacakge can make our plot interactive and alowing to zoom in and out.**

```
library(plotly)
p <- ggplot(US_by_state_cases_deaths_per_day, aes(x = Date, y = New_Deaths, color = Province_State)) +
  geom_line() +
  facet_wrap(~Province_State) +
  labs(x = "", y = "# of COVID-19 Deaths", title = "COVID-19 Deaths in the US", subtitle = "By State/Te
  theme(legend.position = "none")
ggplotly(p)
```

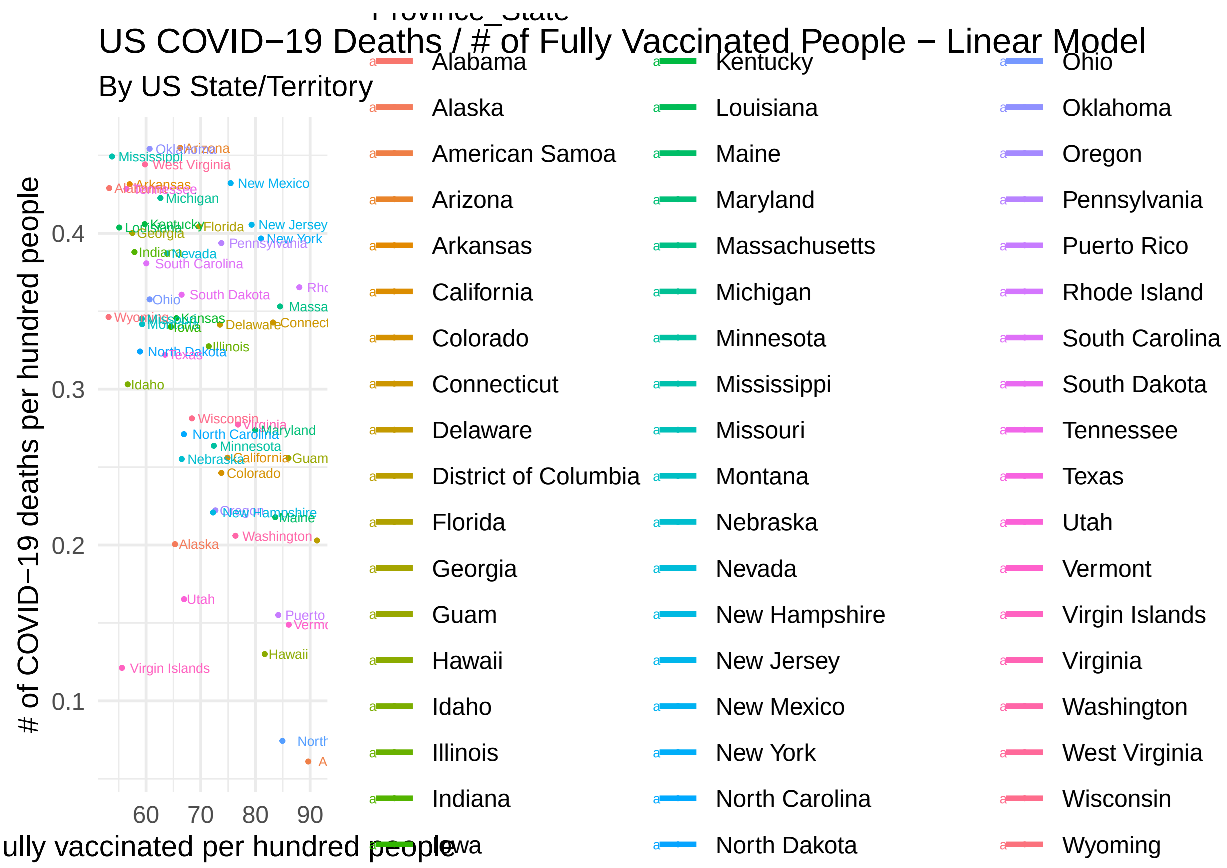

COVID-19 Deaths in the US
# of COVID-19 Deaths
Alabama
Alaska
American Samoa
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
District of Columbia
Florida
Georgia
Guam
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Mississippi
Missouri
Montana
Nebraska
Nevada
New Hampshir
New Jersey
New Mexico
New York
North Carolina
North Dakota
Northern Mariana Islands
Ohio
Oklahoma
Oregon
Pennsylvania
Puerto Rico
Rhode Island
South Carolina
South Dakota
Tennessee
Texas
Utah
Vermont
Virgin Islands
Virginia
Washington
West Virginia
Wisconsin
Wyoming
2500
2000
1500
1000
500
0
2020
2021
2022
2023

### Here I am trying to build different model.

##We will build a model based on cases per 1000 and deaths per 1000, output the summary, then add the predictions to the Mass. county data.

```
ggplot(data = US_by_state_deaths_vaccinations_per_hundred, aes(x = people_fully_vaccinated_per_hundred,
                                                                y = Deaths_per_hundred,
                                                                color = Province_State,  # Color by stat
                                                                label = Province_State)) +
  geom_point(size = .4) +
  geom_text(size = 1.7, vjust = .5, hjust = -.1) +
  geom_smooth(method = "lm", se = FALSE) +  # Add linear regression line without confidence interval
  labs(x = "# fully vaccinated per hundred people",
       y = "# of COVID-19 deaths per hundred people",
       title = "US COVID-19 Deaths / # of Fully Vaccinated People - Linear Model",
       subtitle = "By US State/Territory") +
  theme_minimal()  # Use a minimal theme for better readability
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```
cases <- confirmed_us %>%
  pivot_longer(cols = -c(UID:Combined_Key), names_to = "date", values_to = "Cases")%>%
  select(-c(iso2, iso3, code3, FIPS, UID, Country_Region))%>%
  mutate(date = mdy(date))
```

```r
summary(cases)
```

```
##     Admin2         Province_State          Lat             Long_
##  Length:3819906   Length:3819906   Min.   :-14.27   Min.   :-174.16
##  Class :character   Class :character   1st Qu.: 33.90   1st Qu.: -97.81
##  Mode  :character   Mode  :character   Median : 38.01   Median : -89.49
##                                          Mean   : 36.72   Mean   : -88.64
##                                          3rd Qu.: 41.58   3rd Qu.: -82.31
##                                          Max.   : 69.31   Max.   : 145.67
##  Combined_Key            date                Cases
##  Length:3819906   Min.   :2020-01-22   Min.   :  -3073
##  Class :character   1st Qu.:2020-11-02   1st Qu.:    330
##  Mode  :character   Median :2021-08-15   Median :   2272
##                     Mean   :2021-08-15   Mean   :  14088
##                     3rd Qu.:2022-05-28   3rd Qu.:   8159
##                     Max.   :2023-03-09   Max.   :3710586
```

```r
deaths <- deaths_us %>%
  pivot_longer(cols = -c(UID:Population), names_to = "date", values_to = "deaths")%>%
  select(-c(iso2, iso3, code3, FIPS, UID, Country_Region))%>%
  mutate(date = mdy(date))

summary(deaths)
```

```
##     Admin2         Province_State          Lat             Long_
##  Length:3819906   Length:3819906   Min.   :-14.27   Min.   :-174.16
##  Class :character   Class :character   1st Qu.: 33.90   1st Qu.: -97.81
##  Mode  :character   Mode  :character   Median : 38.01   Median : -89.49
##                                          Mean   : 36.72   Mean   : -88.64
##                                          3rd Qu.: 41.58   3rd Qu.: -82.31
##                                          Max.   : 69.31   Max.   : 145.67
##  Combined_Key         Population            date                deaths
##  Length:3819906   Min.   :       0   Min.   :2020-01-22   Min.   :  -82.0
##  Class :character   1st Qu.:    9917   1st Qu.:2020-11-02   1st Qu.:    4.0
##  Mode  :character   Median :   24892   Median :2021-08-15   Median :   37.0
##                     Mean   :   99604   Mean   :2021-08-15   Mean   :  186.9
##                     3rd Qu.:   64979   3rd Qu.:2022-05-28   3rd Qu.:  122.0
##                     Max.   :10039107   Max.   :2023-03-09   Max.   :35545.0
```

```r
Mass_Cases <- cases %>%
  filter(Province_State == "Massachusetts")#%>%
#  group_by(Admin2)

Mass_Deaths <- deaths %>%
  filter(Province_State == "Massachusetts")#%>%
#  group_by(Admin2)

All_Mass <- Mass_Cases %>%
  full_join(Mass_Deaths)
```

```
## Joining with ‘by = join_by(Admin2, Province_State, Lat, Long_, Combined_Key,
## date)‘
```

```
Mass <- All_Mass %>%
  mutate(deaths_per_k= deaths * 1000 / Population, cases_per_k= Cases * 1000 / Population, month_year =

summary(Mass)
```

```
##     Admin2          Province_State          Lat             Long_
##  Length:16002       Length:16002       Min.   :41.29   Min.   :-73.21
##  Class :character   Class :character   1st Qu.:41.79   1st Qu.:-72.59
##  Mode  :character   Mode  :character   Median :42.24   Median :-71.16
##                                        Mean   :42.11   Mean   :-71.47
##                                        3rd Qu.:42.37   3rd Qu.:-70.81
##                                        Max.   :42.67   Max.   :-70.09
##  Combined_Key            date                Cases          Population
##  Length:16002       Min.   :2020-01-22   Min.   :     0   Min.   :  11399
##  Class :character   1st Qu.:2020-11-02   1st Qu.:  1475   1st Qu.: 124944
##  Mode  :character   Median :2021-08-15   Median : 23197   Median : 493787
##                     Mean   :2021-08-15   Mean   : 65074   Mean   : 492322
##                     3rd Qu.:2022-05-28   3rd Qu.:104131   3rd Qu.: 789034
##                     Max.   :2023-03-09   Max.   :437431   Max.   :1611699
##      deaths      deaths_per_k     cases_per_k       month_year
##  Min.   :   0   Min.   :0.0000   Min.   :  0.000   Length:16002
##  1st Qu.:  77   1st Qu.:0.7267   1st Qu.:  7.139   Class :character
##  Median : 683   Median :1.8719   Median : 75.010   Mode  :character
##  Mean   :1028   Mean   :1.7397   Mean   :106.850
##  3rd Qu.:1794   3rd Qu.:2.6503   3rd Qu.:203.338
##  Max.   :4822   Max.   :4.5843   Max.   :368.944
##       Lng             month
##  Min.   :-73.21   Min.   : 1.000
##  1st Qu.:-72.59   1st Qu.: 3.000
##  Median :-71.16   Median : 6.000
##  Mean   :-71.47   Mean   : 6.335
##  3rd Qu.:-70.81   3rd Qu.: 9.000
##  Max.   :-70.09   Max.   :12.000
```

```
head(Mass)
```

```
## # A tibble: 6 x 14
##   Admin2     Province_State   Lat Long_ Combined_Key date       Cases Population
##   <chr>      <chr>          <dbl> <dbl> <chr>        <date>     <dbl>      <dbl>
## 1 Barnstable Massachusetts   41.7 -70.3 Barnstable,~ 2020-01-22     0     212990
## 2 Barnstable Massachusetts   41.7 -70.3 Barnstable,~ 2020-01-23     0     212990
## 3 Barnstable Massachusetts   41.7 -70.3 Barnstable,~ 2020-01-24     0     212990
## 4 Barnstable Massachusetts   41.7 -70.3 Barnstable,~ 2020-01-25     0     212990
## 5 Barnstable Massachusetts   41.7 -70.3 Barnstable,~ 2020-01-26     0     212990
## 6 Barnstable Massachusetts   41.7 -70.3 Barnstable,~ 2020-01-27     0     212990
## # i 6 more variables: deaths <dbl>, deaths_per_k <dbl>, cases_per_k <dbl>,
## #   month_year <chr>, Lng <dbl>, month <dbl>
```

```
'Deaths & Population: '
```

```
## [1] "Deaths & Population: "
```

```r
cor(Mass$deaths, Mass$Population)
```

```
## [1] 0.7976191
```

```r
'Cases & Population: '
```

```
## [1] "Cases & Population: "
```

```r
cor(Mass$Cases, Mass$Population)
```

```
## [1] 0.6327786
```

```r
'Cases & Deaths: '
```

```
## [1] "Cases & Deaths: "
```

```r
cor(Mass$Cases, Mass$deaths)
```

```
## [1] 0.924681
```

```r
'Cases/1000 & Deaths/1000: '
```

```
## [1] "Cases/1000 & Deaths/1000: "
```

```r
cor(Mass$cases_per_k, Mass$deaths_per_k)
```

```
## [1] 0.91824
```

```r
Mass_Model <- lm(cases_per_k ~ deaths_per_k, Mass)
summary(Mass_Model)
```

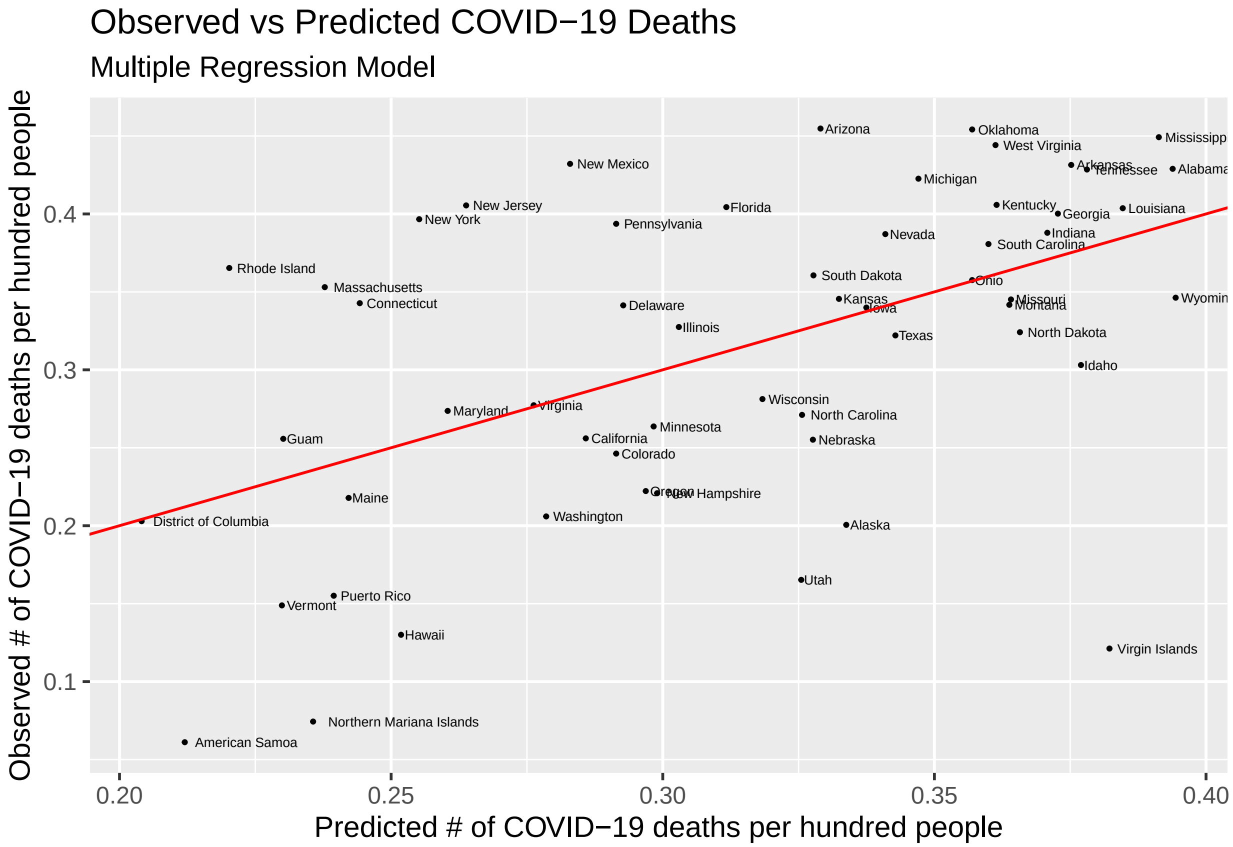
```
##
## Call:
## lm(formula = cases_per_k ~ deaths_per_k, data = Mass)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.130 -38.676   6.675  30.785 146.494
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -30.7845     0.5764  -53.41   <2e-16 ***
## deaths_per_k  79.1149     0.2698  293.29   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 42.33 on 16000 degrees of freedom
## Multiple R-squared:  0.8432, Adjusted R-squared:  0.8432
## F-statistic: 8.602e+04 on 1 and 16000 DF,  p-value: < 2.2e-16
```

```r
Mass_Pred <- Mass %>%
  mutate(Prediction = predict(Mass_Model))
```

```r
Mass_Model <- lm(Deaths_per_hundred ~ people_fully_vaccinated_per_hundred,
            data = US_by_state_deaths_vaccinations_per_hundred)
summary(Mass_Model)
```

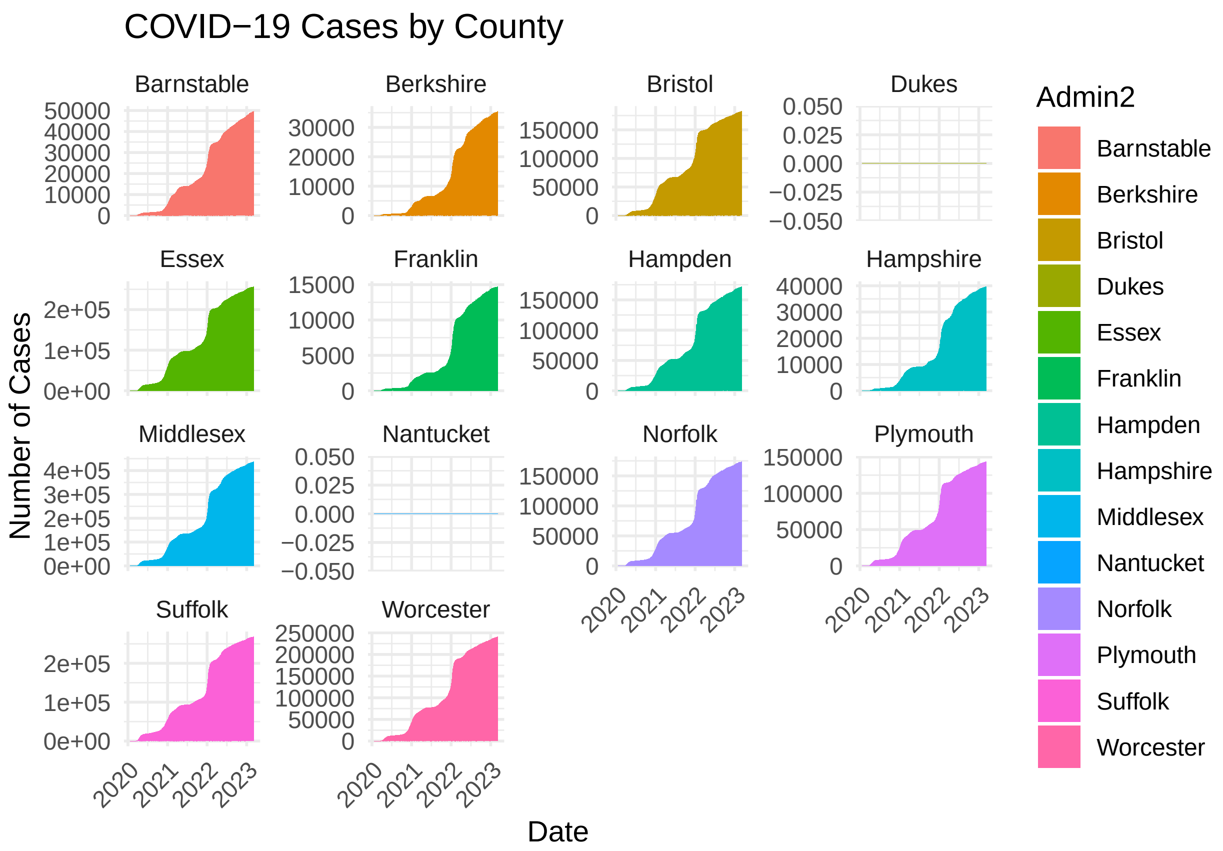
```
##
## Call:
## lm(formula = Deaths_per_hundred ~ people_fully_vaccinated_per_hundred,
##     data = US_by_state_deaths_vaccinations_per_hundred)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.261032 -0.049752  0.007771  0.051880  0.149193
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          0.659862   0.076443   8.632 9.54e-12 ***
## people_fully_vaccinated_per_hundred -0.004994   0.001088  -4.590 2.68e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.08772 on 54 degrees of freedom
## Multiple R-squared:  0.2807, Adjusted R-squared:  0.2674
## F-statistic: 21.07 on 1 and 54 DF,  p-value: 2.68e-05
```

```r
US_by_state_deaths_vaccinations_per_hundred$predicted_deaths <- predict(Mass_Model, US_by_state_deaths_

ggplot(data = US_by_state_deaths_vaccinations_per_hundred, aes(x = predicted_deaths, y = Deaths_per_hun
  geom_point(size = .4) +
  geom_text(size = 1.7, vjust = .5, hjust = -.1) +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(x = "Predicted # of COVID-19 deaths per hundred people",
       y = "Observed # of COVID-19 deaths per hundred people",
       title = "Observed vs Predicted COVID-19 Deaths",
       subtitle = "Multiple Regression Model")
```



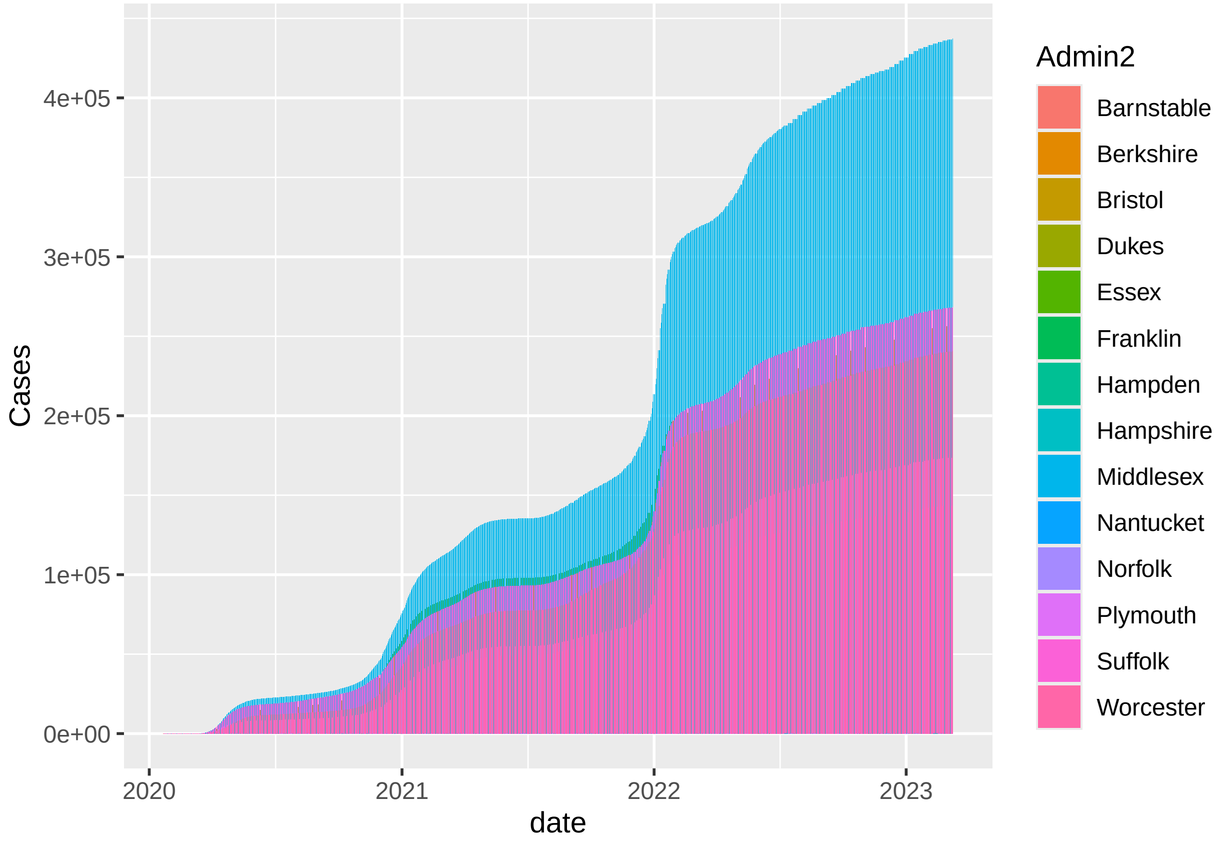
##Using a bar plot (geom_col()) to visualize predictions for individual counties can make sense, especially if you want to compare the number of cases across different counties on specific dates. However, there are a few considerations to ensure the plot is clear and informative:

```r
ggplot(Mass_Pred, aes(x = date, y = Cases, fill = Admin2)) +
  geom_col(position = "dodge") +
  facet_wrap(~ Admin2, scales = "free_y") +
  labs(x = "Date", y = "Number of Cases", title = "COVID-19 Cases by County") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

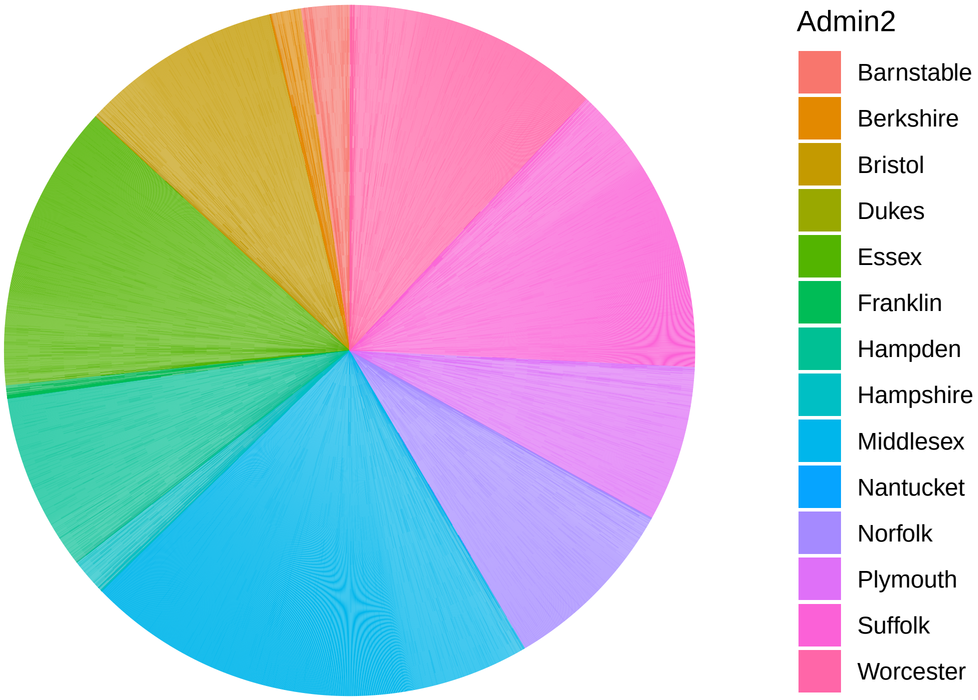


##Using a bar plot (geom_col()) to visualize predictions for individual counties can make sense, especially if you want to compare the number of cases across different counties on specific dates.

```r
ggplot(Mass_Pred, aes(x=date, y=Cases, fill=Admin2)) +
  geom_col(position = "dodge")
```



```
ggplot(Mass_Pred, aes(x="", y=Cases, fill=Admin2)) +
  geom_bar(stat="identity", width=1) +
  coord_polar(theta="y") +
  theme_void()
```



```r
Mass_Model <- lm(cases_per_k ~ deaths_per_k, Mass)
summary(Mass_Model)
```

```
##
## Call:
## lm(formula = cases_per_k ~ deaths_per_k, data = Mass)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.130 -38.676   6.675  30.785 146.494
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -30.7845     0.5764  -53.41   <2e-16 ***
## deaths_per_k  79.1149     0.2698  293.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.33 on 16000 degrees of freedom
## Multiple R-squared:  0.8432, Adjusted R-squared:  0.8432
## F-statistic: 8.602e+04 on 1 and 16000 DF,  p-value: < 2.2e-16
```

```r
Mass_Pred <- Mass %>%
  mutate(Prediction = predict(Mass_Model))
```

```
Mass_County <- Mass %>%
  group_by(Admin2)%>%
  summarize(Max_Deaths=max(deaths), Total_Deaths = sum(deaths),Max_Cases = max(Cases), Total_Cases = sum

summary(Mass_County)
```

```
##     Admin2            Max_Deaths      Total_Deaths       Max_Cases
##  Length:14          Min.   :   0.0   Min.   :      0   Min.   :     0
##  Class :character   1st Qu.: 459.8   1st Qu.: 276102   1st Qu.: 36522
##  Mode  :character   Median :2050.0   Median :1382960   Median :157826
##                     Mean   :1735.9   Mean   :1175195   Mean   :143892
##                     3rd Qu.:2498.8   3rd Qu.:1665867   3rd Qu.:226117
##                     Max.   :4822.0   Max.   :3378924   Max.   :437431
##   Total_Cases          Population
##  Min.   :        0   Min.   :  11399
##  1st Qu.: 15948862   1st Qu.: 133916
##  Median : 81161396   Median : 493787
##  Mean   : 74379791   Mean   : 492322
##  3rd Qu.:118428435   3rd Qu.: 768469
##  Max.   :220834357   Max.   :1611699
```

```
head(Mass_County)
```

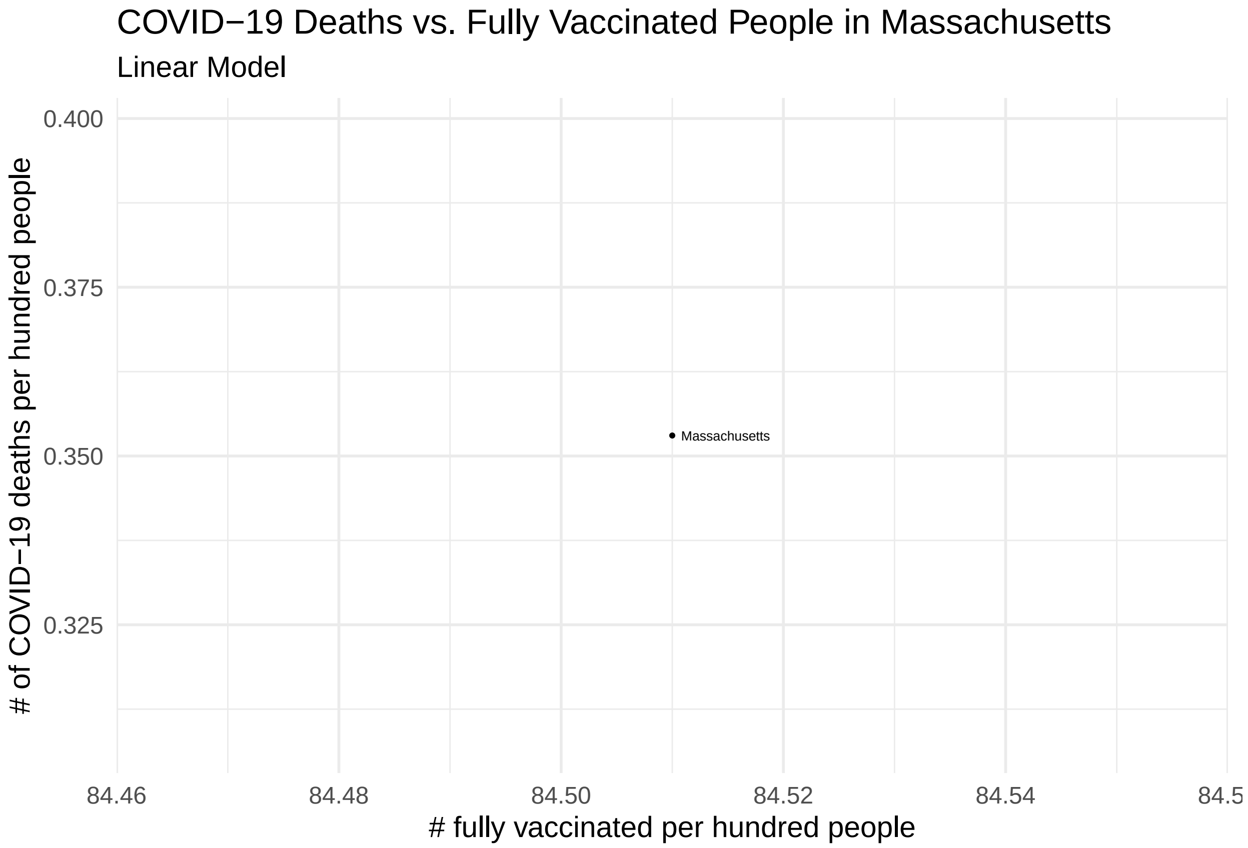
```
## # A tibble: 6 x 6
##   Admin2     Max_Deaths Total_Deaths Max_Cases Total_Cases Population
##   <chr>           <dbl>        <dbl>     <dbl>       <dbl>      <dbl>
## 1 Barnstable        785       447242     49617    23514236     212990
## 2 Berkshire         480       276606     35456    15223406     124944
## 3 Bristol          2555      1619263    182344    98339486     565217
## 4 Dukes               0            0         0           0      17332
## 5 Essex            3272      2235421    256987   140031284     789034
## 6 Franklin          198       108143     14736     6453660      70180
```

```
massachusetts_data <- US_by_state_deaths_vaccinations_per_hundred %>%
  filter(Province_State == "Massachusetts")
```

**To assess whether the data is unbiased, I would need to consider the data collection methods, potential confounding variables, and whether the data accurately represents the population without systematic errors.**

```
ggplot(data = massachusetts_data, aes(x = people_fully_vaccinated_per_hundred,
                                      y = Deaths_per_hundred,
                                      label = Province_State)) +
  geom_point(size = .4) +
  geom_text(size = 1.7, vjust = .5, hjust = -.1) +
  geom_smooth(method = "lm") +
  labs(x = "# fully vaccinated per hundred people",
       y = "# of COVID-19 deaths per hundred people",
       title = "COVID-19 Deaths vs. Fully Vaccinated People in Massachusetts",
       subtitle = "Linear Model") +
  theme_minimal()
```

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



**Conclusion**

##Our prediction managed to show multiple regression and linear regression. Regression is a powerful tool for predictive analytics. Linear is fundamental and widely used in predictive analytics. Linear regression relies on several assumptions, including linearity, independence, homoscedasticity (constant variance of errors), and normality of errors.

---

## Step 5

**Personal Bias**

To assess whether the data is unbiased, you would need to consider the data collection methods, potential confounding variables, and whether the data accurately represents the population without systematic errors. A dot in the middle of the plot doesn't necessarily prove that the data is unbiased. It simply represents a specific observation where the number of fully vaccinated people and the number of deaths fall around the middle range of your dataset.