Covid19Study2 Un-Vacine for Red-state Deaths compare to blue-states deaths

Hao Do

10/10/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## Getting dataset from John Hopkin University on Github

url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi

files_names <- c("time_series_covid19_confirmed_US.csv" ,
                 "time_series_covid19_deaths_US.csv" ,
                 "time_series_covid19_confirmed_global.csv" ,
                 "time_series_covid19_deaths_global.csv")

urls <- str_c(url_in,files_names)

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

```
## Rows: 3342 Columns: 639
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (633): 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[2])
```

```
## Rows: 3342 Columns: 640
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (634): 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.
```

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

```
## Rows: 279 Columns: 632
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (630): 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[4])
```

```
## Rows: 279 Columns: 632
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (630): 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.
## Cleaning up the data

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

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

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

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

```
global <- global %>% filter(cases >0)


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

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

US <- US_cases %>%
  full_join(US_deaths)
```

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

```
global <- global %>%
  unite("Combined_Key",
        c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)
```

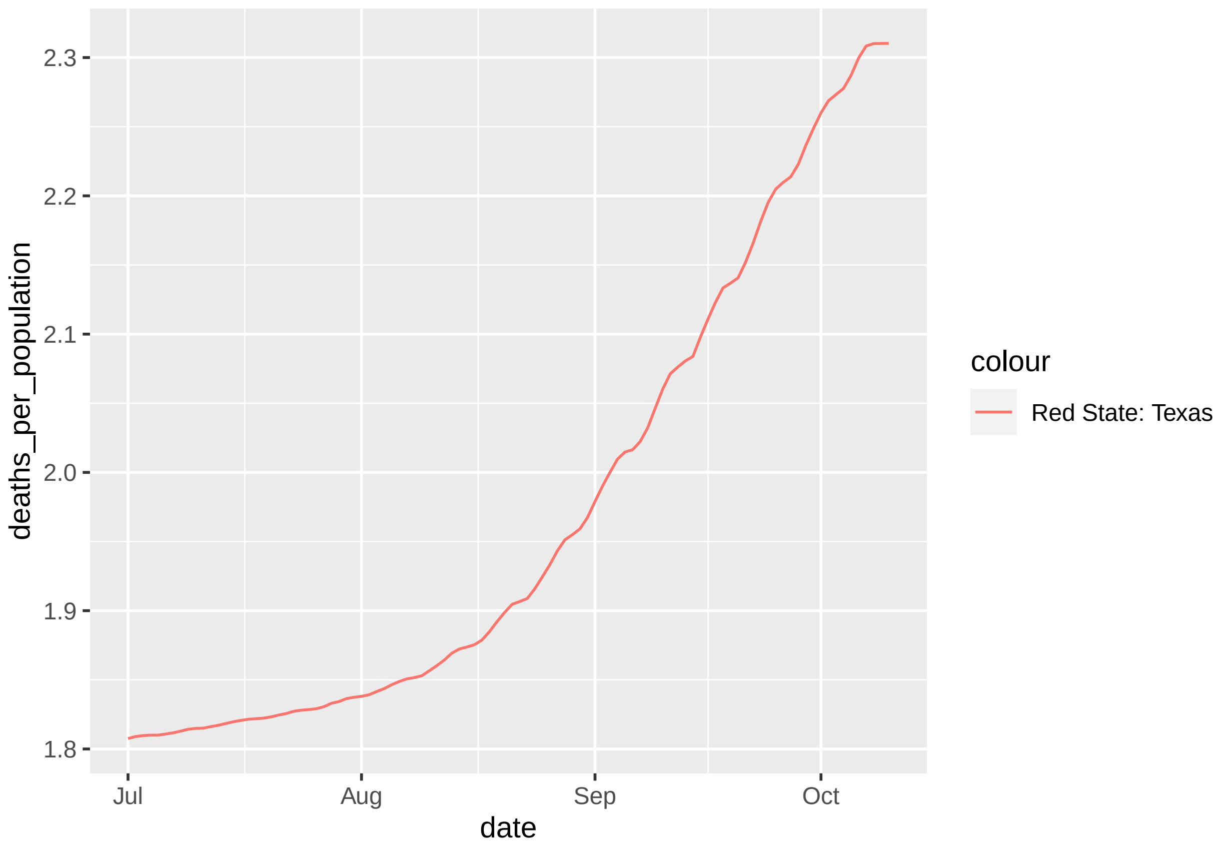
```
## Get the Population

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

```
## Rows: 4213 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.
```

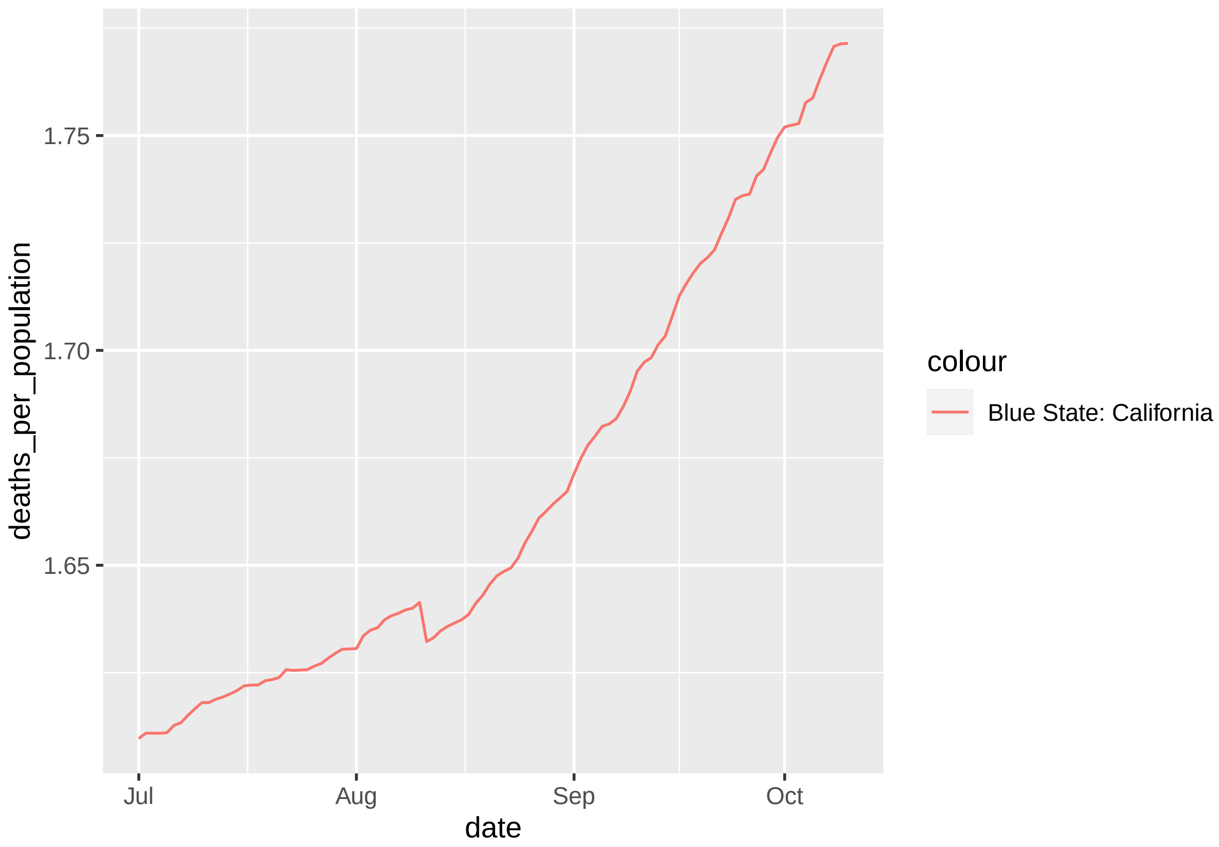
```
#Cleaning up the data

global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date, cases,deaths,Population,Combined_Key)

US_by_state <- US %>%
  filter(date >= as.Date("2021-07-01")) %>%
  group_by(Province_State, Country_Region, date)

state_total <- US_by_state %>%
  group_by(Province_State, date) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_population = 1000 * deaths / Population)
```

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

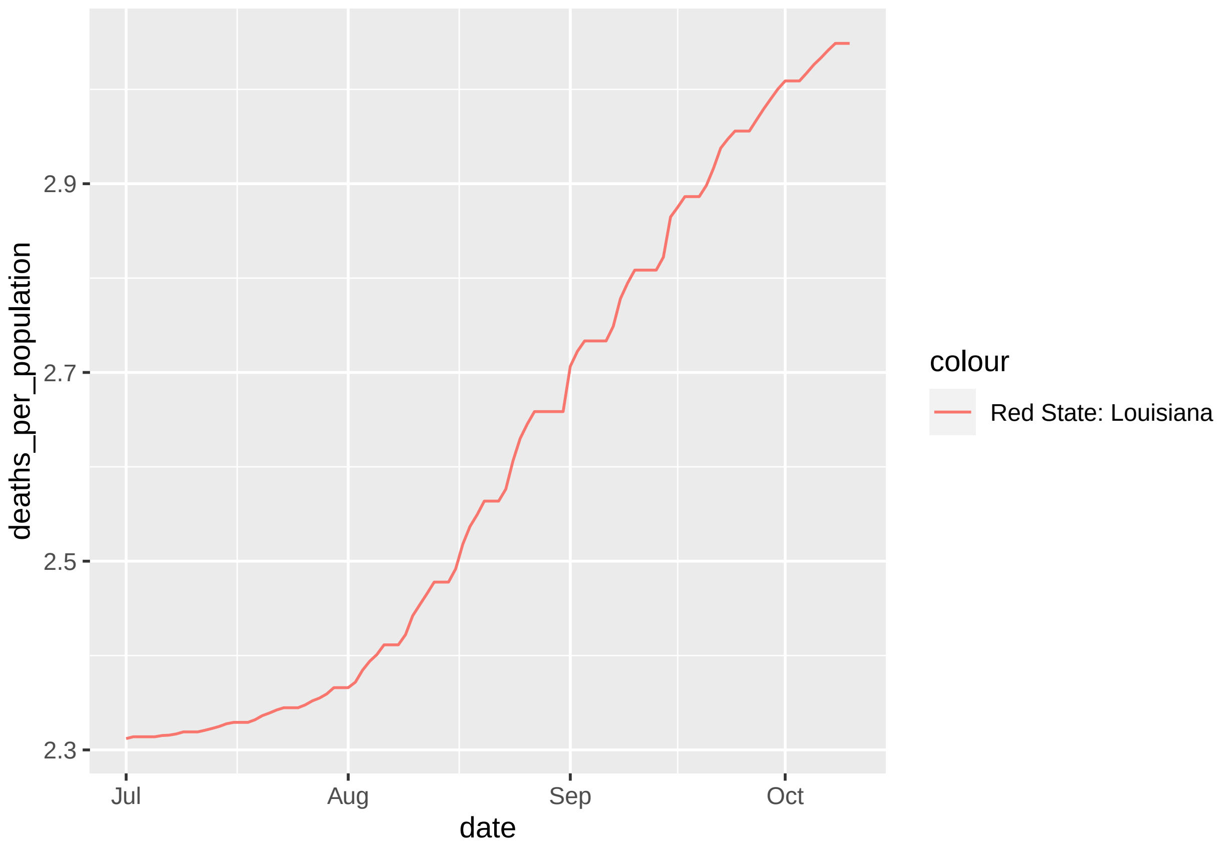
```
# Analize the Deaths per population between Red states and Blue States
red_state1 <- "Texas"
state_total %>%
filter(Province_State == red_state1) %>%
ggplot(aes(x=date , y = deaths_per_population))+
geom_line(aes(color = "Red State: Texas"))
```



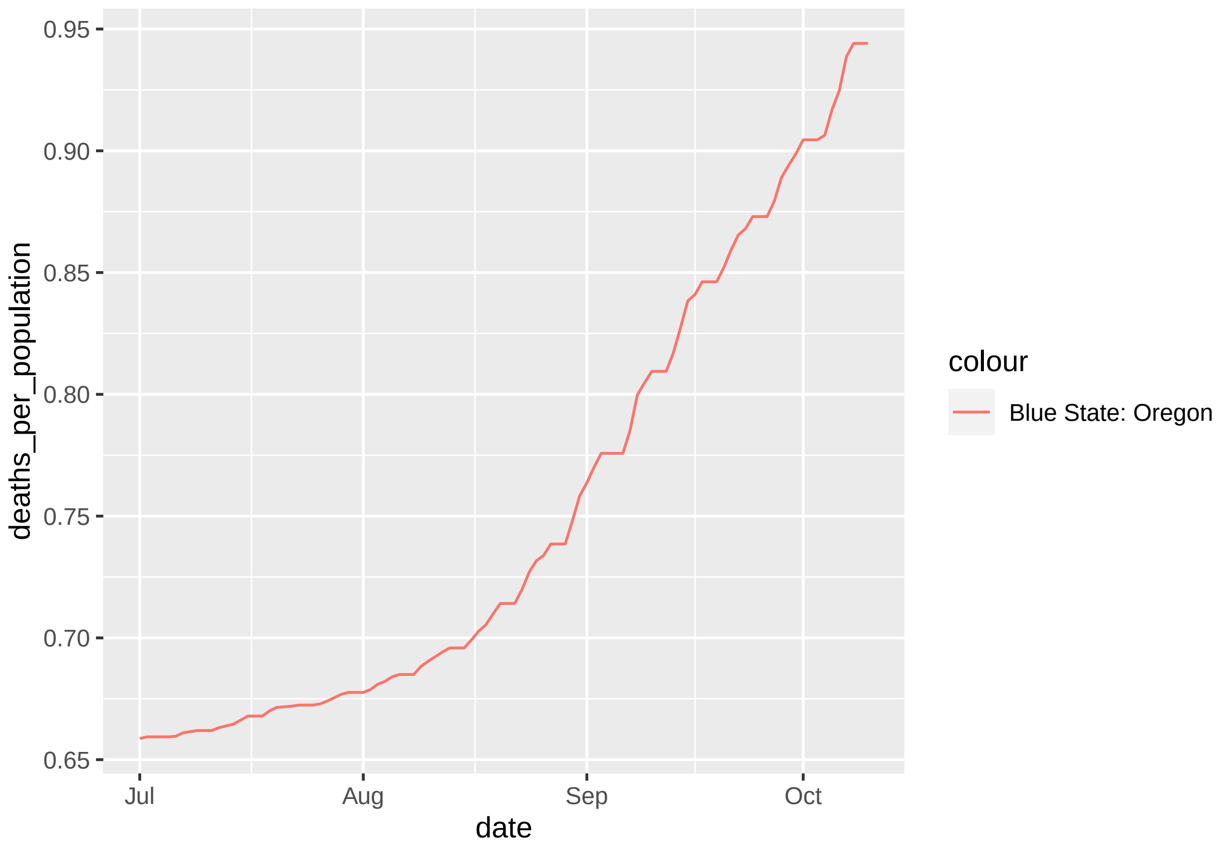
```
blue_state1 <- "California"
state_total %>%
  filter(Province_State == blue_state1) %>%
  ggplot(aes(x= date, y = deaths_per_population))+
  geom_line(aes(color = "Blue State: California"))
```



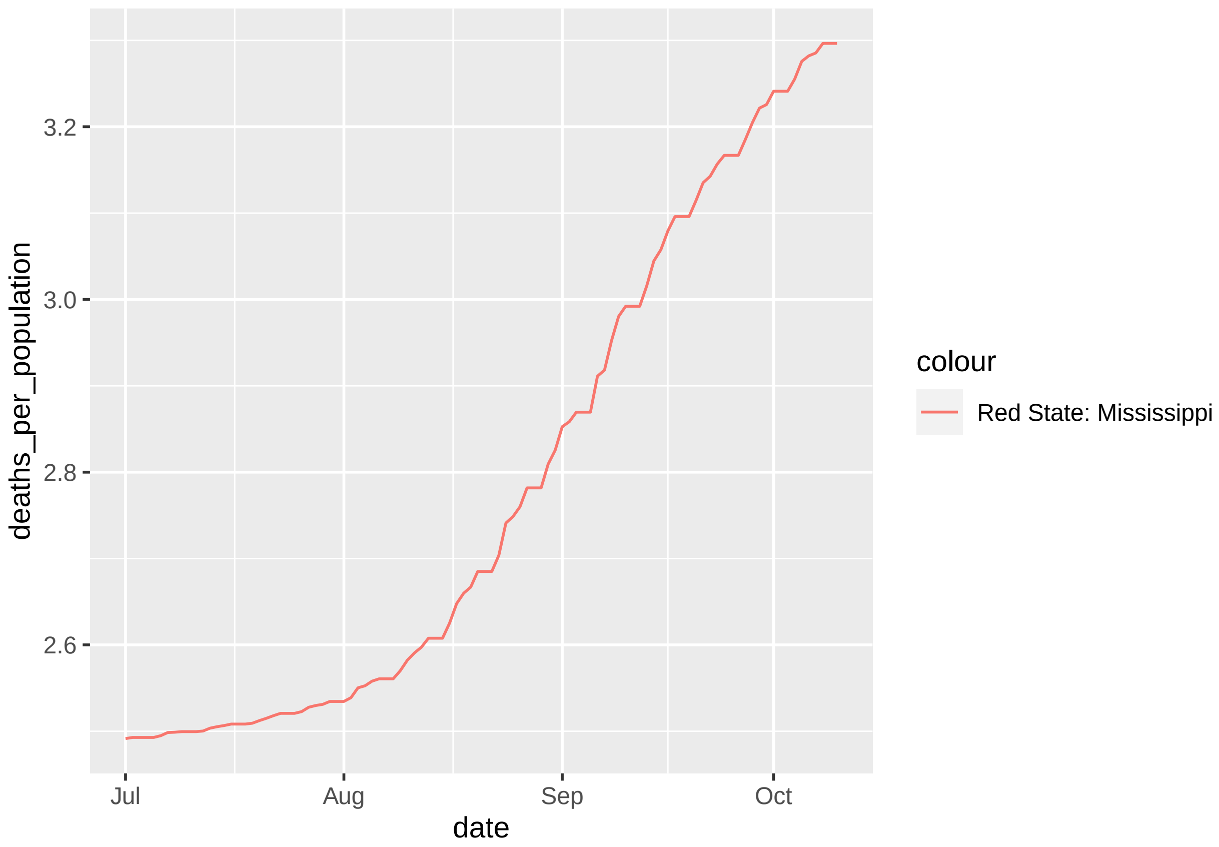
```
red_state2 <- "Louisiana"
state_total %>%
filter(Province_State == red_state2) %>%
ggplot(aes(x=date , y = deaths_per_population))+
geom_line(aes(color = "Red State: Louisiana"))
```



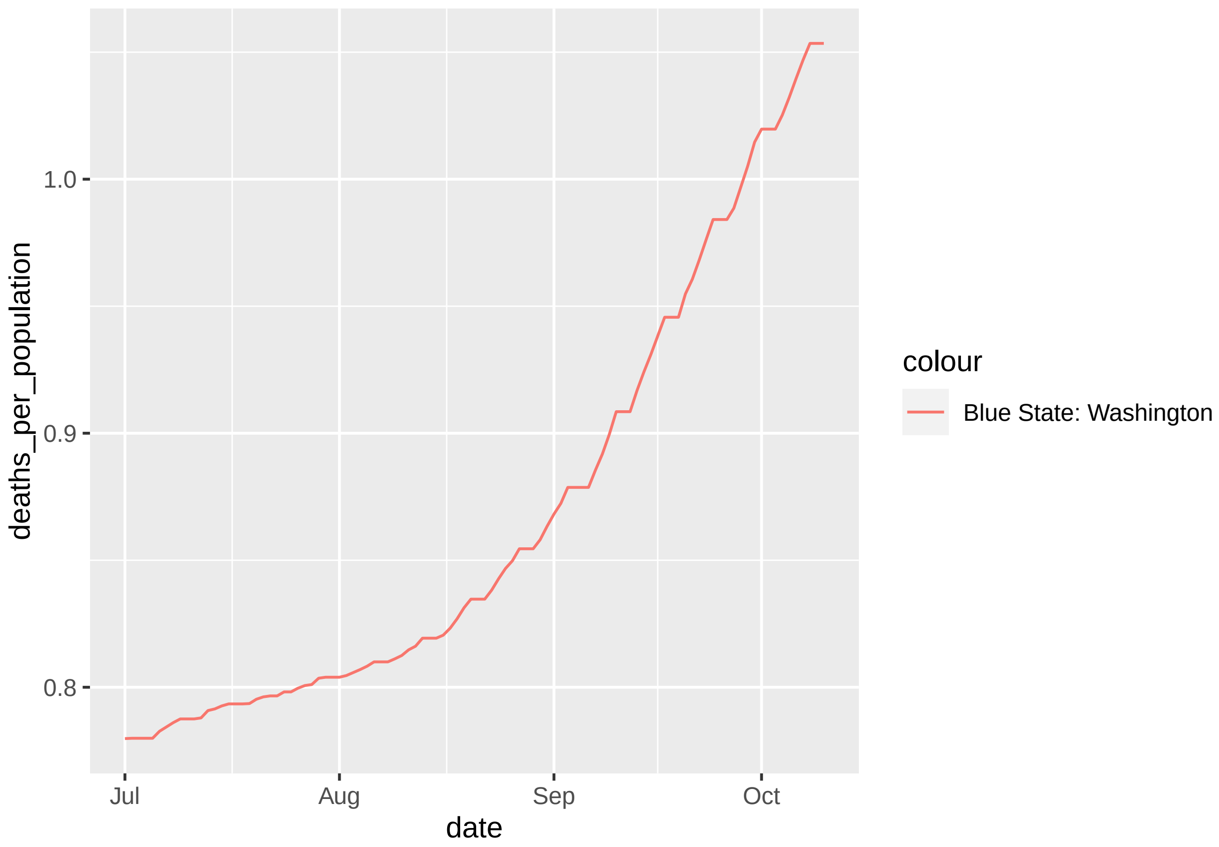
```
blue_state2 <- "Oregon"
state_total %>%
  filter(Province_State == blue_state2) %>%
  ggplot(aes(x= date, y = deaths_per_population))+
  geom_line(aes(color = "Blue State: Oregon"))
```



```
red_state3 <- "Mississippi"
state_total %>%
filter(Province_State == red_state3) %>%
ggplot(aes(x=date , y = deaths_per_population))+
geom_line(aes(color = "Red State: Mississippi"))
```



```
blue_state3 <- "Washington"
state_total %>%
  filter(Province_State == blue_state3) %>%
  ggplot(aes(x= date, y = deaths_per_population))+
  geom_line(aes(color = "Blue State: Washington"))
```


deaths_per_population
1.0
0.9
0.8
Jul
Aug
Sep
Oct
date
colour
Blue State: Washington