

jared\_is\_back2

## Import Longitude-Latitude Dataset

```
lats_long <- read.csv("https://raw.githubusercontent.com/albertyw/avenews/master/old/data/average-latitude-longitude.csv")

lats_long <- lats_long %>%
  rename(location = Country)
```

## Import John Hopkins Covid Dataset from 12/5/2021 named covid

```
covid <- read.csv("https://raw.githubusercontent.com/ST47S-CompStats-Fall2021/GroupJ-COVID/main/data/covid.csv")

# change date to date variable
covid <- covid %>%
  mutate(date = as.Date(date))

# add longitude-latitude to covid dataset
covid <- left_join(covid, lats_long, by = "location")
```

## Functions

```
filter_continents <- function(covid_data) {
  covid_data <- covid_data %>%
    filter(continent == "Asia" |
           continent == "Africa" |
           continent == "Europe" |
           continent == "North America" |
           continent == "Oceania" |
           continent == "South America" |
           continent == "Antarctica")
}
```

`filter_continents` : filter covid dataset with continent variable for only seven separate continents

```

filter_world <- function(covid_data) {
  covid_data <- covid_data %>%
    filter(location != "World" |
      location != "Asia" |
      location != "Africa" |
      location != "Europe" |
      location != "North America" |
      location != "Oceania" |
      location != "South America" |
      location != "Antarctica"
    )
}

```

`only_countries` : filter covid dataset with `continent` variable for only seven separate continents

```

replace_all_na <- function(covid_data) {
  covid_data %>%
    replace(is.na(.), 0)
}

```

`replace_all_na` : replaces all NA's in a dataframe with 0s

## Creating Sub-datasets

```

#select only numeric variables (outside of iso_code, location, and date)
covid_numeric <- cbind(
  covid %%
    select(iso_code, location, date, continent),
  covid %>%
    select_if(is.numeric)
)

```

`covid_numeric` : Only numeric variables (and iso\_code, location, date, continent)

`covid_per` : Only variables that containing `per` (and iso\_code, location, date, continent)

```

covid_per <- cbind(
  covid %%
    select(iso_code, location, date, continent, population),
  covid %>%
    select(contains("per"))
)

```

`covid_pm` : Only variables that end in per\_million (and iso\_code, location, date, continent)

```
covid_pm <- cbind(  
  covid %>%  
    select(iso_code, location, date, continent, population),  
  covid %>%  
    select(ends_with("per_million"))  
)
```

`covid_spm` : Only variables that end with smoothed\_per\_million (and iso\_code, location, date)

```
covid_spm <- cbind(  
  covid %>%  
    select(iso_code, location, date, continent),  
  covid %>%  
    select(ends_with("smoothed_per_million"))  
)
```

```
covid_spm_location <- covid_spm %>%  
  pivot_wider(names_from = date, values_from = !c(iso_code, date, location, continent)) %>%  
  replace_all_na()
```

`covid_spm_location` : (pivot wider)

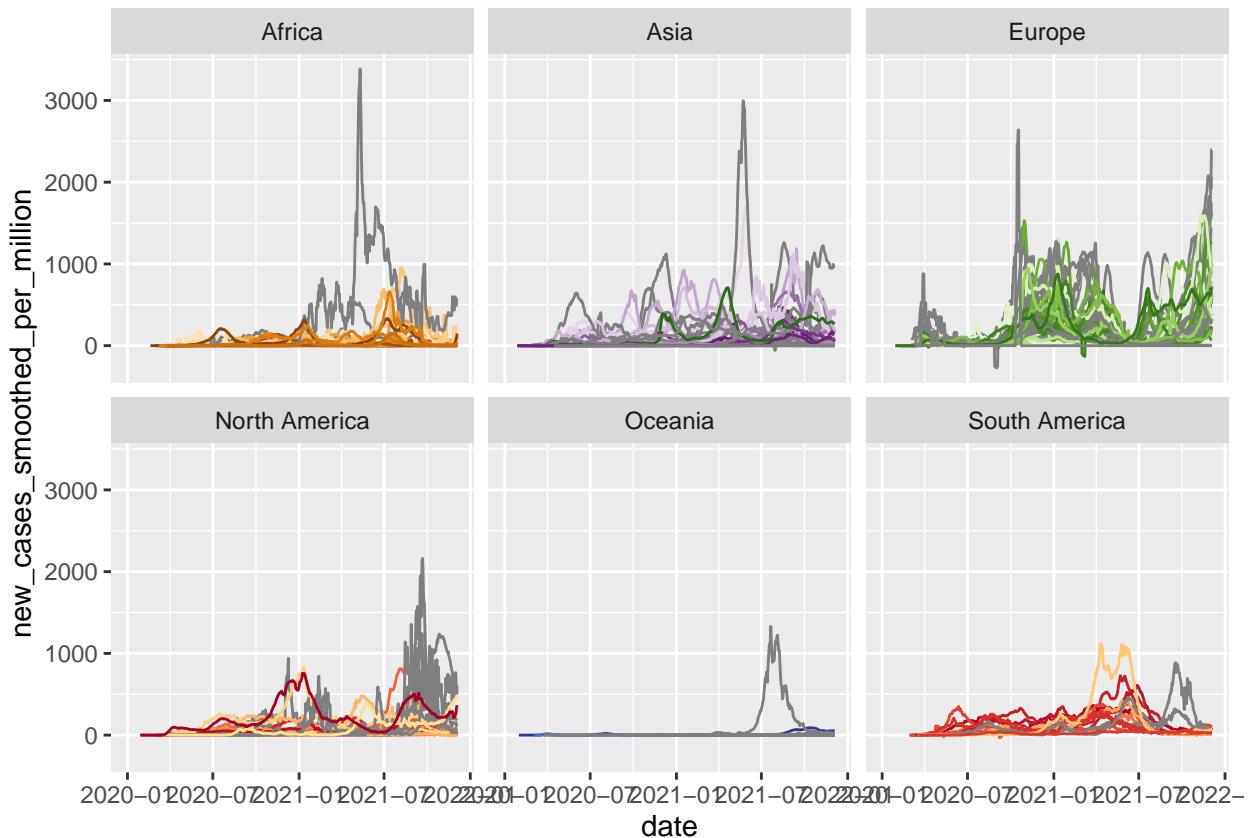
## EDA - Visualizations

```
covid_spm %>% head()
```

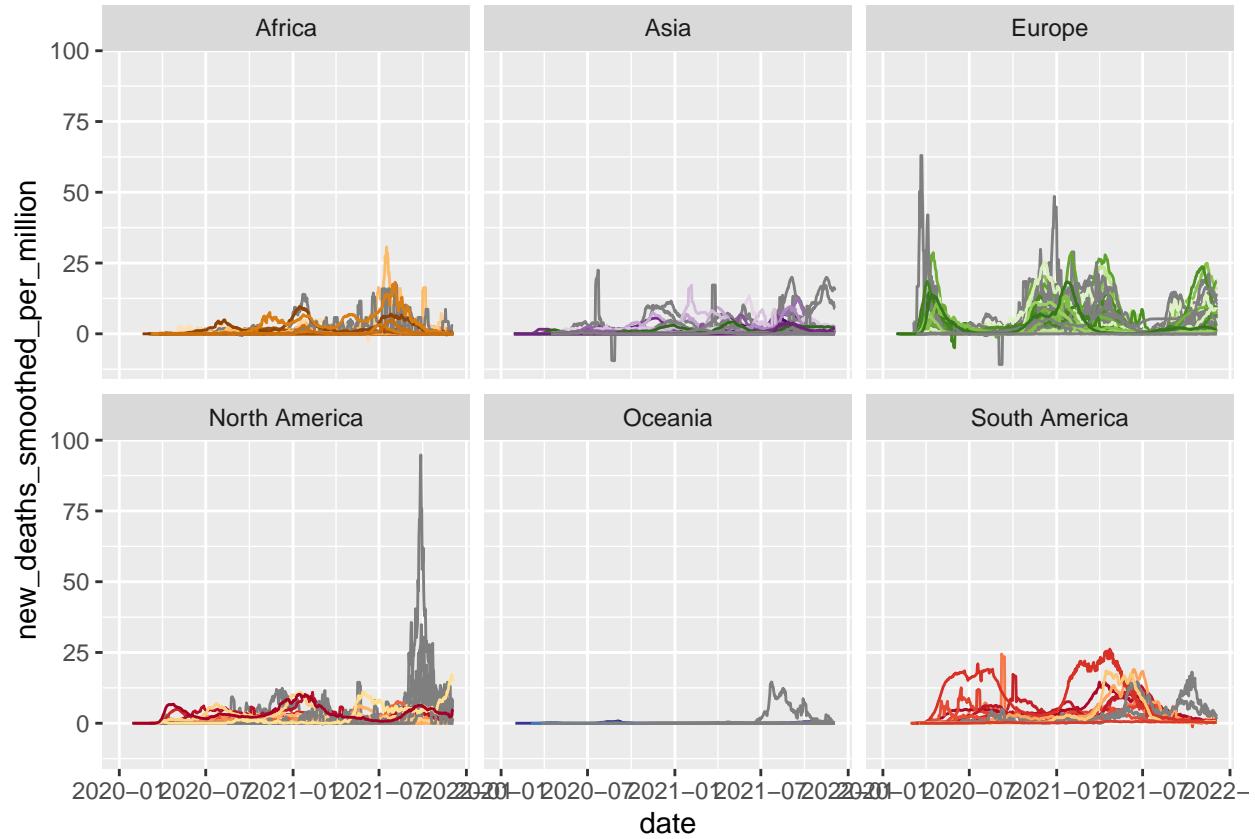
### New cases Smooth Per Million Over Time By Continent

```
##   iso_code      location      date continent new_cases_smoothed_per_million  
## 1     AFG Afghanistan 2020-02-24      Asia                 NA  
## 2     AFG Afghanistan 2020-02-25      Asia                 NA  
## 3     AFG Afghanistan 2020-02-26      Asia                 NA  
## 4     AFG Afghanistan 2020-02-27      Asia                 NA  
## 5     AFG Afghanistan 2020-02-28      Asia                 NA  
## 6     AFG Afghanistan 2020-02-29      Asia                0.018  
##   new_deaths_smoothed_per_million new_vaccinations_smoothed_per_million  
## 1                           NA                               NA  
## 2                           NA                               NA  
## 3                           NA                               NA  
## 4                           NA                               NA  
## 5                           NA                               NA  
## 6                           0                                NA
```

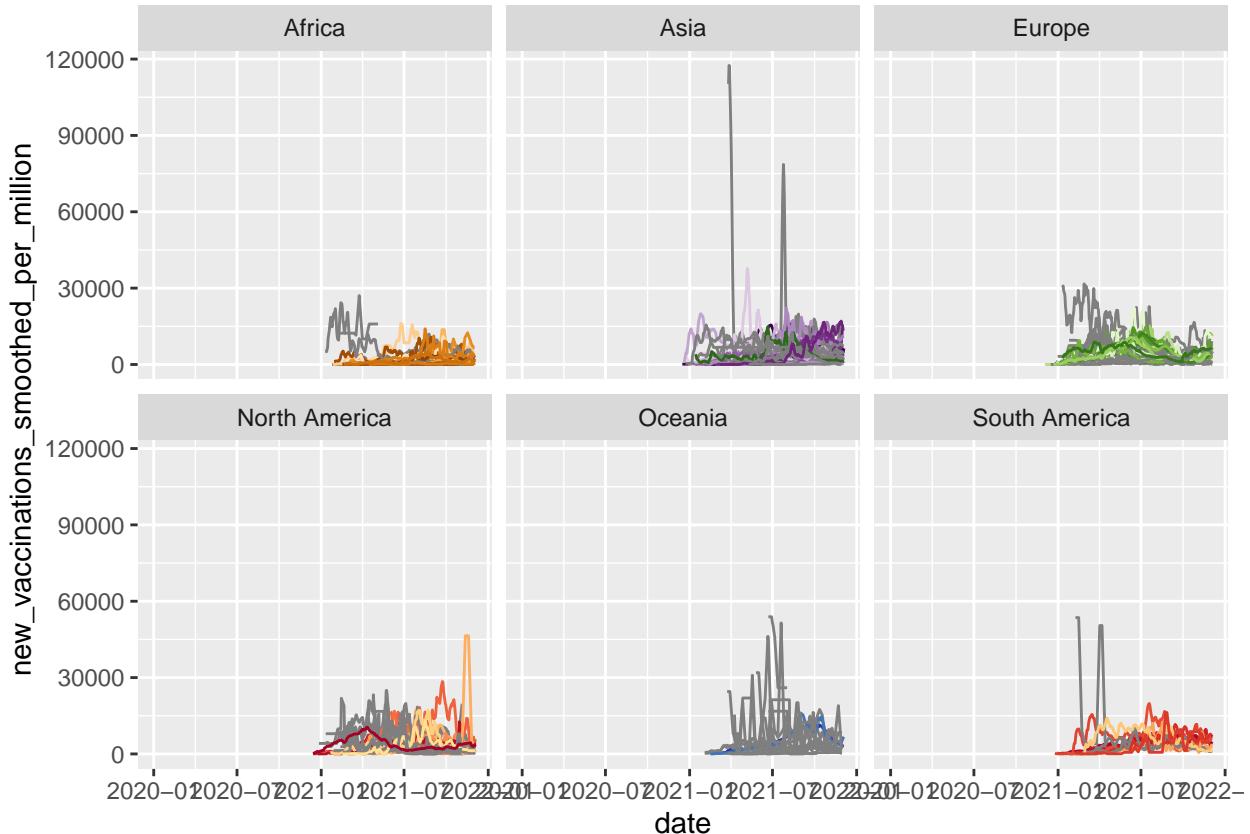
```
covid_spm %>%
  filter_continents %>%
  ggplot(aes(x = date, y = new_cases_smoothed_per_million, color=location)) +
  geom_line(aes(color = location), show.legend = FALSE) +
  facet_wrap(~continent) +
  scale_colour_manual(values = country_colors)
```



```
covid_spm %>%
  filter_continents %>%
  ggplot(aes(x = date, y = new_deaths_smoothed_per_million, color=location)) +
  geom_line(aes(color = location), show.legend = FALSE) +
  facet_wrap(~continent) +
  scale_colour_manual(values = country_colors)
```



```
covid_spm %>%
  filter_continents %>%
  ggplot(aes(x = date, y = new_vaccinations_smoothed_per_million, color=location)) +
  geom_line(aes(color = location), show.legend = FALSE) +
  facet_wrap(~continent) +
  scale_colour_manual(values = country_colors)
```



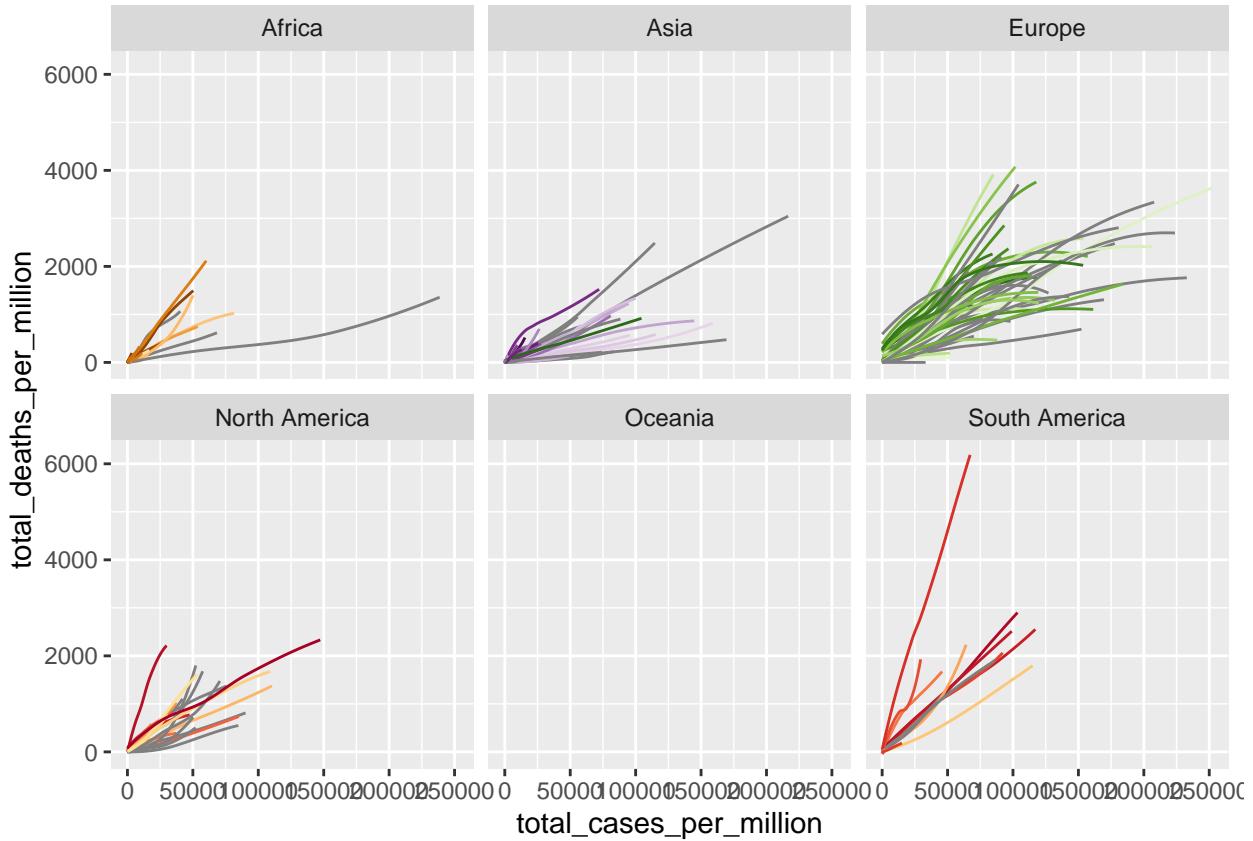
## Misc/Experiments

```
covid_spm1 <- covid_spm %>%
  replace_all_na() %>%
  filter_world()
```

```
covid_pm1 <- covid_pm %>%
  replace_all_na() %>%
  filter_world() %>%
  filter_continents()
```

```
ggplot(covid_pm1, aes(total_cases_per_million, total_deaths_per_million, color = location, size = population))
  #geom_point(alpha = 0.7, show.legend = FALSE) +
  geom_smooth(size=0.5, se=FALSE, show.legend=FALSE) +
  facet_wrap(~continent) +
  scale_color_manual(values = country_colors)
```

ganimate example



```

cases_death_animate <- ggplot(covid_pm1, aes(total_cases_per_million, total_deaths_per_million, color =
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_color_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  facet_wrap(~continent) +
  labs(title = 'Date: {frame_time}', x = 'Total Cases per Million', y = 'Total Deaths per Million') +
  transition_time(date) +
  ease_aes('linear') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))

animate(cases_death_animate, duration = 10)

```

## gganimate experiment

```
world <- map_data("world")
```

```

map <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),

```

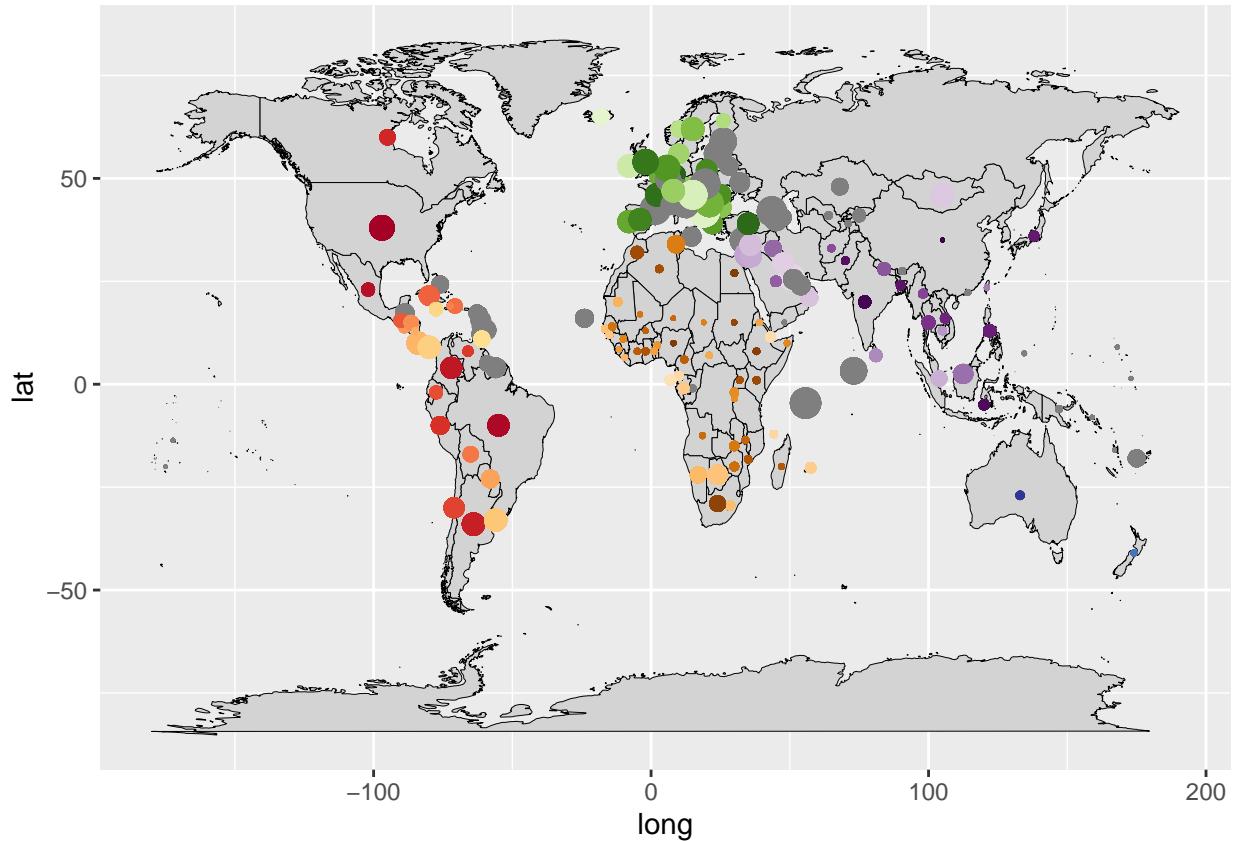
```

        color = "black", fill = "lightgray", size = 0.01
    ) +
geom_point(data = covid, aes(Longitude, Latitude, color=location, size=total_cases_per_million), show
scale_color_manual(values = country_colors) +
scale_size(range = c(0.1,5))

```

map

cases per million



```

map_anim <- ggplot() +
geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "black", fill = "lightgray", size = 0.01
) +
geom_point(data = covid, aes(Longitude, Latitude, color=location, size=total_cases_per_million), show
scale_color_manual(values = country_colors) +
scale_size(range = c(0.1, 5)) +
transition_time(date)

animate(map_anim, duration = 10)

```

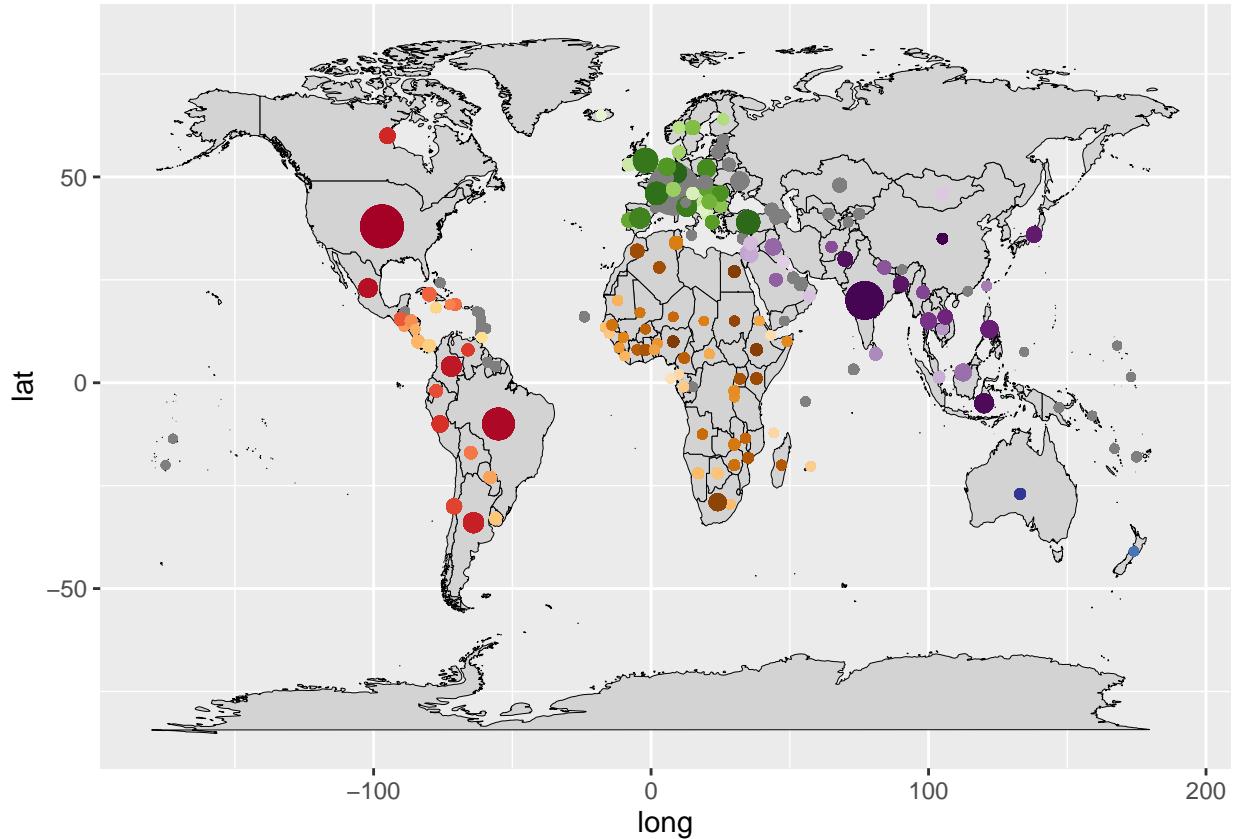
```

map <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "black", fill = "lightgray", size = 0.01
  ) +
  geom_point(data = covid, aes(Longitude, Latitude, color=location, size=total_cases), show.legend = FALSE) +
  scale_color_manual(values = country_colors) +
  scale_size(range = c(1, 15))

map

```

cases total



```

anim_map_total_cases <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "black", fill = "lightgray", size = 0.01
  ) +
  geom_point(data = covid, aes(Longitude, Latitude, color=location, size=total_cases), show.legend = FALSE) +
  scale_color_manual(values = country_colors) +
  scale_size(range = c(1, 15)) +
  transition_reveal(date)

```

```
transition_time(date)

animate(anim_map_total_cases, duration=10)
```