Covid Mirrors Geography

Introduction

We used unsupervised learning to find hidden relationship between countries and the covid pandemic. Since the covid pandemic has affected different counties around the world in different ways, it's difficult to figure out where to begin with comparisons between countries. Such comparisons are motivated by grasping the similarities and differences in countries' situations. Whether it be commonality in health polices like mask or vaccine mandates, or just similar weather, comparing countries could provide insight in combatting the virus. Our analysis offers that first step in finding countries to compare.

Apart from using unsupervised methods to find these hidden relationships, we also wanted to make sure of process and methods would be applicable to other datasets and similar situations. Since this probably will not be the last pandemic we face, we wanted to use methods which can be directly applied to other datasets. Using clustering and PCA allows us this luxury since we do not have to tune a model or perform large amounts of wrangling to use our analysis on related data.

Data

Datasets

Our World in Data COVID Dataset

The primary dataset used is Our World in Data's COVID-19 dataset on github that synthesizes significant variables surrounding the pandemic around the world from reputable sources. Sources of the dataset include Our World in Data themselves, Johns Hopkins University's COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE), European Centre for Disease Prevention and Control (ECDC), official national reports, United Nations, World Bank, Global Burden of Disease, Blavatnik School of Government, and more.

Some examples of significant variables in this dataset include scaled and variations of confirmed cases, comfirmed deaths, excess mortality, hospitalizations and intensive care unit statistics, policy responses, covid tests and test positivity, vaccinations.

This dataset gets updated daily so for many of our analyses and plots, we selected a cut off date. Exploratory data analysis and data visualization were done on dataset up to 12-5-2021 (linked on main github repo).

Links

- (dataset raw csv) https://github.com/owid/covid-19-data/tree/master/public/data
- (github repo) https://github.com/owid/covid-19-data
- (github account) https://github.com/owid

World Bank Open Data

To get an idea of how different global development indicators correlate with COVID developments, we used data from World Bank Open Data to create visualizations.

Link: https://data.worldbank.org/

World Map and Coordinates

To plot world maps, we used latitude and longitude data from Albert Wang's github.

Links

- (dataset raw csv) https://raw.githubusercontent.com/albertyw/avenews/master/old/data/average-latitude-longitude-countries.csv
- (github repo) https://github.com/albertyw/avenews/blob/master/old/data/average-latitude-longitude-countries.csv
- (github) https://github.com/albertyw

gapminder (R library)

We used gapminder's country_colors vector for similar colors for countries from the same continent.

Exploratory Data Analysis

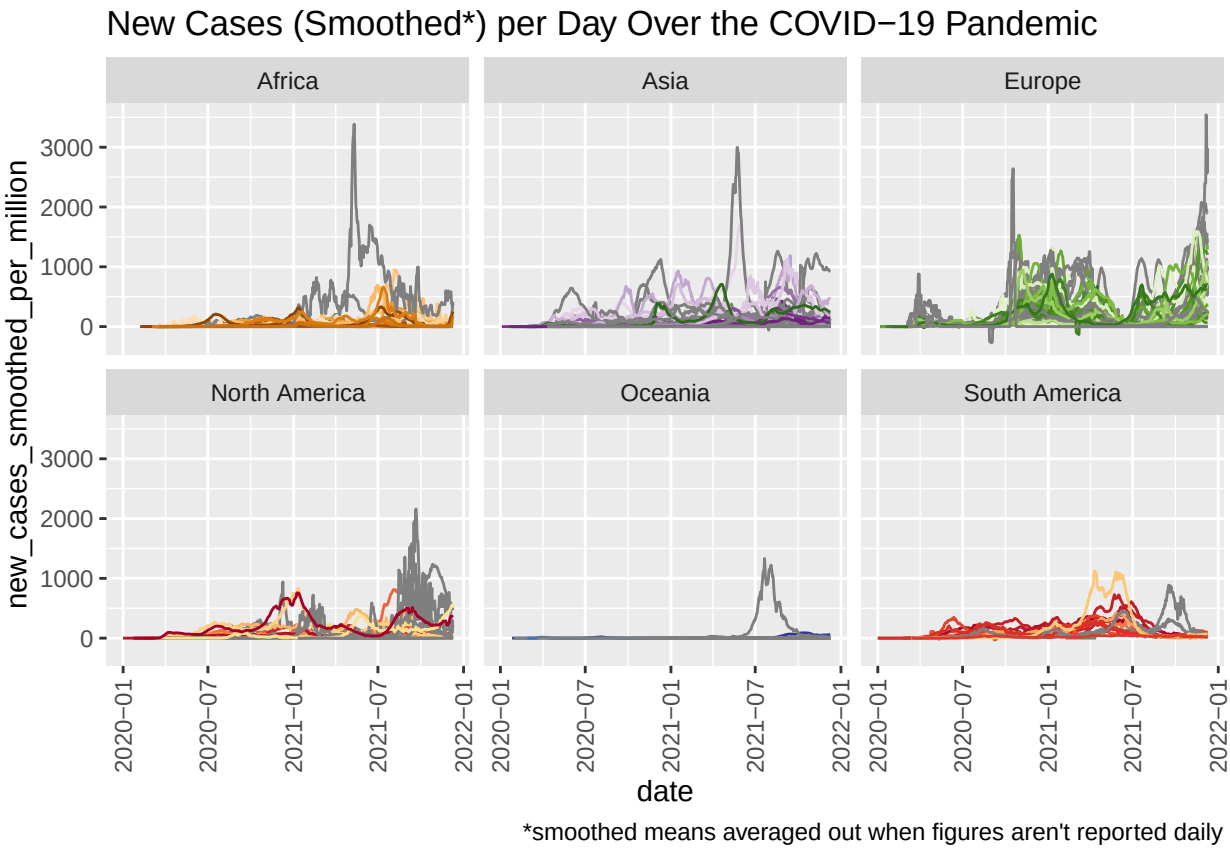
Functions

```
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_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"
            )
}

# replace NA's with 0s
replace_all_na <- function(covid_data) {
  covid_data %>%
    replace(is.na(.), 0)
}
```

```
covid <- read.csv("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-da

# change date to date variable
covid <- covid %>%
  mutate(date = as.Date(date)) %>%
  filter_world %>%
  filter_continents %>%
  filter(date < as.Date("2021-12-10"))


# latitude longitude dataset

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

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

covid <- left_join(covid, lats_long, by = "location")
world <- map_data("world")
```

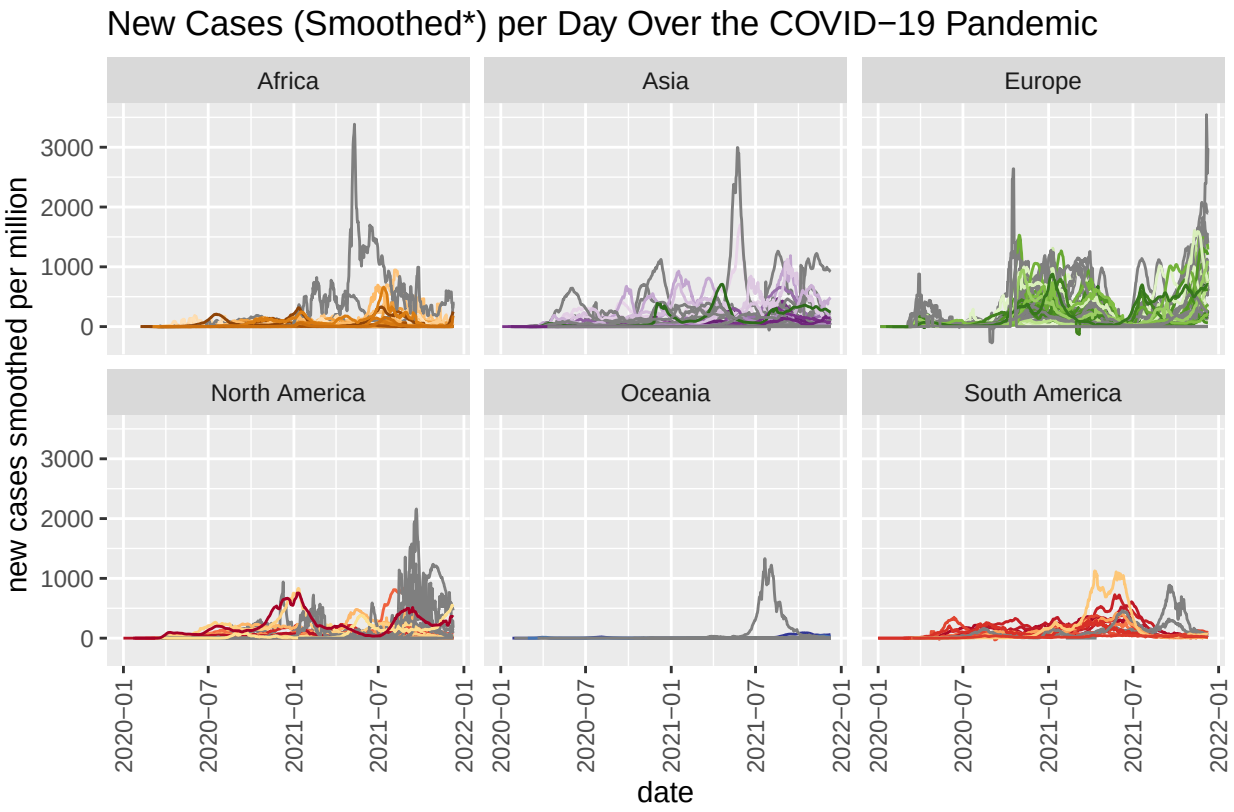
Visualizations

```
covid %>%
  replace_all_na() %>%
  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) +
  labs(title = "New Cases (Smoothed*) per Day Over the COVID-19 Pandemic",
       caption = "*smoothed means averaged out when figures aren't reported daily") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

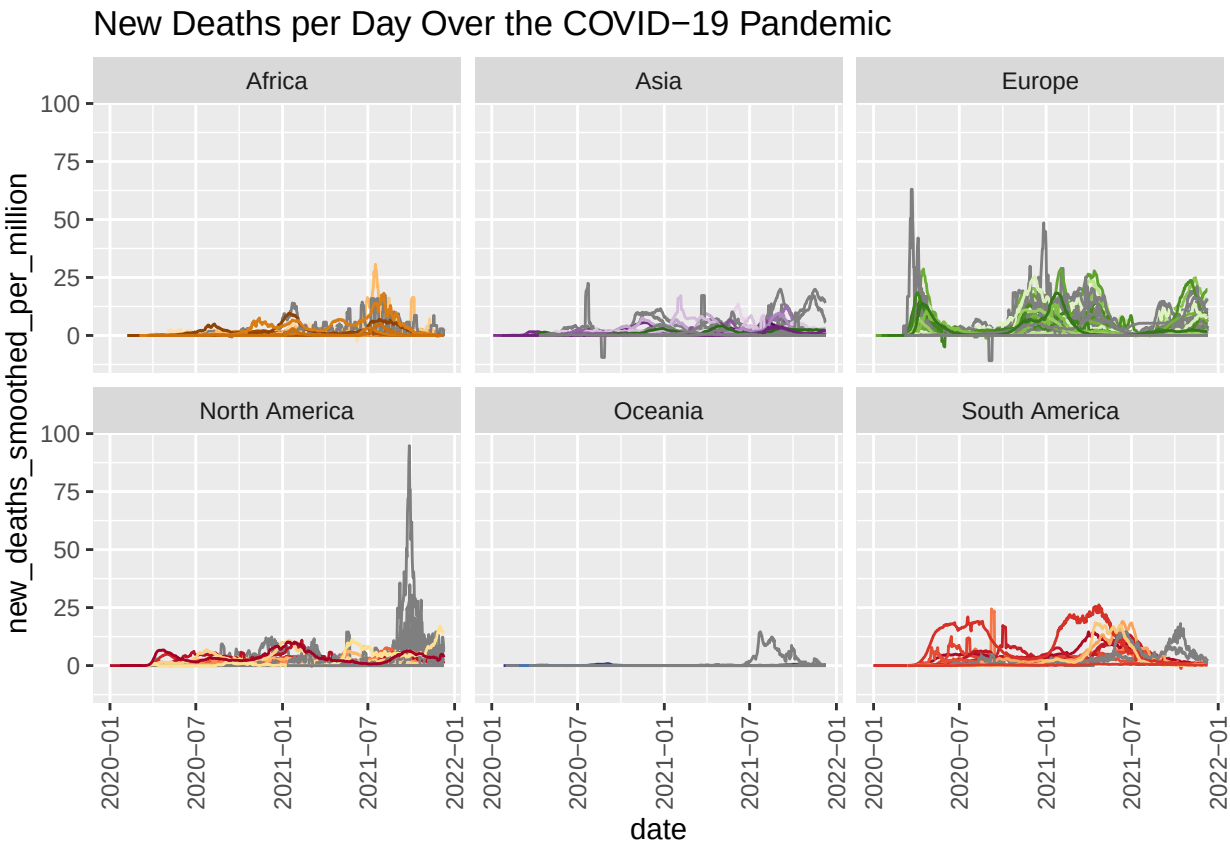


As we can see, it appears that Oceania experienced the lowest rate of new cases in comparison with the other regions. We can see that Europe seemed to experience the largest spike in new cases per million as well.

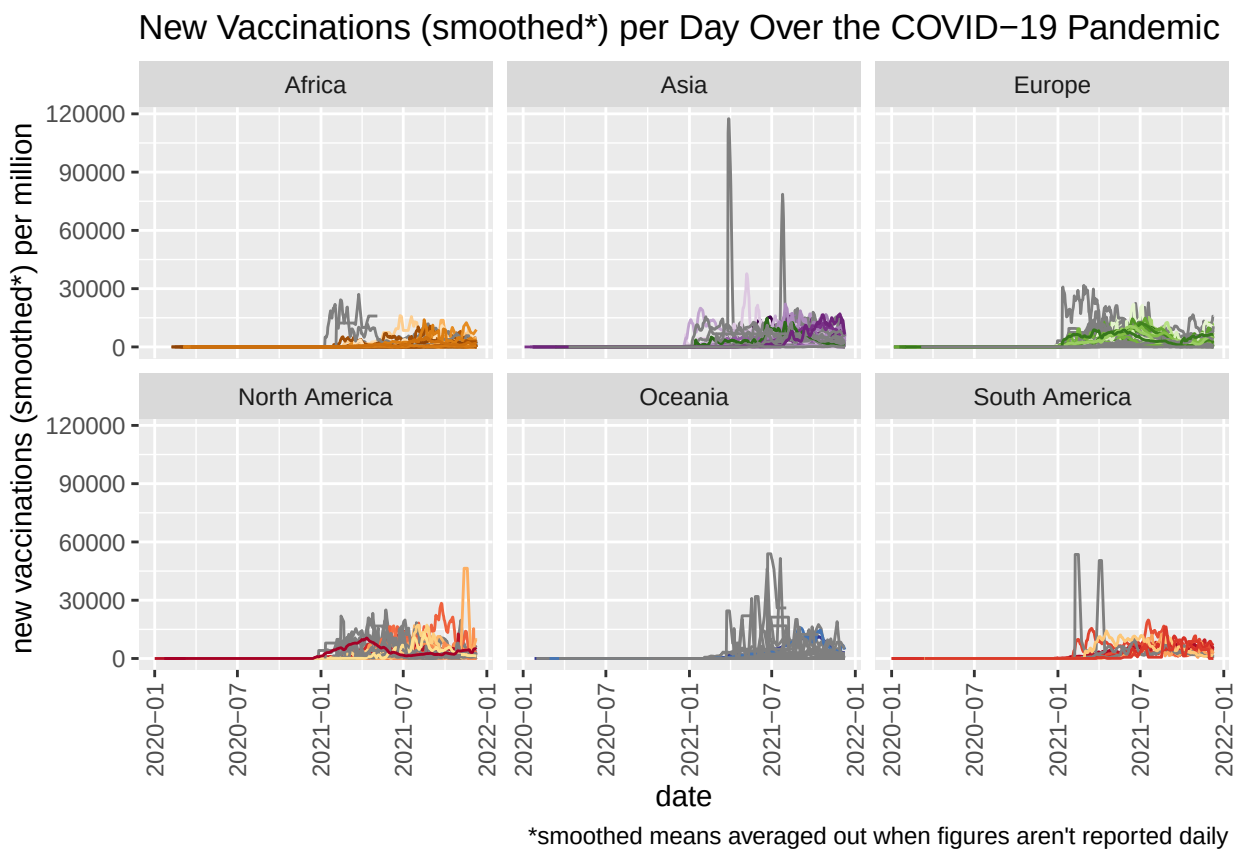
```
covid %>%
  replace_all_na() %>%
  ggplot(aes(x = date, y = new_cases_smoothed_per_million, color=location)) +
  geom_line(aes(color = location), show.legend = FALSE) +
  facet_wrap(~continent) +
  scale_colour_manual(values = country_colors) +
  labs(title = "New Cases (Smoothed*) per Day Over the COVID-19 Pandemic",
       caption = "*smoothed means averaged out when figures aren't reported daily",
       y = "new cases smoothed per million") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
covid %>%
  replace_all_na() %>%
  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) +
  labs(title = "New Deaths per Day Over the COVID-19 Pandemic") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



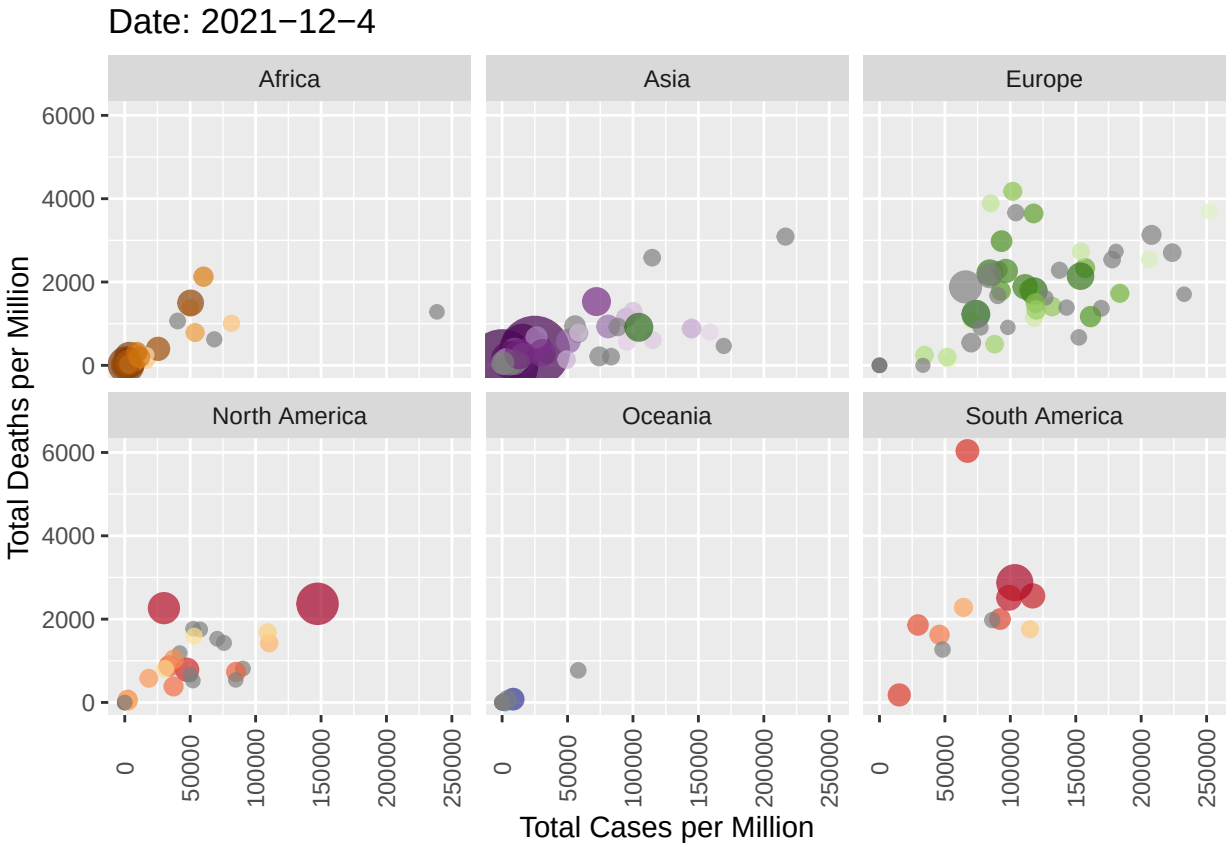

```
covid %>%
  replace_all_na %>%
  ggplot(aes(x = date, y = new_vaccinations_smoothed_per_million, color=location)) +
  geom_line(aes(color = location), show.legend = FALSE) +
  facet_wrap(~continent) +
  scale_colour_manual(values = country_colors) +
  labs(title = "New Vaccinations (smoothed*) per Day Over the COVID-19 Pandemic",
       caption = "*smoothed means averaged out when figures aren't reported daily",
       y = "new vaccinations (smoothed*) per million") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Although we have a good grasp of the developments of the coronavirus pandemic due to the abundance of new reports, we wanted to getter a better idea of the data itself. These exploratory plots, facetted by continent, give us a quick snapshot of how some important variables such as new cases, deaths and vaccinations differ across continents over time.

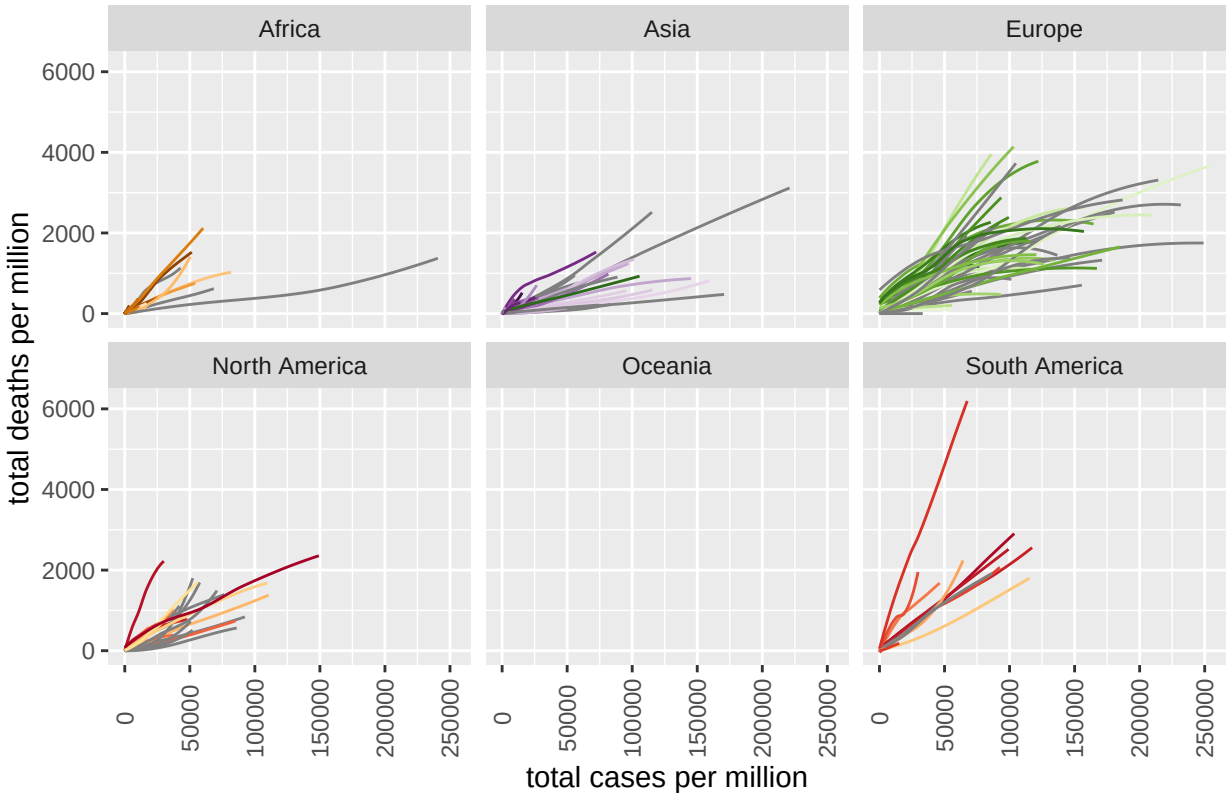
```
case_death_anim <- covid %>%
  replace_all_na() %>%
  filter_continents() %>%
  ggplot(aes(total_cases_per_million, total_deaths_per_million, color = location, size=population)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_color_manual(values = country_colors) +
  scale_size(range =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

animate(case_death_anim, duration = 10)
```

In order to retain the time component while investigating relationships between other variables such as the total cases vs total deaths we decided to create animations, progressing over time. This allows us to see the rate of increase, where for some countries have faster rates of deaths and new cases than others.



We also wanted to be able to express the relationship between cases and deaths over time without be reliant on an animation, so this static plot provide the same information. We lose out on visualising the rate of increase for both variables over time, but we are still able to communicate that more cases leads to more deaths.

```
covid %>%
  replace_all_na() %>%
  filter_continents() %>%
  ggplot(aes(total_cases_per_million, total_deaths_per_million, color = location, size = population)) +
  geom_smooth(size=0.5, se=FALSE, show.legend=FALSE) +
  facet_wrap(~continent) +
  scale_color_manual(values = country_colors) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(title = "",
       y = "total deaths per million", x = "total cases per million")
```

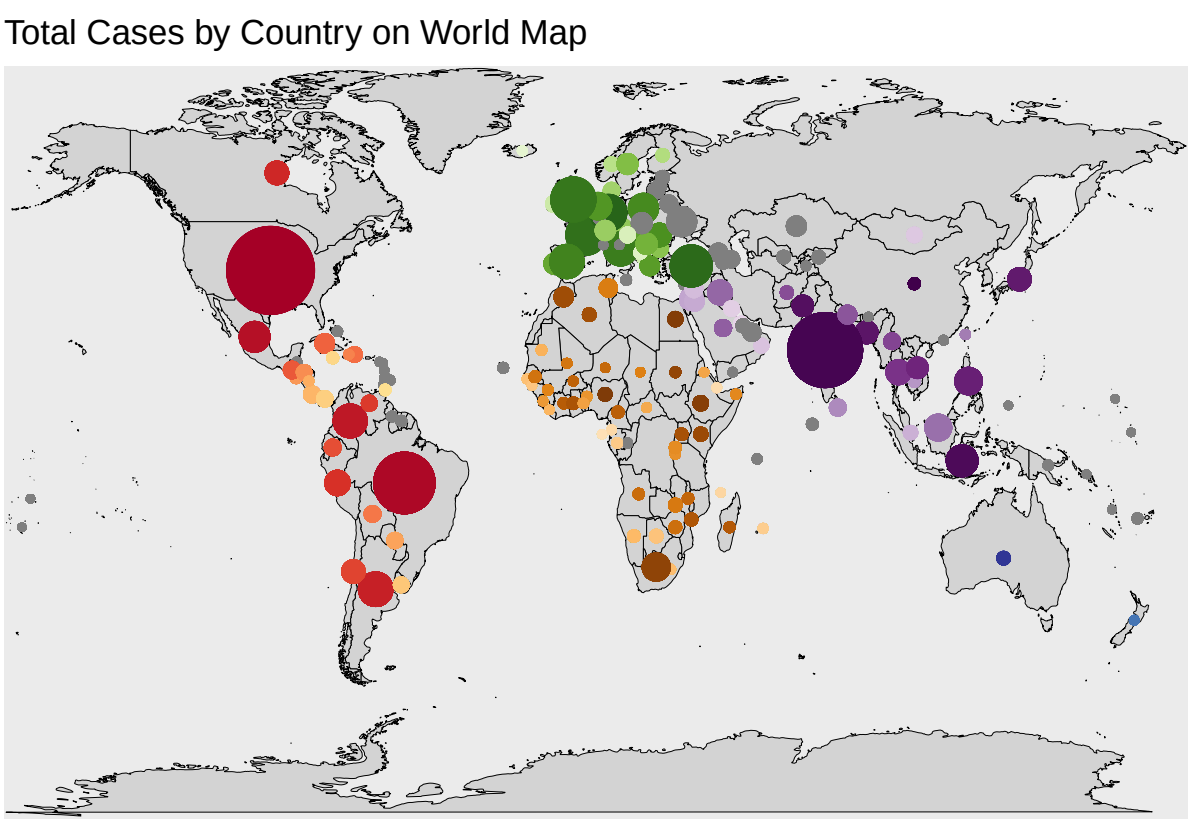


Since we facetted by continent, we are unable to fully distinguish each countries rates of change for new cases. So we utilise a world map where we can fully visualise every countries relationship with the coronavirus, in both animation and static form.

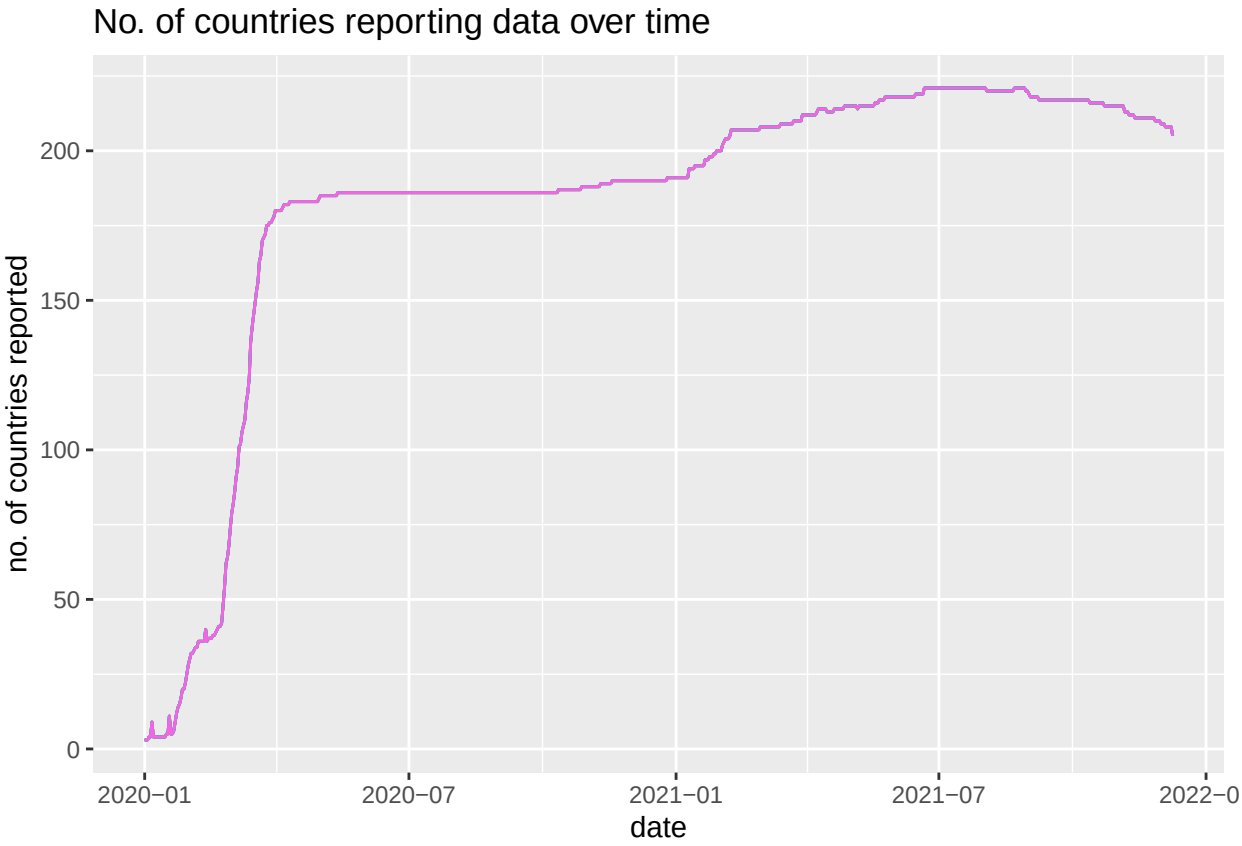
```
anim_map_total_cases <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "black", fill = "lightgray", size = 0.01
  ) +
  geom_point(data = covid, aes(Longitude, Latitude, color=location, size=total_cases), show.legend = FA
  scale_color_manual(values = country_colors) +
  scale_size(range = c(1, 15)) +
  transition_time(date) +
  labs(x = "", y = "") +
  scale_x_discrete(labels=NULL, breaks=NULL) +
  scale_y_discrete(labels=NULL, breaks=NULL)

animate(anim_map_total_cases, 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 = FA
```

```
  scale_color_manual(values = country_colors) +
  scale_size(range = c(1, 15)) +
  labs(title = "Total Cases by Country on World Map", x = "", y = "") +
  scale_x_discrete(labels=NULL, breaks=NULL) +
  scale_y_discrete(labels=NULL, breaks=NULL)

map
```

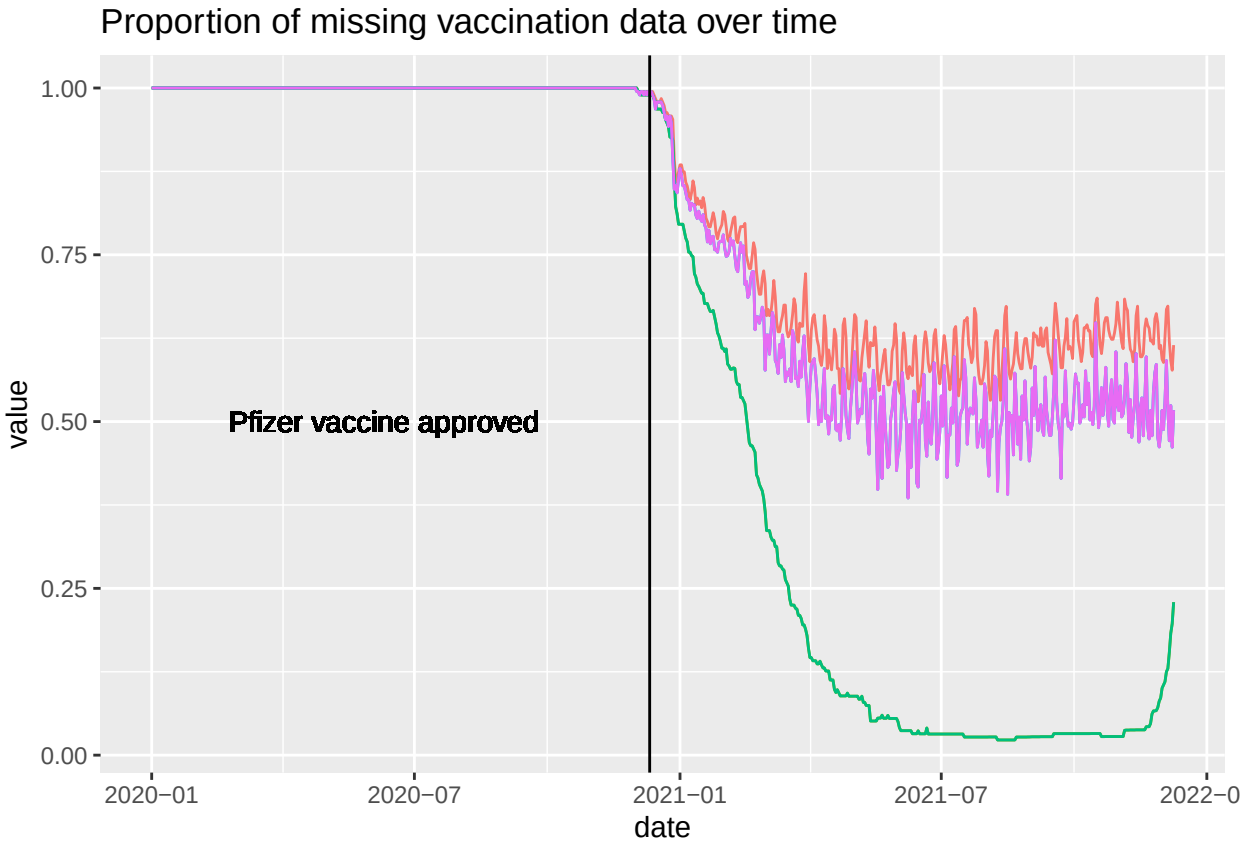



```
vars <- names(covid)

vars <- vars[! vars %in% c('date')]

len_per_day <- covid %>% group_by(date) %>%
  summarise(across(vars, length))

lengthsummary <- len_per_day %>%
  pivot_longer(!date, names_to="variable") %>%
  mutate(date=ymd(date))

lengthsummary %>% filter(grepl('cases', variable)) %>% ggplot() + geom_line(aes(x=date, y=value, color=
```



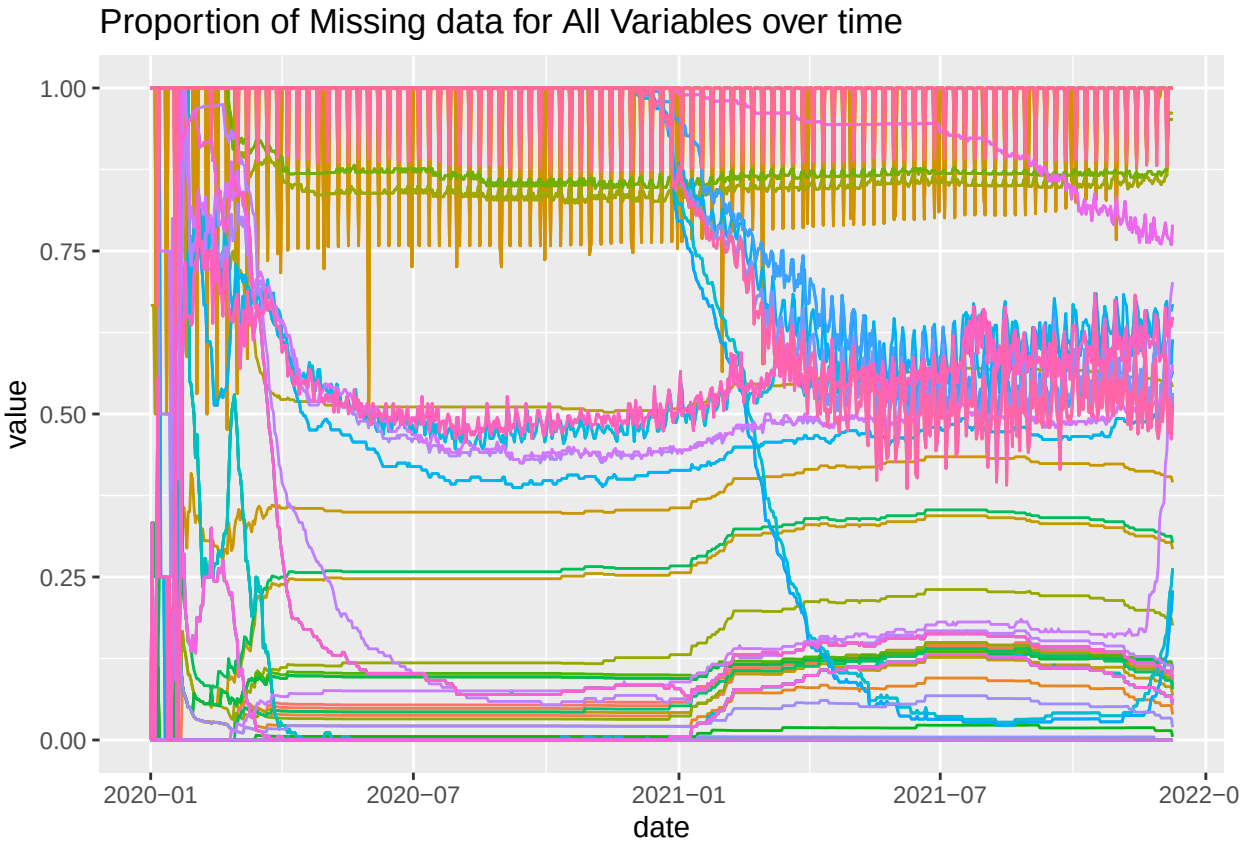
As well as performing EDA on data which is present, we also wanted to better understand the data that is missing. Not every country in the world will be reporting covid data at the same time due to different situations as well as the nature of the pandemic. So we visualise the number of countries that are reporting data over time.

```
prop <- covid %>% group_by(date) %>%
  summarise(across(vars, funs(sum(is.na(.)) / length(.))))

NAsummary <- prop %>%
  pivot_longer(!date, names_to="variable") %>%
  mutate(date=ymd(date))

NAsummary %>% filter(grepl('vaccination', variable)) %>% ggplot() + geom_line(aes(x=date, y=value, colo
```



We can see that there is missing data on vaccinations from prior to the first vaccine approval which makes sense. We just need to be aware of analysis which we do pre and post vaccinations because of the missing data.

```
NAsummary %>% ggplot(aes(x=date, y=value, color=variable)) +
  geom_line() +
  theme(legend.position = "none") +
  labs(title = "Proportion of Missing data for All Variables over time")
```



Since the visualisation is very difficult to interpret due to the sheer number of variables, we pivot to a different visualisation technique which offers flexibility and variable choice. The shiny app!

```
bad_cols <- c("excess_mortality_cumulative", "excess_mortality", "excess_mortality_cumulative_per_millio
```

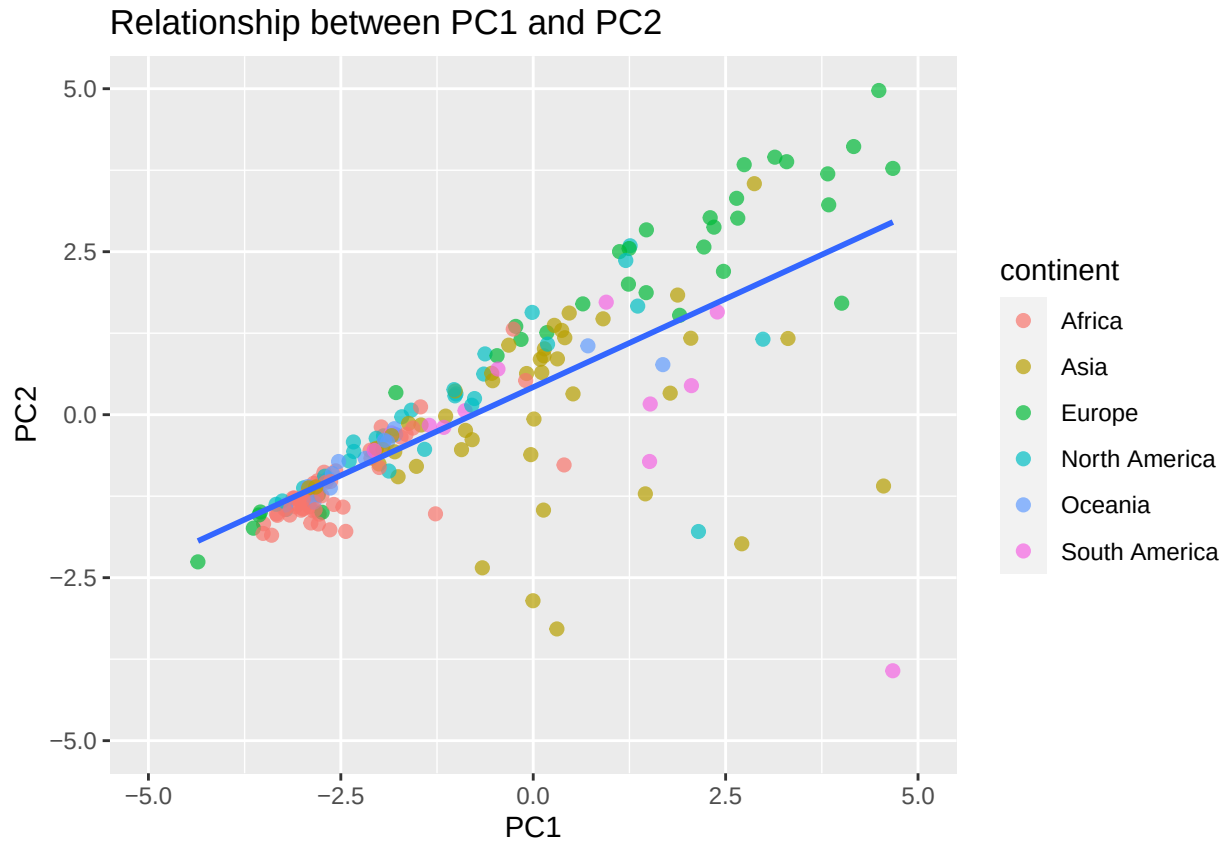
```
test_cleaning <- covid %>%
  select(!bad_cols) %>%
  #select(!is.character) %>%
  replace_all_na() %>%
  filter(date > as.Date("2020-12-28")) %>%
  group_by(date) %>%
  summarize_all(sd) %>%
  filter_all(all_vars(. < 0.1))

test_cleaning
```

```
## # A tibble: 0 x 58
## # ... with 58 variables: date <date>, iso_code <dbl>, continent <dbl>,
## #   location <dbl>, total_cases <dbl>, new_cases <dbl>,
## #   new_cases_smoothed <dbl>, total_deaths <dbl>, new_deaths <dbl>,
## #   new_deaths_smoothed <dbl>, total_cases_per_million <dbl>,
## #   new_cases_per_million <dbl>, new_cases_smoothed_per_million <dbl>,
## #   total_deaths_per_million <dbl>, new_deaths_per_million <dbl>,
## #   new_deaths_smoothed_per_million <dbl>, icu_patients <dbl>, ...
```

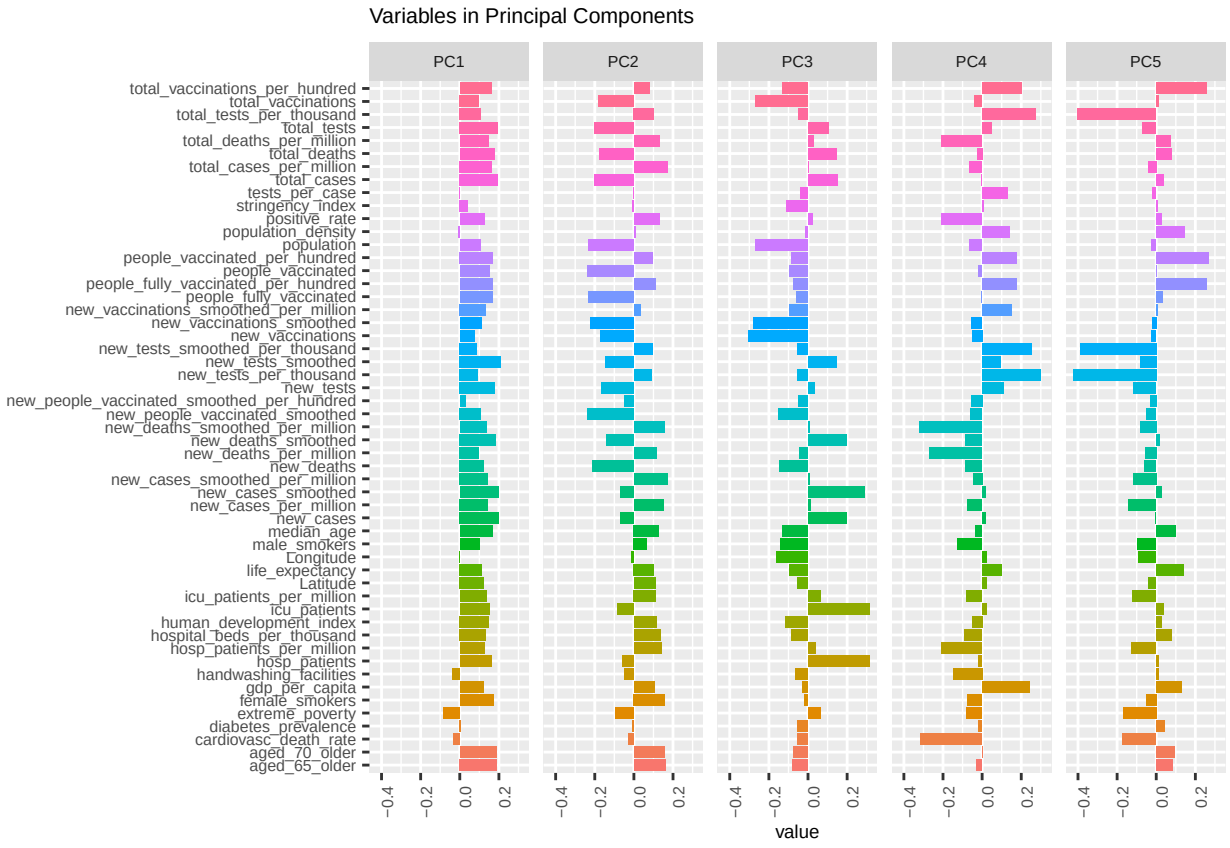
Now we can start to visualise the principle components analysis. Using only 1 days worth of data as a proof of concept, we try to run and visualise how each of the principle components react to each other, as well as

what variables are most important in each of the principle components. We can see that most of variables are utilised in the first principle component, where new cases and testing are the most important contributions to the principle component.

```
one_day <- covid %>% filter(date == "2021-12-05")

one_day_prepped <- replace_all_na(one_day) %>%
  filter_continents()

ready <- one_day_prepped %>% select(!bad_cols)

pca_recipe <- recipe(~., data = ready) %>%
  step_center(all_numeric()) %>%
  step_scale(all_numeric()) %>%
  step_pca(all_numeric(), id = "pca")

pca_estimates <- prep(pca_recipe)

juice(pca_estimates) %>%
  ggplot(aes(PC1, PC2)) +
  geom_point(aes(color = continent), alpha = 0.7, size = 2)+
  labs(title="Relationship between PC1 and PC2") +
  xlim(-5, 5) + ylim(-5, 5) +
  geom_smooth(method = "lm", se = FALSE)
```

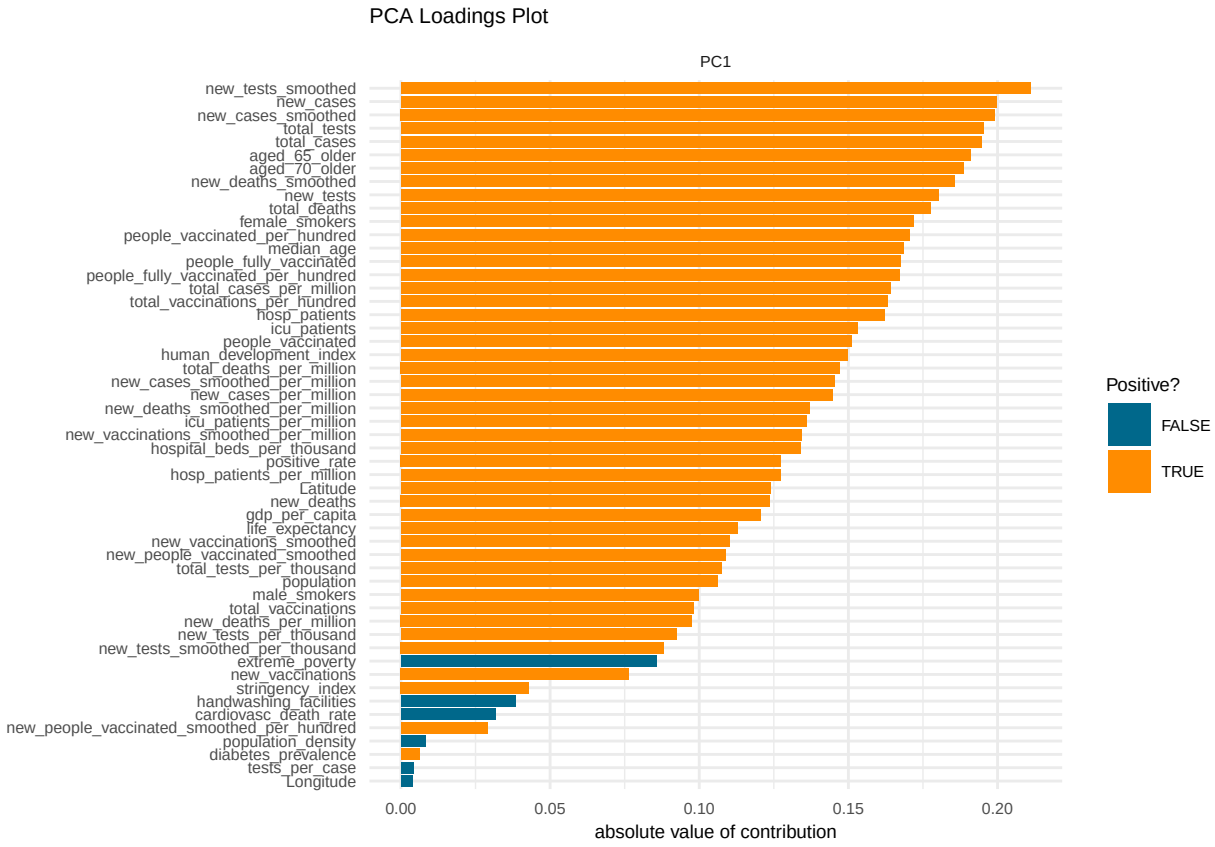


```
tidied_pca <- tidy(pca_estimates, 2)
tidy_pca_loadings <- pca_estimates%>%
  tidy(id = "pca")

tidy_pca_loadings %>%
  filter(component %in% paste0("PC", 1:5)) %>%
  mutate(component = fct_inorder(component)) %>%
  ggplot(aes(value, terms, fill = terms)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~component, nrow = 1) +
  labs(y = NULL) +
  theme(text = element_text(size = 7), axis.text.x = element_text(angle=90, hjust=1)) +
  labs(title = "Variables in Principal Components")
```




```
plot_loadings <- tidy_pca_loadings %>%
  filter(component %in% c("PC1")) %>%
  mutate(terms = tidytext::reorder_within(terms,
                                          abs(value),
                                          component)) %>%
  ggplot(aes(abs(value), terms, fill = value>0)) +
  geom_col() +
  facet_wrap( ~component, scales = "free_y") +
  scale_y_reordered() + # appends ___ and then the facet at the end of each string
  scale_fill_manual(values = c("deepskyblue4", "darkorange")) +
  labs( x = "absolute value of contribution",
        y = NULL,
```

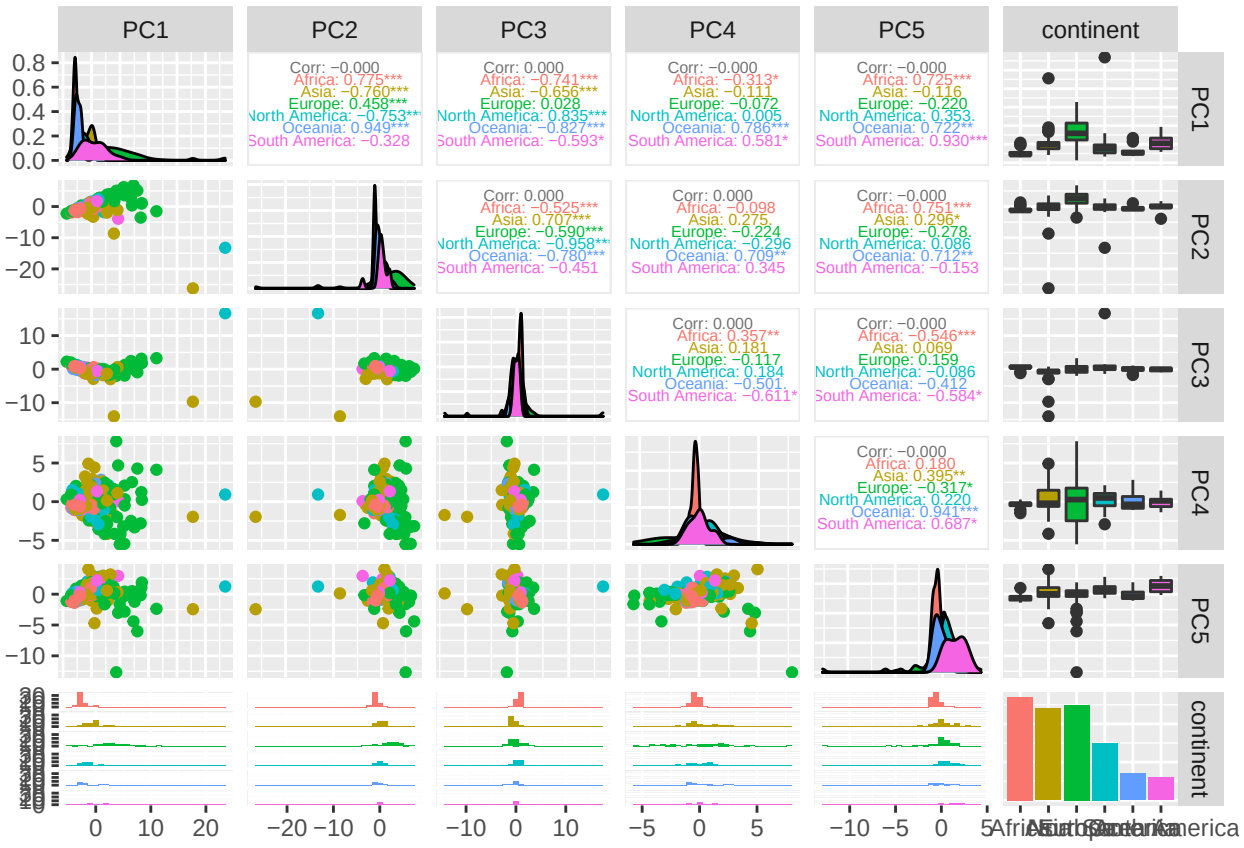
```
        fill = "Positive?",
        title = "PCA Loadings Plot") +
  theme_minimal() +
  theme(text = element_text(size = 7))

plot_loadings
```



Pretty neat! Now, we'll try to perform clustering on the PCA components

```
juice_df <- juice(pca_estimates)
```

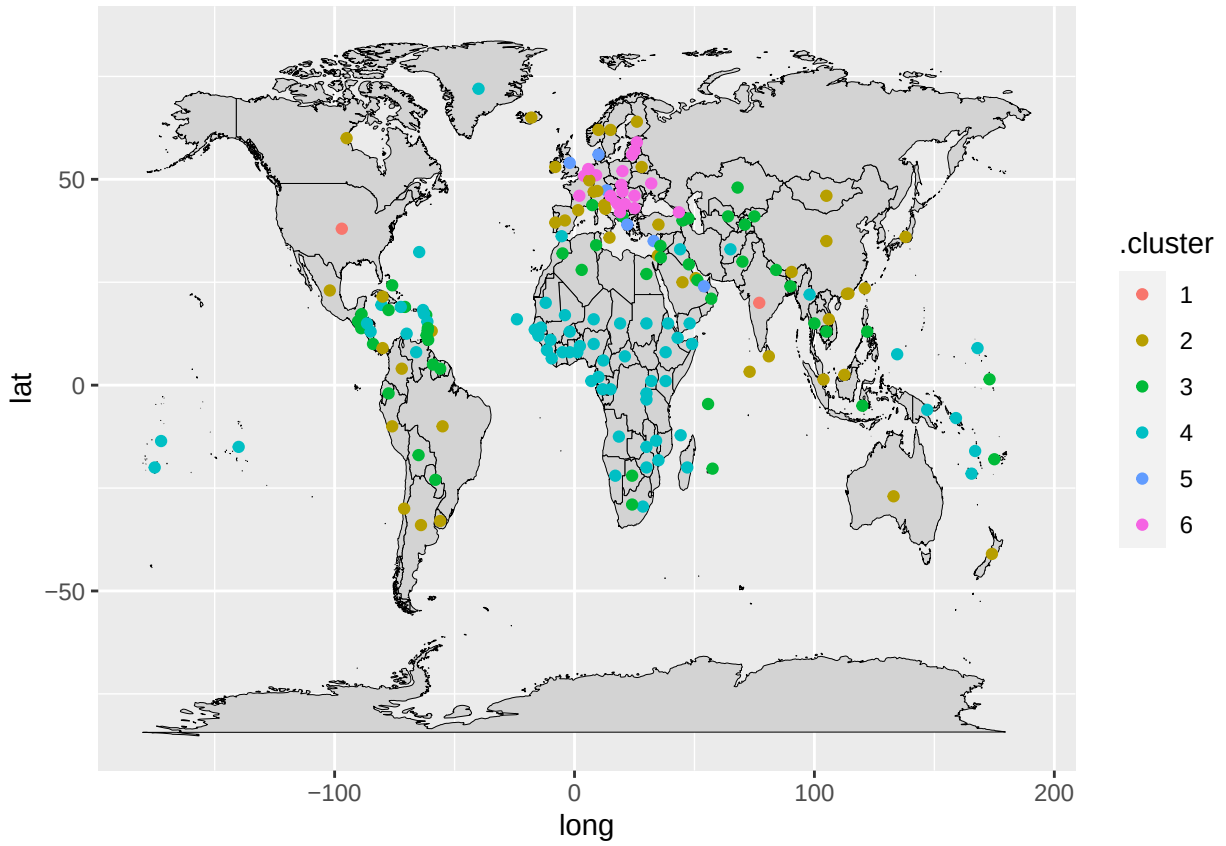
```
juice_df %>%
  select(starts_with("PC"), continent) %>%
  ggpairs(aes(color = continent), colour = "cyl",
    upper = list(continuous = wrap("cor", size = 2)))
```



As we can see, there is no immediately obvious groupings between the continents based on the principle components.

```
set.seed(47)
pca_kclust <- juice_df %>%
  select(starts_with("PC")) %>%
  kmeans(centers=6)
```

```
loc_clusters <- pca_kclust %>% augment(juice_df)
long_lat_clusters <- left_join(loc_clusters, lats_long, by="location")

cluster_map <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "black", fill = "lightgray", size = 0.01
  ) +
  geom_point(data = long_lat_clusters, aes(x=Longitude, y=Latitude, color=.cluster))
cluster_map
```



woah~ Covid mirrors geography? :o 8D Here's an idea, what if we did this for every day in the dataset >;)

Since covid situations change for every country every day, performing our PCA clustering analysis per day and animating could show how the clusters change over time. Where some countries who are experiencing a rise in cases at the same time may be clustered together.

PCA-Clustering by Day

We want to include vaccination data, and vaccinations did not start for all the countries included until December 16th, 2020, so we will only look at days since then.

```
clean_covid <- covid %>%
  replace_all_na() %>%
  select(!bad_cols)

cleanest_covid <- clean_covid %>%
  filter(date > as.Date("2020-12-28"))
```

```
sorted_dates <- sort(unique(cleanest_covid$date))
length(sorted_dates)
```

```
## [1] 346
```

We will be looking at 346 days.

```r
first_day_df <- clean_covid %>% filter(date == as.Date("2020-12-28"))
first_day_df %>%
  group_by(date)  %>%
  summarise_all(sd)
```

```
## # A tibble: 1 x 58
##   date       iso_code continent location total_cases new_cases new_cases_smooth~
##   <date>        <dbl>     <dbl>    <dbl>       <dbl>     <dbl>             <dbl>
## 1 2020-12-28       NA        NA       NA    1720345.    12872.            14094.
## # ... with 51 more variables: total_deaths <dbl>, new_deaths <dbl>,
## #   new_deaths_smoothed <dbl>, total_cases_per_million <dbl>,
## #   new_cases_per_million <dbl>, new_cases_smoothed_per_million <dbl>,
## #   total_deaths_per_million <dbl>, new_deaths_per_million <dbl>,
## #   new_deaths_smoothed_per_million <dbl>, icu_patients <dbl>,
## #   icu_patients_per_million <dbl>, hosp_patients <dbl>,
## #   hosp_patients_per_million <dbl>, new_tests <dbl>, total_tests <dbl>, ...
```

```r
# function to perform PCA and cluster based and colors!
get_cluster_colored_df <- function(day_df) {
  pca_recipe <- recipe(~., data=day_df) %>%
    step_center(all_numeric()) %>%
    step_scale(all_numeric()) %>%
    step_pca(all_numeric(), id="pca")

  pca_estimates <- prep(pca_recipe)
  juice_df <- juice(pca_estimates)

  pca_kclust <- juice_df %>%
    select(starts_with("PC")) %>%
    kmeans(centers=6)

  loc_clusters <- pca_kclust %>%
    augment(juice_df)

  us_row <- loc_clusters %>%
    dplyr::filter(location == "United States")
  us_clust = us_row$.cluster

  car_row <- loc_clusters %>%
    dplyr::filter(location == "Central African Republic")
  car_clust = car_row$.cluster

  ger_row <- loc_clusters %>%
    dplyr::filter(location == "Germany")
  ger_clust = ger_row$.cluster

  ;costa_row <- loc_clusters %>%
    dplyr::filter(location == "Costa Rica")
  costa_clust = costa_row$.cluster

  remaining_clusts <- setdiff(as.factor(seq(1,6)), c(us_clust, car_clust, ger_clust, costa_clust))

  loc_clusters <- loc_clusters %>%
```

```
    mutate(my_color = ifelse(.cluster == us_clust, "red", "orange1")) %>%
    mutate(my_color = ifelse(.cluster == costa_clust, "springgreen3", my_color)) %>%
    mutate(my_color = ifelse(.cluster == remaining_clusts[1], "purple3", my_color)) %>%
        mutate(my_color =  ifelse(.cluster == ger_clust, "orchid2", my_color)) %>%
        mutate(my_color = ifelse(.cluster == car_clust, "steelblue2", my_color))

  return(loc_clusters)
}
```

```
# the first date
all_loc_clusters <- get_cluster_colored_df(first_day_df)

# the rest of the dates
for (day in sorted_dates){
  day_data <- cleanest_covid %>%
    filter(date == day)

  loc_clusters <- get_cluster_colored_df(day_data)
  all_loc_clusters <- rbind(all_loc_clusters, loc_clusters)
}
```

```
all_long_lat_clusters <- left_join(all_loc_clusters, lats_long, by="location")
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 = all_long_lat_clusters, aes(x=Longitude, y=Latitude, color=my_color)) +
  transition_time(date) +
  scale_x_discrete(labels=NULL, breaks=NULL) +
  scale_y_discrete(labels=NULL, breaks=NULL) +
  scale_color_manual(labels=c("1", "2", "3", "4", "5", "6"), values=c("orange1", "orchid2", "purple3",
  labs(title = 'Date: {frame_time}', x="", y="", color="cluster")

animate(map_anim, duration = 20)
```

```
head(all_loc_clusters)
```

```
## # A tibble: 6 x 12
##   iso_code continent  location date       ISO.3166.Countr~    PC1    PC2     PC3
##   <fct>    <fct>      <fct>    <date>     <fct>             <dbl>  <dbl>   <dbl>
## 1 AFG      Asia       Afghani~ 2020-12-28 AF               -2.01   2.59   0.165
## 2 ALB      Europe     Albania  2020-12-28 AL               -0.113 -1.60  -0.818
## 3 DZA      Africa     Algeria  2020-12-28 DZ               -1.22   0.910 -0.283
## 4 AND      Europe     Andorra  2020-12-28 AD                0.297 -2.00   0.310
## 5 AGO      Africa     Angola   2020-12-28 AO               -1.94   2.53   0.320
## 6 ATG      North Ame~ Antigua~ 2020-12-28 AG               -1.26   0.922  0.0802
## # ... with 4 more variables: PC4 <dbl>, PC5 <dbl>, .cluster <fct>,
## #   my_color <chr>
```

```r
cleaner_covid_2 <- covid %>%
  replace_all_na() %>%
  select(contains("per"), c(continent, location, date, aged_65_older, aged_70_older, gdp_per_capita, ex
  select(!c(excess_mortality_cumulative_per_million, total_boosters_per_hundred, weekly_icu_admissions_
cleanest_covid_2 <- cleaner_covid_2 %>%
  dplyr::filter(date > as.Date("2020-12-28")) %>%
  dplyr::filter(date < as.Date("2021-12-10"))
```

```r
get_cluster_colored_df2 <- function(day_df) {
  pca_recipe <- recipe(~., data=day_df) %>%
    step_center(all_numeric()) %>%
    step_scale(all_numeric()) %>%
    step_pca(all_numeric(), id="pca")

  pca_estimates <- prep(pca_recipe)
  juice_df <- juice(pca_estimates)

  pca_kclust <- juice_df %>%
    select(starts_with("PC")) %>%
    kmeans(centers=6)

  loc_clusters <- pca_kclust %>%
    augment(juice_df)

  tunisia_row <- loc_clusters %>%
    dplyr::filter(location == "Tunisia")
  tunisia_clust = tunisia_row$.cluster

  car_row <- loc_clusters %>%
    dplyr::filter(location == "Central African Republic")
  car_clust = car_row$.cluster

  ger_row <- loc_clusters %>%
    dplyr::filter(location == "Germany")
  ger_clust = ger_row$.cluster

  uae_row <- loc_clusters %>%
    dplyr::filter(location == "United Arab Emirates")
  uae_clust = uae_row$.cluster

  remaining_clusts <- setdiff(as.factor(seq(1,6)), c(tunisia_clust, car_clust, ger_clust, uae_clust))

  loc_clusters <- loc_clusters %>%
    mutate(my_color = ifelse(.cluster == remaining_clusts[1], "red", "orange1")) %>%
    mutate(my_color = ifelse(.cluster == uae_clust, "purple3", my_color)) %>%
        mutate(my_color = ifelse(.cluster == tunisia_clust, "springgreen3", my_color)) %>%
        mutate(my_color =  ifelse(.cluster == ger_clust, "orchid2", my_color)) %>%
        mutate(my_color = ifelse(.cluster == car_clust, "steelblue2", my_color))

  return(loc_clusters)
}
```

```
first_day_df2 <- cleaner_covid_2 %>% dplyr::filter(date == as.Date("2020-12-28"))
all_loc_clusters_colored2 <- get_cluster_colored_df2(first_day_df2)

# the rest of the dates
for (day in sorted_dates){
  day_data <- cleaner_covid_2 %>%
    dplyr::filter(date == day)

  loc_clusters <- get_cluster_colored_df2(day_data)
  all_loc_clusters_colored2 <- rbind(all_loc_clusters_colored2, loc_clusters)
}

all_long_lat_clusters_colored2 <- left_join(all_loc_clusters_colored2, lats_long, by="location")
head(all_long_lat_clusters_colored2)
```

```
## # A tibble: 6 x 13
##   continent  location  date          PC1     PC2    PC3     PC4     PC5 .cluster
##   <fct>      <chr>     <date>      <dbl>   <dbl>  <dbl>   <dbl>   <dbl> <fct>
## 1 Asia       Afghanis~ 2020-12-28 -2.98   0.269   0.458  0.340   0.653  5
## 2 Europe     Albania   2020-12-28  0.761 -0.788  -0.668  0.985   0.268  6
## 3 Africa     Algeria   2020-12-28 -1.63  -0.125  -0.918  0.645   0.683  3
## 4 Europe     Andorra   2020-12-28  2.60  -0.558   2.75  -0.0326  2.93   6
## 5 Africa     Angola    2020-12-28 -2.75   0.416   0.720 -0.602  -0.0812 5
## 6 North Ame~ Antigua ~ 2020-12-28 -1.17   0.0241 -1.29  -0.363   0.987  3
## # ... with 4 more variables: my_color <chr>, ISO.3166.Country.Code <chr>,
## #   Latitude <dbl>, Longitude <dbl>
```

```
map_anim_colored2 <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "black", fill = "lightgray", size = 0.01
  ) +
  geom_point(data = all_long_lat_clusters_colored2, aes(x=Longitude, y=Latitude, color=my_color)) +
  transition_time(date) +
  scale_x_discrete(labels=NULL, breaks=NULL) +
  scale_y_discrete(labels=NULL, breaks=NULL) +
  scale_color_manual(labels=c("1", "2", "3", "4", "5", "6"), values=c("orange1", "orchid2", "purple3",
  labs(title = 'Date: {frame_time}', x="", y="", color="cluster")

animate(map_anim_colored2, duration = 20)
```

HDI-covid plots

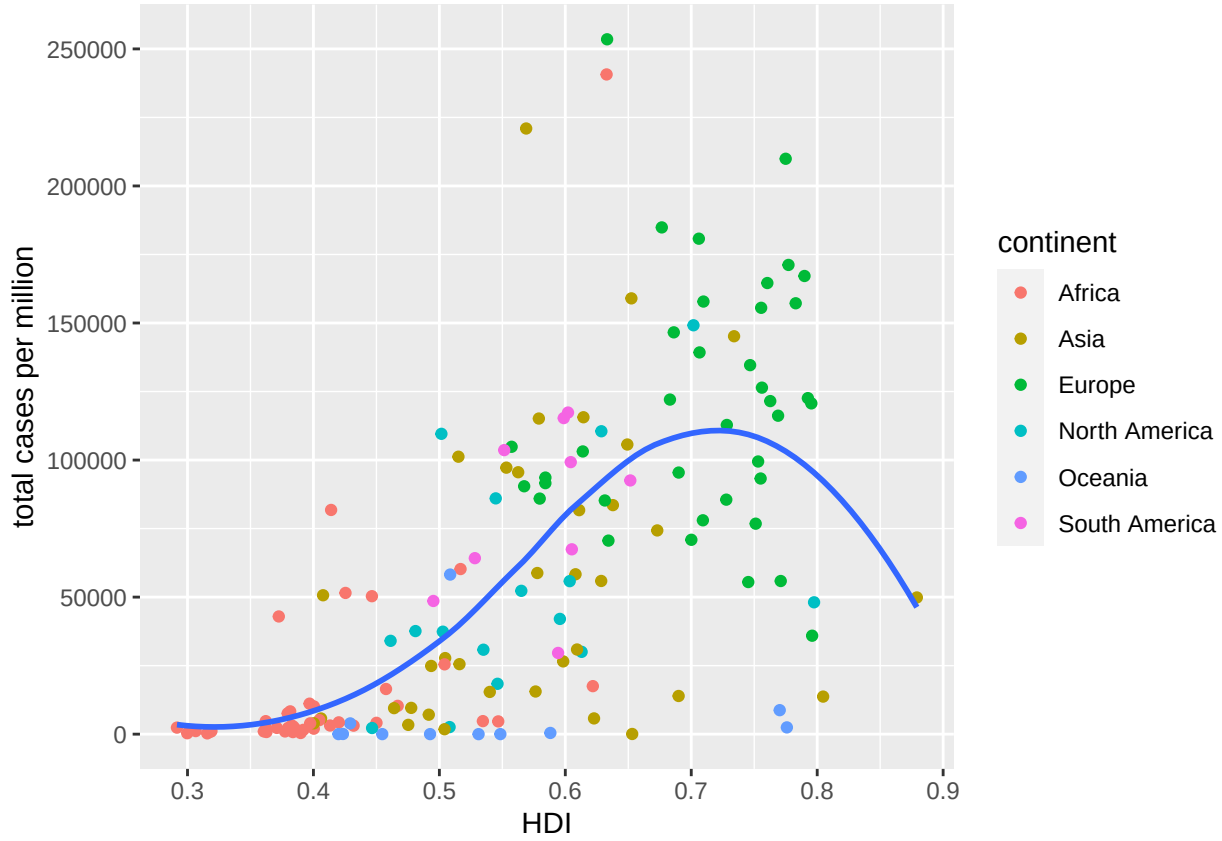
In order to kickstart our analysis of external factors affecting different countries' covid situations, we utilise world bank data.

```
covid_latest <- covid %>%
  filter(date == as.Date("2021-12-9"))

# data
HDI_2020 <- read.csv("https://raw.githubusercontent.com/ST47S-CompStats-Fall2021/GroupJ-COVID/main/data
```

```
HDI_2020 <- HDI_2020 %>%
  rename(location = Country.Name) %>%
  rename(HDI = X2020)
covid_latest_hdi <- left_join(covid_latest, HDI_2020, by = "location")
covid_latest_hdi <- covid_latest_hdi %>%
  filter(!is.na(HDI)) %>%
  arrange(desc(HDI))
covid_hdi <- left_join(covid, HDI_2020, by = "location") %>%
  filter(!is.na(HDI))
```

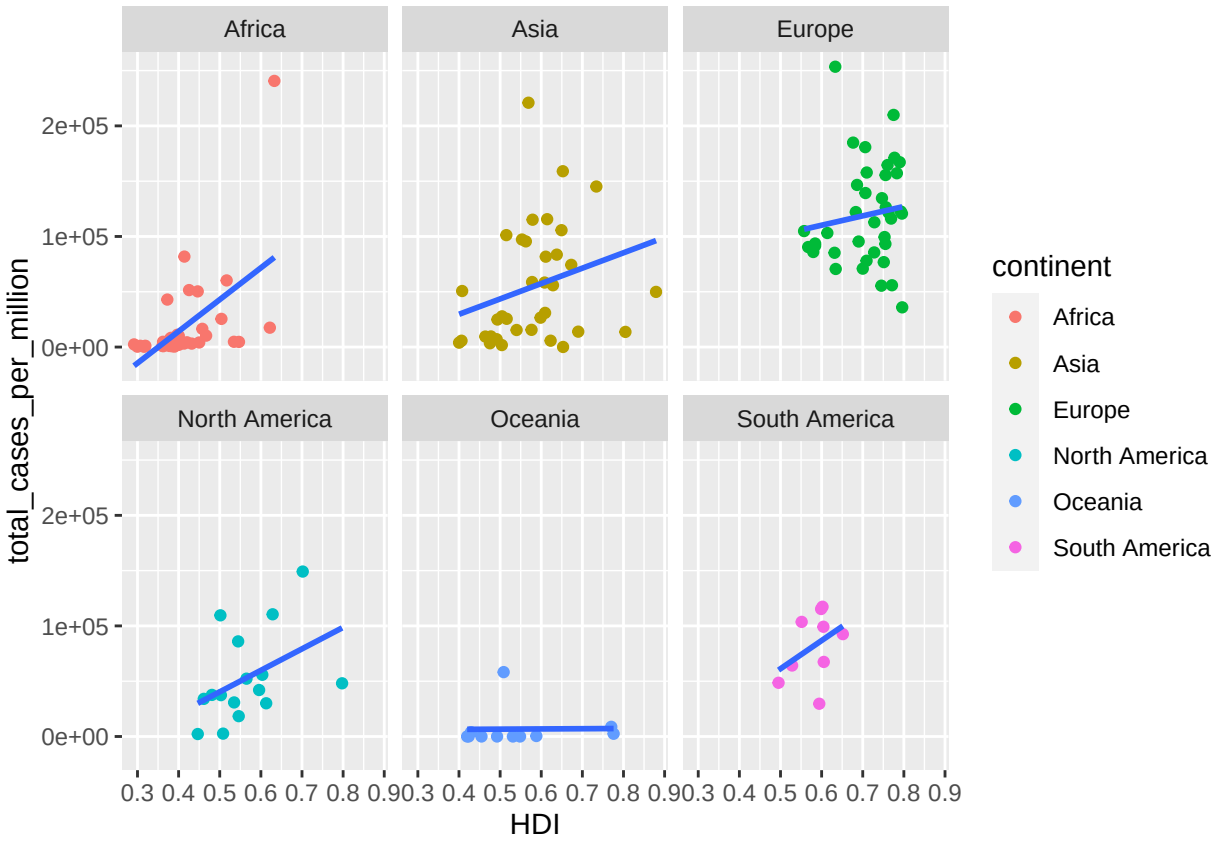
We use the HDI variable, which is the human development index, which evaluates a country's development based on its people, and human growth and not just its economic growth. Our plot shows us that a higher human development index leads to higher covid cases.

```
covid_latest_hdi %>%
  ggplot(aes(x = HDI, y = total_cases_per_million)) +
  geom_point(aes(color = continent)) +
  labs(y = "total cases per million") +
  geom_smooth(se = FALSE)
```



We try to establish a clearer plot by separating by continent. Then we investigate the rate of increase of covid cases with HDI by animating our plot. Where covid cases seems to increase faster for countries with higher HDI.

```
covid_latest_hdi %>%
  ggplot(aes(x = HDI, y = total_cases_per_million)) +
  geom_point(aes(color = continent)) +
  facet_wrap(~continent) +
  geom_smooth(method = "lm", se = FALSE)
```



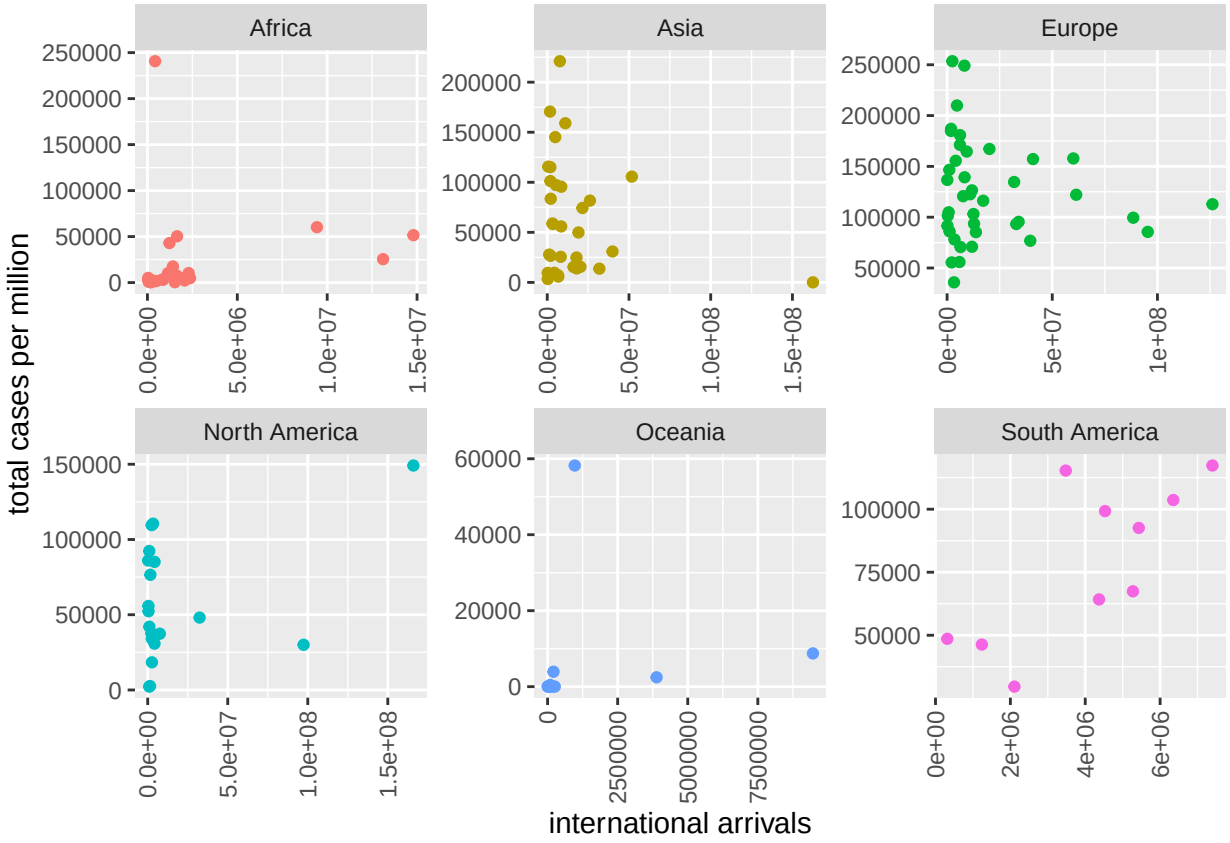

```
hdi_animate <- ggplot(covid_hdi, aes(HDI, total_cases_per_million, color = location)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_color_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  facet_wrap(~continent) +
  labs(title = 'Date: {frame_time}', x = 'HDI', y = 'total cases per million') +
  transition_time(date) +
  ease_aes('linear') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
animate(hdi_animate, duration = 10)
```

Tourism-covid plots

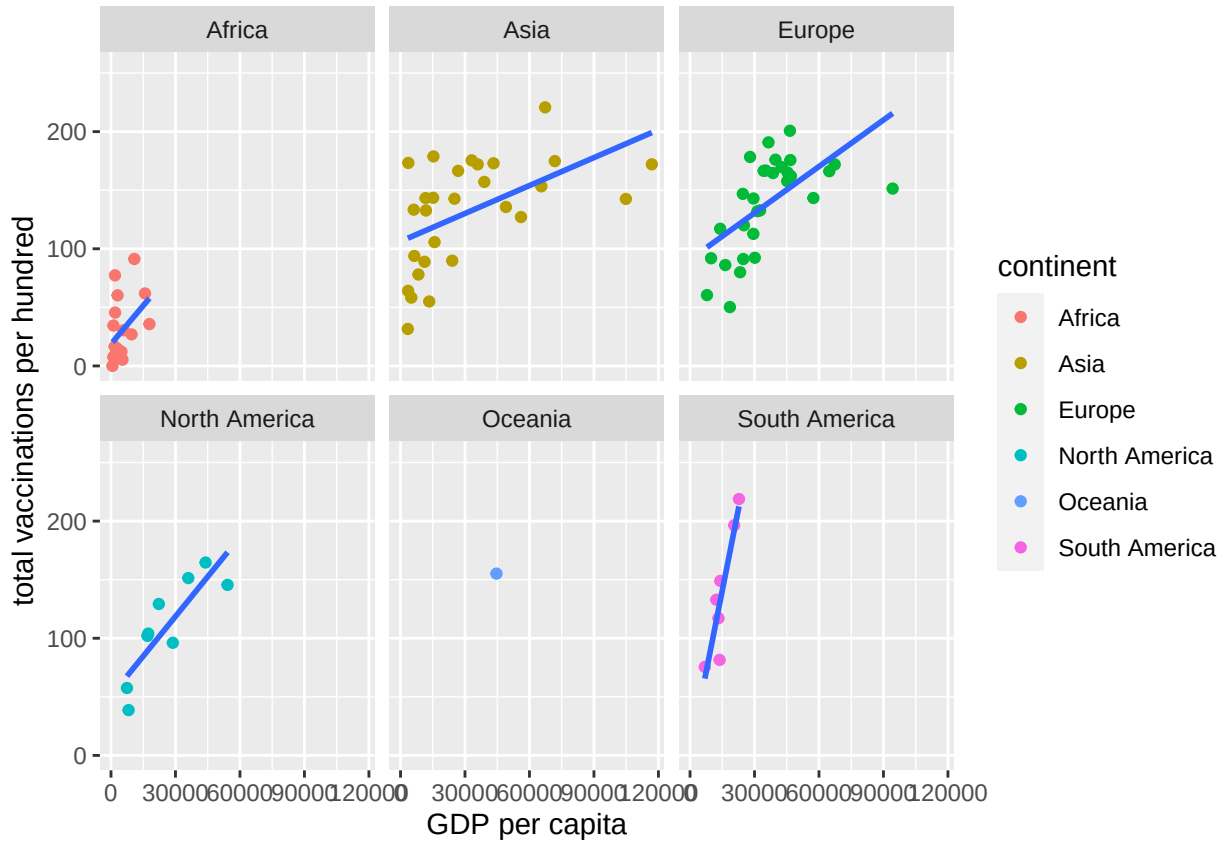
Apart from human development index, we also wanted to explore the how a country's tourism could affect covid. We hypothesise that countries who are heavily reliant on tourism in the past may be more motivated to combat the virus to avoid travel bans. However, after creating plots to investigate the relationship between international arrivals in 2019 and new covid cases, we do not observe a relationship.

```
# data
inter_arrivals <- read.csv("https://raw.githubusercontent.com/ST47S-CompStats-Fall2021/GroupJ-COVID/mai
IA_2019 <- inter_arrivals %>%
  rename(location = Country.Name) %>%
  rename(inter_arrivals = X2019) %>%
  select(location, inter_arrivals)
covid_IA <- left_join(covid, IA_2019, by = "location")
covid_IA <- covid_IA %>%
  filter(date == max(date)) %>%
  filter(!is.na(inter_arrivals)) %>%
  arrange(desc(inter_arrivals))
```

```
covid_IA %>%
  ggplot(aes(x = inter_arrivals, y = total_cases_per_million)) +
  geom_point(aes(color = continent), show.legend = FALSE) +
  facet_wrap(~continent, scales = "free") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(x = "international arrivals", y = "total cases per million")
```




```
covid %>%
  filter(date == max(date)) %>%
  ggplot(aes(x = gdp_per_capita, y = total_vaccinations_per_hundred)) +
  geom_point(aes(color = continent)) +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~continent) +
  labs(x="GDP per capita", y="total vaccinations per hundred")
```



Results

From clustering of the principle components, we observe that the countries of continents Europe and Africa are well clustered together over time. Our brief analysis on external factors such as HDI may give insight on these clusters. From the HDI against total cases plot, we see the stark differences in total cases and HDI for countries in Europe and Africa, where all the countries in Europe have high HDI and high cases, but Africa has low HDI and low cases. Apart from this, the well populated countries, the United States, China, India and Brazil are occasionally clustered together. However the United States is primarily its own cluster. We can see how the most important variables in the principle components has an affect in the clustering. New cases and new tests are heavily weighted in the construction of the principle components, so it is no surprise that countries with the highest increase in cases, are clustered together. With that said, we cannot be certain of causation from our clusters and we would have to extend our research of external factors to find relationships between countries given our covid data.

Discussion

Limitations

Principal component analysis is heavily involved in our work and so it dictates some of our limitations. We could not include any variables in our PCA which were constant due to the normalisation because we can't divide by a standard deviation of 0. As a work around, we filtered out the variables which were constant on

a given day. This is a limitation to our analysis because these variables may have been crucial variable in explaining the variability in our data. To combat this, we could have increased our time interval, from PCA per day to PCA per week or even per month. This would have allowed for the standard deviation of some variables to not be 0. Apart from this, because of the inconsistency of data reporting, NAs are scattered throughout the dataset for many different variables. In order to not remove those variables entirely from our PCA, we had to replace all NAs with 0. This may not have been the best way, rather we could have taken the average between the value prior and after the missing value.

Since we are conducting analyses which affect countries worldwide it is important to consider how the data was collected, who is collecting it and which source are we getting it from. In our case, the data was aggregated from many different reputable sources worldwide, and so we don't believe that our data is centered specifically around any country. However, some metrics in our dataset could lead to misrepresentation of a specific country due to differences in accessible resources to record such a metric. Where some countries would use low touch recording methods, relying on digital systems, and others using high touch recording methods, there could be some variation in the accuracy of data collection and consistency in the reporting.

We try to address some of the misrepresentation and inconsistencies in data reporting by visualising the missing data over time as well as the number of countries reporting data over time. This better informs us of what variables are consistently reported and the time we should begin our analysis. If only 10-15 countries were reporting data at the very beginning of the pandemic, performing PCA and clustering over that time period would not accurately represent the entire world population. However, we are unable to investigate whether the data from each country was accurately collected since that would require researching each individuals countries' data collection methods.

Extension

We can extend our project by incorporating new country specific factors to try to explain some of the PCA clusters. One example of this is using the world bank data to find external factors which influenced the PCA clustering. We can also look at similarities in policies, which would require scraping articles and string parsing to better explain the clusters. We could have also done clustering for pre and post vaccination approval to observe the impact of the vaccination variables in explaining the variability of the data.

Additionals (Something New)

Principal Component Analysis

For our something new, we decided to do principal component analysis because our dataset's variables (outside of indicator variables like location„ iso_code, etc.) are all numeric and the dataset is relatively large (130,000+ observations, 60+ variables). When the data is pivot-wider by date, the dataset has 230+ countries, and 30,000+ variables.

Principal Component Analysis (PCA) is often used with large dataset to reduce dimensionality (combine and syntehsize dataset variables into less variables), which increases simplicity in pattern finding and convinience in model building, while minimizing information loss. PCA does this by preserving as much variability as possible by keeping/weighing uncorrelated variable and reducing weights of correlated variables. Essentially, PCA reduces dimensionality of datasets while maximizing variance. Variables are reduces to principal components (PCs) which are linear combinations of the dataset's variables. Because, PCs are combinations of original variables, it is sometimes difficult to interpret what these PCs actually mean. For example, when two principal components are significant in clustering or predicting, it is difficult to say how our original variables are contributing. With that said, PCs can be ordered by significance and we can view which variables are present in each principal component which gives us insight about patterns in a large dataset, which gives us some variable interpretability.

Low interpretability of principal components. Principal components are linear combinations of the features from the original data, but they are not as easy to interpret

Source: https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202

Shiny App for Visualizing Missingness

Since the data contains many variables across many days and countries we very quickly ran into an issue with missing data. Covid data is sensitive in many different ways. First, the data is very time sensitive. Not every country will be recording data for the variables at the same time. For example vaccine availability dictates when a country can record new vaccinations. Apart from this, each country may be inconsistent as to when they report the data. Some may report only once a week, while others will report daily. Lastly some countries may not have the capabilities of recording certain variables in the dataset at all.

Because we primarily want to explore how covid changes over time, we want to see how the missingness changes over time as well. Filtering by day, we can see the number of missing entries in our data. Since there are an inconsistent number of countries reporting data each day, we normalise the counts by finding the proportion of NAs per day. After plotting the proportions of NAs for each day per variable, the plot becomes difficult to interpret due to the large number of variables. This led to the motivation for the shiny app, which allows you to pick and choose which variables to display on the plot for easier interpretability. This plot reflects the timely impact of the vaccine, where prior to the vaccine approval, they entries were all NA. It also emphasised the inconsistency in data reporting, where the proportion of NAs for the excess_mortality oscillates as time goes on.

Source: https://www.jakobwillforss.com/post/shiny-from-scratch-hands-on-tutorial/

Citations (and R libraries)