pca, clustering, maps, and viz

```
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"
           )
}
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"
           )
}
```

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

vars <- names(covid)

vars
```

```
##  [1] "iso_code"
##  [2] "continent"
##  [3] "location"
##  [4] "date"
##  [5] "total_cases"
##  [6] "new_cases"
##  [7] "new_cases_smoothed"
##  [8] "total_deaths"
##  [9] "new_deaths"
## [10] "new_deaths_smoothed"
## [11] "total_cases_per_million"
## [12] "new_cases_per_million"
## [13] "new_cases_smoothed_per_million"
## [14] "total_deaths_per_million"
## [15] "new_deaths_per_million"
```

```
## [16] "new_deaths_smoothed_per_million"
## [17] "reproduction_rate"
## [18] "icu_patients"
## [19] "icu_patients_per_million"
## [20] "hosp_patients"
## [21] "hosp_patients_per_million"
## [22] "weekly_icu_admissions"
## [23] "weekly_icu_admissions_per_million"
## [24] "weekly_hosp_admissions"
## [25] "weekly_hosp_admissions_per_million"
## [26] "new_tests"
## [27] "total_tests"
## [28] "total_tests_per_thousand"
## [29] "new_tests_per_thousand"
## [30] "new_tests_smoothed"
## [31] "new_tests_smoothed_per_thousand"
## [32] "positive_rate"
## [33] "tests_per_case"
## [34] "tests_units"
## [35] "total_vaccinations"
## [36] "people_vaccinated"
## [37] "people_fully_vaccinated"
## [38] "total_boosters"
## [39] "new_vaccinations"
## [40] "new_vaccinations_smoothed"
## [41] "total_vaccinations_per_hundred"
## [42] "people_vaccinated_per_hundred"
## [43] "people_fully_vaccinated_per_hundred"
## [44] "total_boosters_per_hundred"
## [45] "new_vaccinations_smoothed_per_million"
## [46] "new_people_vaccinated_smoothed"
## [47] "new_people_vaccinated_smoothed_per_hundred"
## [48] "stringency_index"
## [49] "population"
## [50] "population_density"
## [51] "median_age"
## [52] "aged_65_older"
## [53] "aged_70_older"
## [54] "gdp_per_capita"
## [55] "extreme_poverty"
## [56] "cardiovasc_death_rate"
## [57] "diabetes_prevalence"
## [58] "female_smokers"
## [59] "male_smokers"
## [60] "handwashing_facilities"
## [61] "hospital_beds_per_thousand"
## [62] "life_expectancy"
## [63] "human_development_index"
## [64] "excess_mortality_cumulative_absolute"
## [65] "excess_mortality_cumulative"
## [66] "excess_mortality"
## [67] "excess_mortality_cumulative_per_million"
```

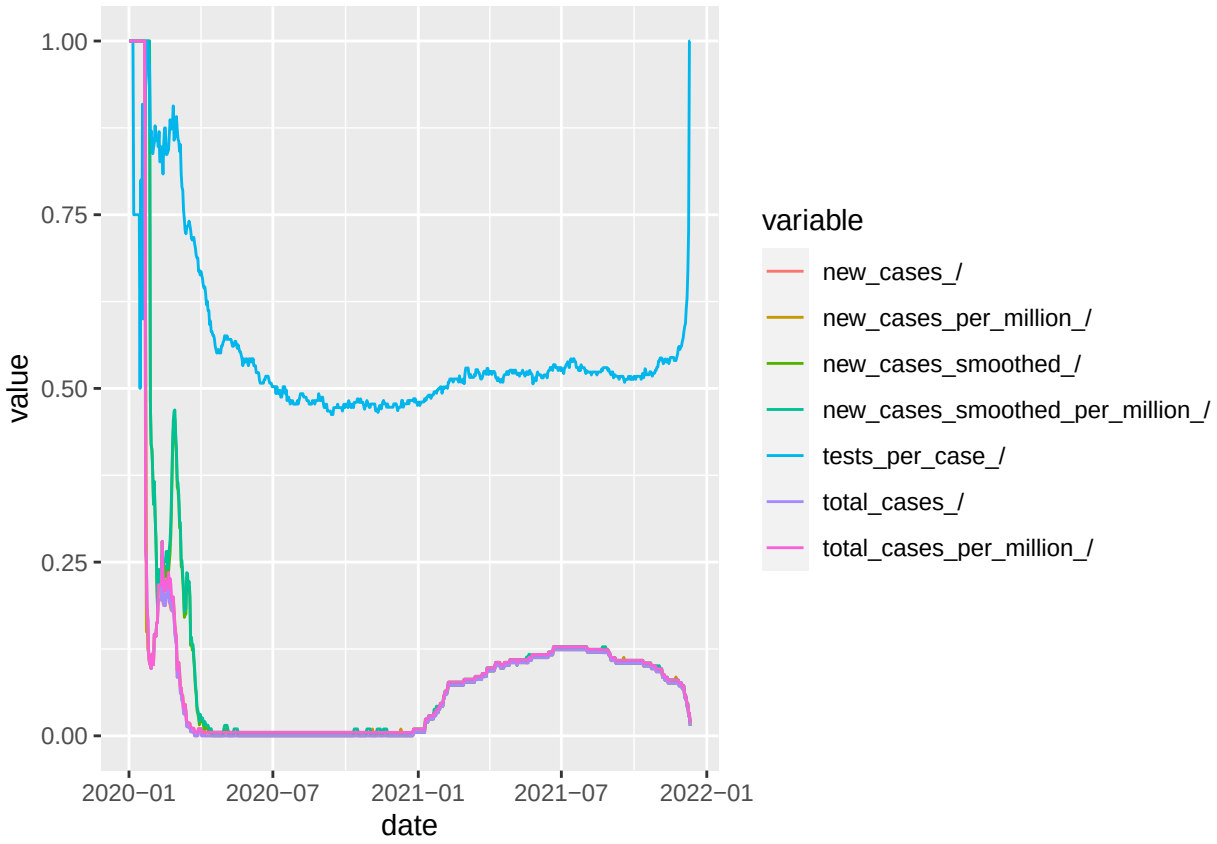
```
vars <- vars[! vars %in% c('date')]

sum <- covid %>% group_by(date) %>%
  summarise(across(vars, ~ sum(is.na(.))))
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(vars)‘ instead of ‘vars‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

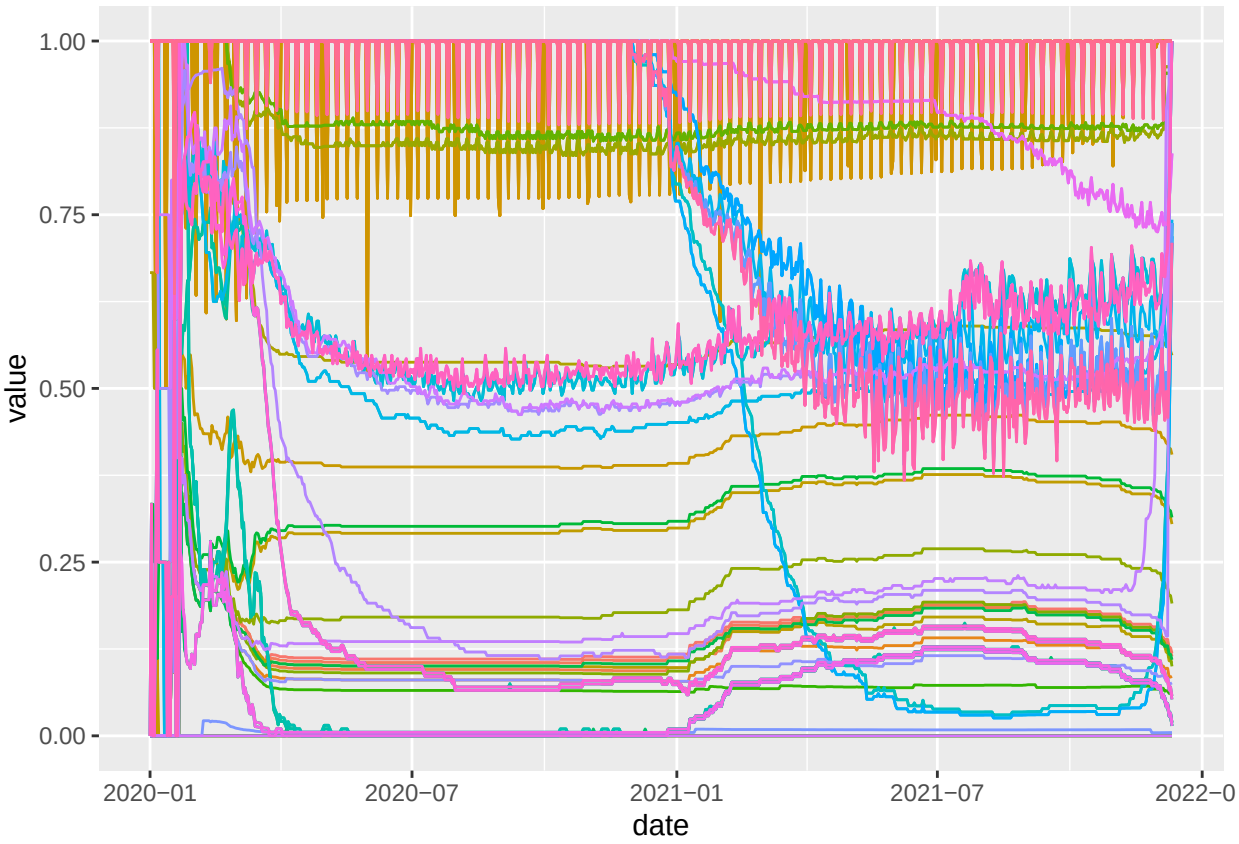
```
prop <- covid %>% group_by(date) %>%
  summarise(across(vars, funs(sum(is.na(.)) / length(.))))
```

```
## Warning: ‘funs()‘ was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with ‘tibble::lst()‘:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.
```

```
test <- covid %>% filter(date == "2020-01-01")

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

NAsummary %>% filter(grepl('case', variable)) %>% ggplot(aes(x=date, y=value, color=variable)) + geom_li
```



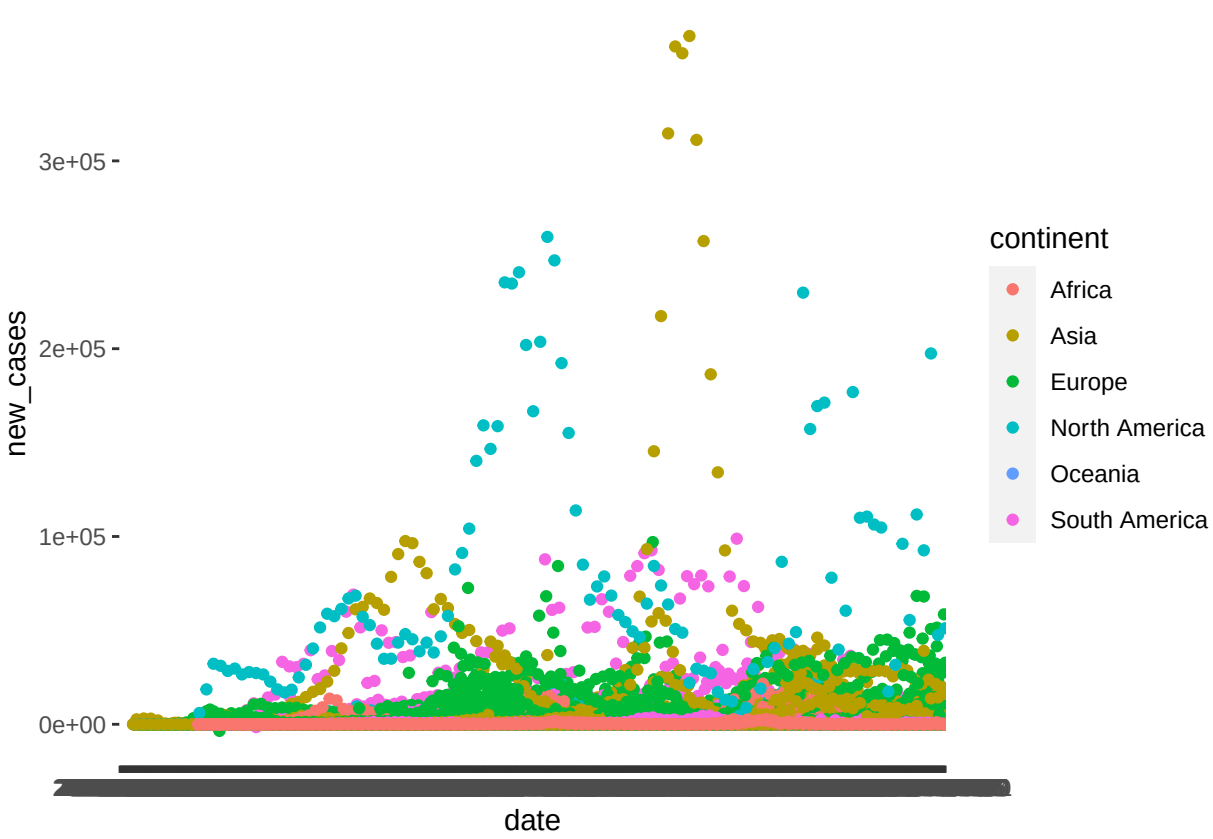

```
NAsummary %>% ggplot(aes(x=date, y=value, color=variable)) +
  geom_line() +
  theme(legend.position = "none")
```




```
covid %>% filter(continent==c("Africa", "Asia", "Europe", "North America", "Oceania", "South America"))
  geom_point()
```

```
## Warning in continent == c("Africa", "Asia", "Europe", "North America",
## "Oceania", : longer object length is not a multiple of shorter object length
```

```
## Warning: Removed 1297 rows containing missing values (geom_point).
```




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

data.frame(colSums(is.na(one_day)))
```

```
##                                    colSums.is.na.one_day..
## iso_code                                                 0
## continent                                                0
## location                                                 0
## date                                                     0
## total_cases                                             12
## new_cases                                               12
## new_cases_smoothed                                      12
## total_deaths                                            20
## new_deaths                                              20
## new_deaths_smoothed                                     12
## total_cases_per_million                                 13
## new_cases_per_million                                   13
## new_cases_smoothed_per_million                          13
## total_deaths_per_million                                21
## new_deaths_per_million                                  21
## new_deaths_smoothed_per_million                         13
## reproduction_rate                                       34
## icu_patients                                           193
## icu_patients_per_million                               193
## hosp_patients                                          191
## hosp_patients_per_million                              191
```

```
## weekly_icu_admissions                                  203
## weekly_icu_admissions_per_million                      203
## weekly_hosp_admissions                                 194
## weekly_hosp_admissions_per_million                     194
## new_tests                                              151
## total_tests                                            145
## total_tests_per_thousand                               145
## new_tests_per_thousand                                 151
## new_tests_smoothed                                     129
## new_tests_smoothed_per_thousand                        129
## positive_rate                                          130
## tests_per_case                                         130
## tests_units                                              0
## total_vaccinations                                     107
## people_vaccinated                                      108
## people_fully_vaccinated                                109
## total_boosters                                         161
## new_vaccinations                                       139
## new_vaccinations_smoothed                               42
## total_vaccinations_per_hundred                         107
## people_vaccinated_per_hundred                          108
## people_fully_vaccinated_per_hundred                    109
## total_boosters_per_hundred                             161
## new_vaccinations_smoothed_per_million                   42
## new_people_vaccinated_smoothed                          49
## new_people_vaccinated_smoothed_per_hundred              49
## stringency_index                                       131
## population                                               1
## population_density                                      19
## median_age                                              29
## aged_65_older                                           31
## aged_70_older                                           30
## gdp_per_capita                                          29
## extreme_poverty                                         94
## cardiovasc_death_rate                                   30
## diabetes_prevalence                                     22
## female_smokers                                          73
## male_smokers                                            75
## handwashing_facilities                                 124
## hospital_beds_per_thousand                              49
## life_expectancy                                         15
## human_development_index                                 30
## excess_mortality_cumulative_absolute                   219
## excess_mortality_cumulative                            219
## excess_mortality                                       219
## excess_mortality_cumulative_per_million                219
```

```
library(tidytext)

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

```r
one_day_prepped <- replace_all_na(one_day) %>%
  filter_continents() %>%
  filter_world()

one_day_prepped %>% summarise_all(mean)
```

```
## Warning in mean.default(iso_code): argument is not numeric or logical: returning
## NA

## Warning in mean.default(continent): argument is not numeric or logical:
## returning NA

## Warning in mean.default(location): argument is not numeric or logical: returning
## NA

## Warning in mean.default(date): argument is not numeric or logical: returning NA

## Warning in mean.default(tests_units): argument is not numeric or logical:
## returning NA
```

```
##   iso_code continent location date total_cases new_cases new_cases_smoothed
## 1       NA        NA       NA   NA     1290664  2119.796           3022.135
##   total_deaths new_deaths new_deaths_smoothed total_cases_per_million
## 1     25514.57   34.79126             38.6207                53924.3
##   new_cases_per_million new_cases_smoothed_per_million total_deaths_per_million
## 1               101.984                       179.1096                862.1628
##   new_deaths_per_million new_deaths_smoothed_per_million reproduction_rate
## 1               1.178481                       1.717874         0.8646117
##   icu_patients icu_patients_per_million hosp_patients hosp_patients_per_million
## 1      144.699                 4.622437      738.5243                   38.0162
##   weekly_icu_admissions weekly_icu_admissions_per_million
## 1              33.80097                         2.387816
##   weekly_hosp_admissions weekly_hosp_admissions_per_million new_tests
## 1               474.4029                           18.7533  27951.82
##   total_tests total_tests_per_thousand new_tests_per_thousand
## 1    12395883                623.2899               1.502718
##   new_tests_smoothed new_tests_smoothed_per_thousand positive_rate
## 1           41265.66                       2.740427    0.03386359
##   tests_per_case tests_units total_vaccinations people_vaccinated
## 1       102.4893          NA           35598652          12586292
##   people_fully_vaccinated total_boosters new_vaccinations
## 1                 9775261       951449.1         93609.01
##   new_vaccinations_smoothed total_vaccinations_per_hundred
## 1                  170832.5                        55.9501
##   people_vaccinated_per_hundred people_fully_vaccinated_per_hundred
## 1                      27.78374                             24.41709
##   total_boosters_per_hundred new_vaccinations_smoothed_per_million
## 1                   3.008738                              2656.733
##   new_people_vaccinated_smoothed new_people_vaccinated_smoothed_per_hundred
## 1                       40479.73                                  0.06984951
##   stringency_index population population_density median_age aged_65_older
## 1         18.68476   38051694           439.8572    27.8165      7.834165
```

```
##   aged_70_older gdp_per_capita extreme_poverty cardiovasc_death_rate
## 1      4.970602       17419.53        8.405825              240.4559
##   diabetes_prevalence female_smokers male_smokers handwashing_facilities
## 1            7.859563       7.459709     22.81748               22.89126
##   hospital_beds_per_thousand life_expectancy human_development_index
## 1                   2.463485        72.09854               0.659335
##   excess_mortality_cumulative_absolute excess_mortality_cumulative
## 1                                    0                           0
##   excess_mortality excess_mortality_cumulative_per_million
## 1                0                                      0
```

```
ready <- one_day_prepped %>% select(!c(excess_mortality_cumulative, excess_mortality, excess_mortality_

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

pca_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##  predictor         62
##
## Operations:
##
## Centering for all_numeric()
## Scaling for all_numeric()
## No PCA components were extracted.
```

```
pca_estimates <- prep(pca_recipe)

juice(pca_estimates)
```

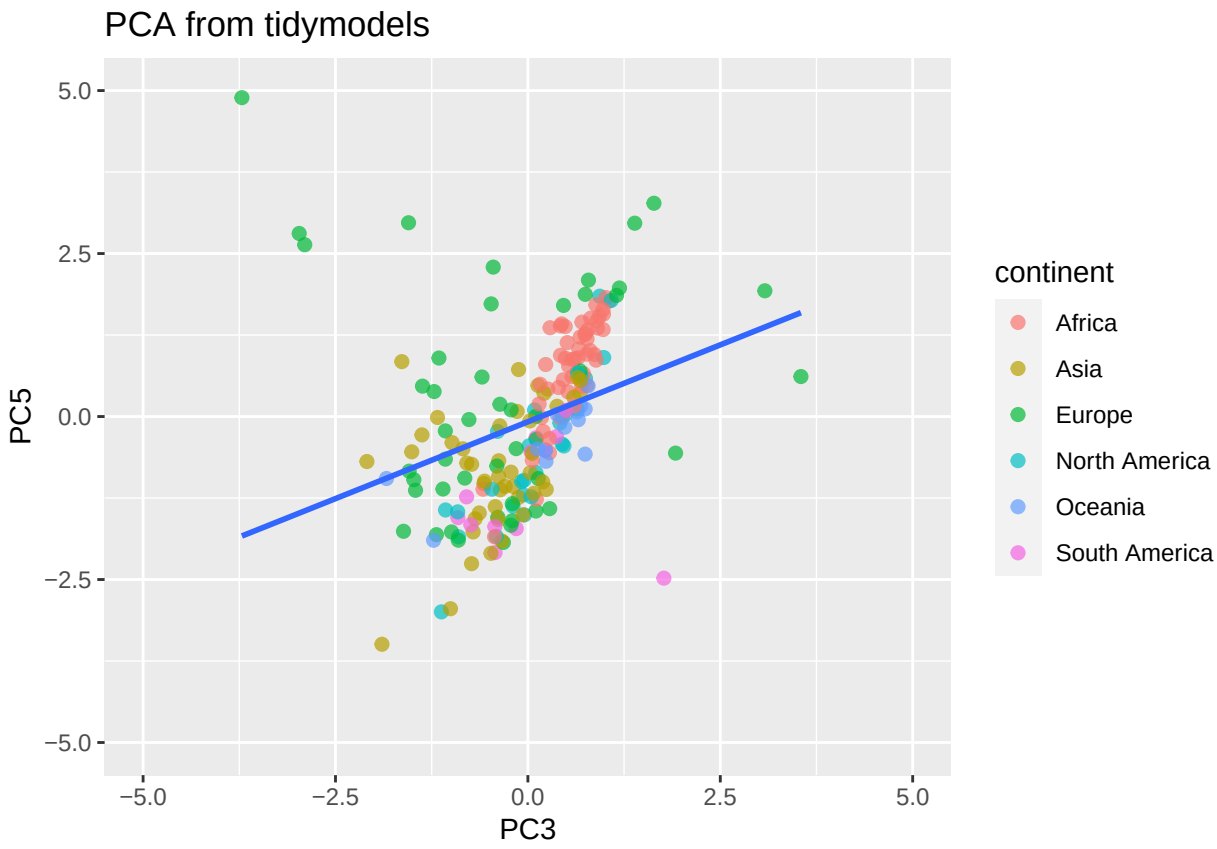
```
## # A tibble: 206 x 10
##    iso_code continent  location date  tests_units     PC1    PC2     PC3     PC4
##    <fct>    <fct>      <fct>    <fct> <fct>         <dbl>  <dbl>   <dbl>   <dbl>
##  1 AFG      Asia       Afghani~ 2021~ ""           -3.12   -1.08   0.681   0.924
##  2 ALB      Europe     Albania  2021~ ""            0.0215  1.03  -0.390  -0.167
##  3 DZA      Africa     Algeria  2021~ ""           -2.15   -0.431  0.280   0.412
##  4 AND      Europe     Andorra  2021~ "people te~   1.28    2.49  -0.596   0.217
##  5 AGO      Africa     Angola   2021~ ""           -2.93   -1.11   0.651   0.0526
##  6 AIA      North Ame~ Anguilla 2021~ ""           -3.31   -1.21   0.934  -0.969
##  7 ATG      North Ame~ Antigua~ 2021~ ""           -0.735   0.544 -0.0448 -1.41
##  8 ARG      South Ame~ Argenti~ 2021~ ""            2.09    0.328 -0.427  -1.35
##  9 ARM      Asia       Armenia  2021~ "tests per~  0.239   1.29  -0.570   2.37
## 10 ABW      North Ame~ Aruba    2021~ ""           -2.02   -0.221  0.417  -1.09
## # ... with 196 more rows, and 1 more variable: PC5 <dbl>
```

```
juice(pca_estimates) %>%
  ggplot(aes(PC3, PC5)) +
  geom_point(aes(color = continent), alpha = 0.7, size = 2)+
  labs(title="PCA from tidymodels") +
  xlim(-5, 5) + ylim(-5, 5) +
  geom_smooth(method = "lm", se = FALSE)
```

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

```
## Warning: Removed 6 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```

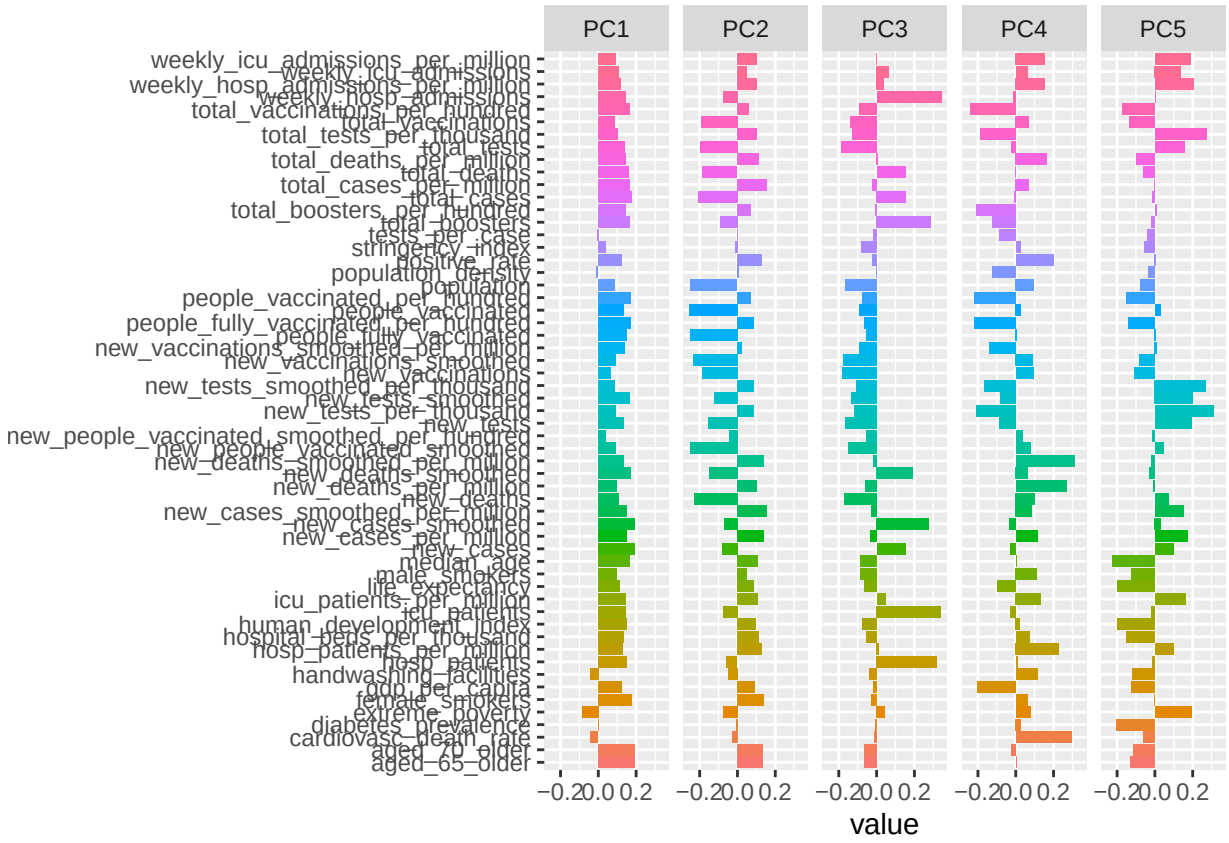



```
tidied_pca <- tidy(pca_estimates, 2)

tidy_pca_loadings <- pca_estimates%>%
  tidy(id = "pca")

tidy_pca_loadings
```

```
## # A tibble: 3,249 x 4
##    terms                          value component id
##    <chr>                          <dbl> <chr>     <chr>
##  1 total_cases                    0.177 PC1       pca
```

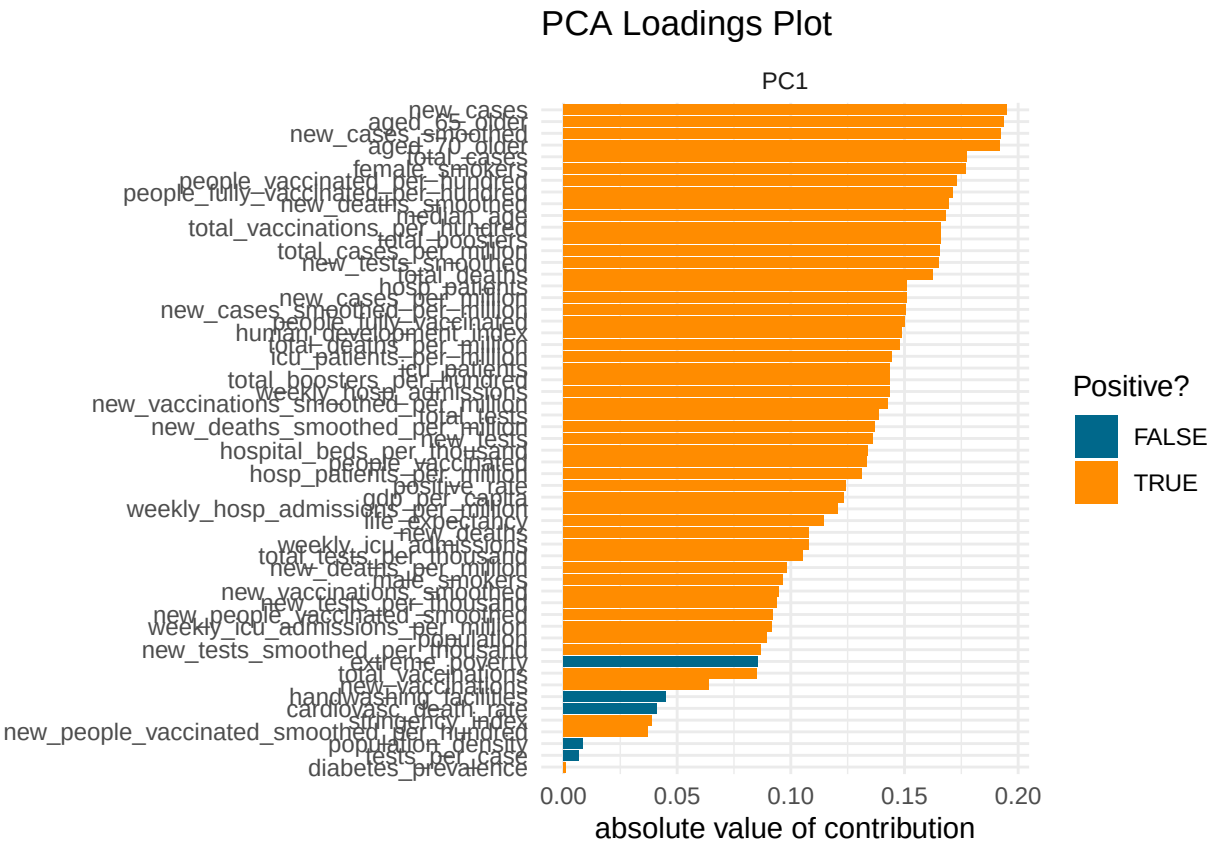
```
##  2 new_cases                         0.195 PC1       pca
##  3 new_cases_smoothed                0.192 PC1       pca
##  4 total_deaths                      0.162 PC1       pca
##  5 new_deaths                        0.108 PC1       pca
##  6 new_deaths_smoothed               0.169 PC1       pca
##  7 total_cases_per_million           0.165 PC1       pca
##  8 new_cases_per_million             0.151 PC1       pca
##  9 new_cases_smoothed_per_million 0.150 PC1       pca
## 10 total_deaths_per_million         0.148 PC1       pca
## # ... with 3,239 more rows
```

```
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)
```




```
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()


plot_loadings
```




```
juice_df <- juice(pca_estimates)

head(juice_df)
```

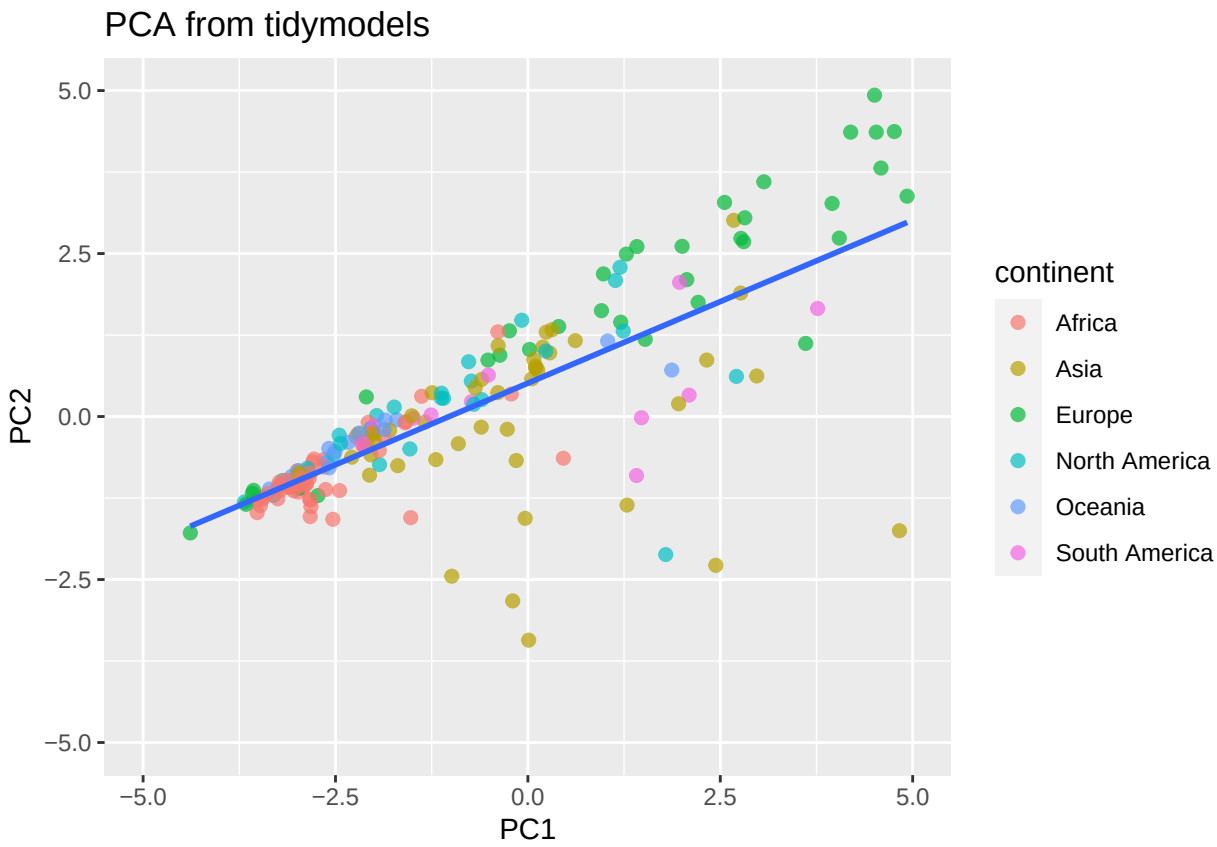
```
## # A tibble: 6 x 10
##   iso_code continent  location  date   tests_units     PC1    PC2    PC3     PC4
##   <fct>    <fct>      <fct>     <fct>  <fct>         <dbl>  <dbl>  <dbl>   <dbl>
## 1 AFG      Asia       Afghanis~ 2021-~ ""           -3.12   -1.08   0.681  0.924
## 2 ALB      Europe     Albania   2021-~ ""            0.0215  1.03  -0.390 -0.167
## 3 DZA      Africa     Algeria   2021-~ ""           -2.15   -0.431  0.280  0.412
## 4 AND      Europe     Andorra   2021-~ "people te~  1.28    2.49  -0.596  0.217
## 5 AGO      Africa     Angola    2021-~ ""           -2.93   -1.11   0.651  0.0526
## 6 AIA      North Ame~ Anguilla  2021-~ ""           -3.31   -1.21   0.934 -0.969
## # ... with 1 more variable: PC5 <dbl>
```

```
juice_df %>%
  ggplot(aes(PC1,PC2)) +
  geom_point(aes(color = continent), alpha = 0.7, size = 2)+
  labs(title="PCA from tidymodels") +
  xlim(-5, 5) + ylim(-5, 5) +
  geom_smooth(method = "lm", se = FALSE)
```

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

```
## Warning: Removed 19 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 19 rows containing missing values (geom_point).
```

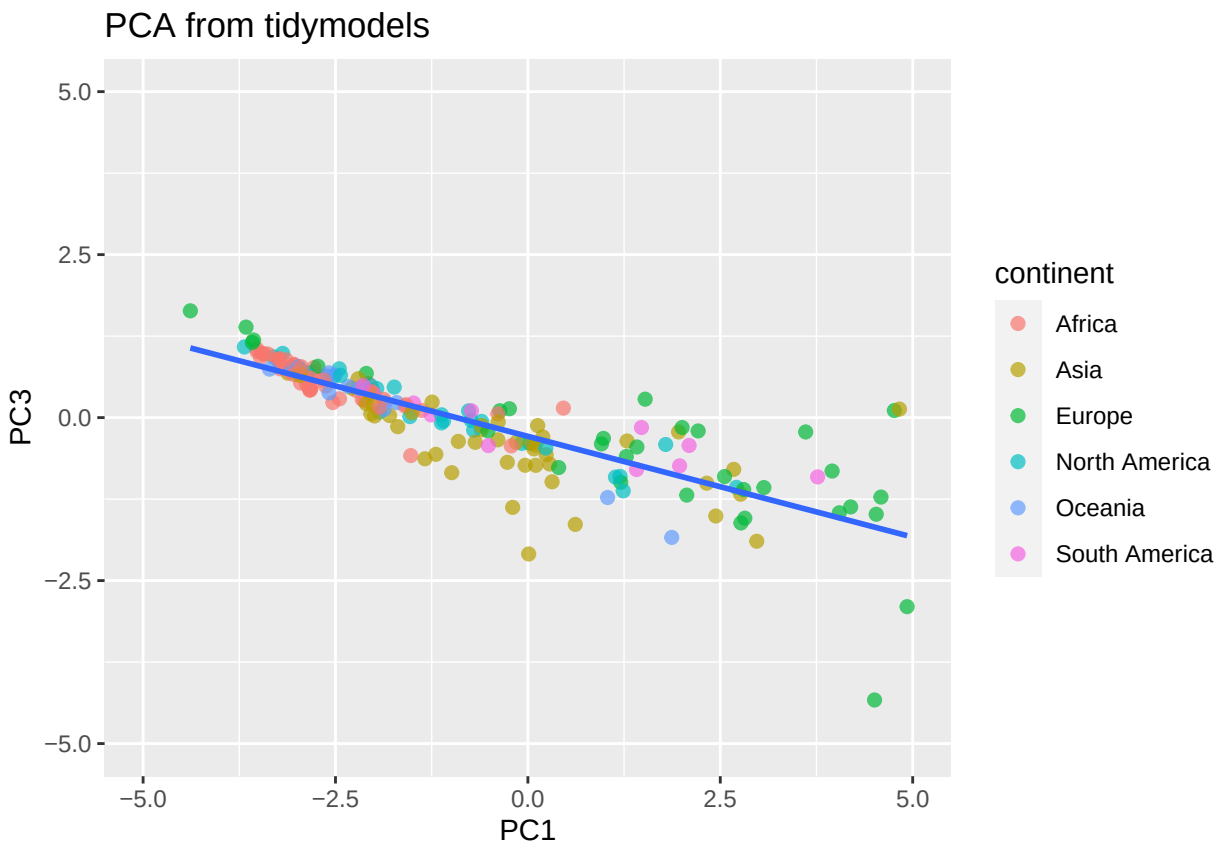



```
juice_df %>%
  ggplot(aes(PC1,PC3)) +
  geom_point(aes(color = continent), alpha = 0.7, size = 2)+
  labs(title="PCA from tidymodels") +
  xlim(-5, 5) + ylim(-5, 5) +
  geom_smooth(method = "lm", se = FALSE)
```

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

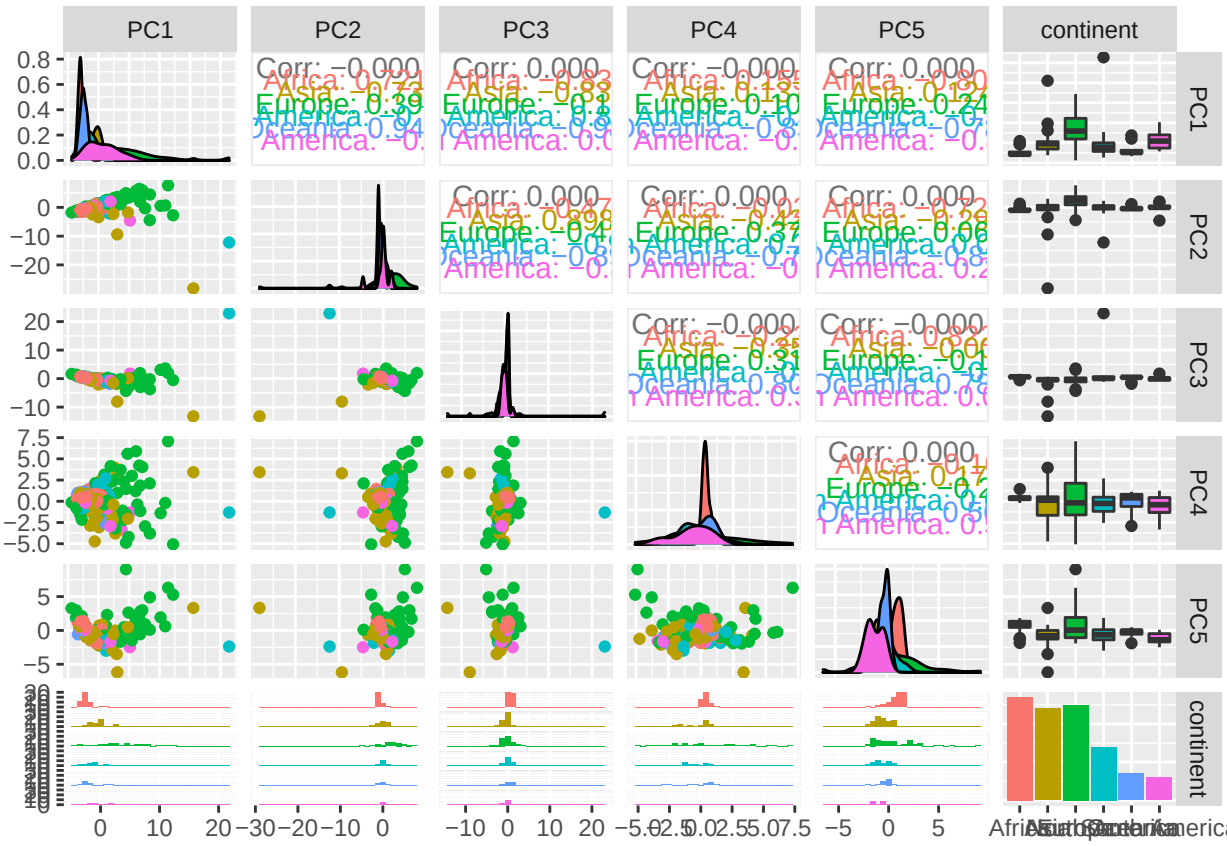
```
## Warning: Removed 19 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 19 rows containing missing values (geom_point).
```




```
juice_df %>%
  select(starts_with("PC"), continent) %>%
  ggpairs(aes(color = continent))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

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

pca_kclust
```

```
## K-means clustering with 6 clusters of sizes 1, 80, 1, 60, 21, 43
##
## Cluster means:
##          PC1          PC2          PC3        PC4        PC5
## 1 21.3975284 -12.2031657  22.9196562 -1.3181022 -2.3765386
## 2 -2.8271949  -0.8887750   0.6716781  0.1697325  0.8760519
## 3 15.5326049 -28.1486605 -13.2352568  3.4287894  3.3223630
## 4 -0.7300093  -0.1834767  -0.3017339  0.1958738 -0.9498453
## 5  7.0621406   3.1833484  -0.5798539  0.6268295  1.8281847
## 6  1.9707223   1.2933050  -0.7706439 -0.9443055 -1.2193287
##
## Clustering vector:
##   [1] 2 4 4 6 2 2 4 6 4 2 6 5 4 4 6 4 6 6 5 4 2 2 4 4 4 4 6 6 5 2 2 4 2 6 2 2 2
##  [38] 6 4 6 2 2 4 2 5 6 2 5 5 2 5 2 2 4 4 4 4 2 2 6 2 2 4 6 5 2 2 2 6 5 2 2 5 2
##  [75] 4 4 2 2 2 4 2 2 6 5 6 3 4 4 2 5 2 6 5 4 6 2 4 4 2 4 2 4 4 2 5 4 2 2 4 6 5
## [112] 6 6 2 2 6 6 2 6 2 2 4 4 2 4 2 6 6 4 2 2 2 4 5 2 6 2 2 2 4 6 4 4 2 2 6 2 4
## [149] 6 4 5 6 4 6 5 2 2 4 4 2 4 2 4 2 6 4 2 4 5 5 2 2 4 6 2 6 4 2 4 6 6 2 4 4 2
```

```
## [186] 4 2 2 4 6 4 6 2 6 4 5 1 6 4 2 2 2 6 2 2 2
##
## Within cluster sum of squares by cluster:
## [1]   0.0000 109.8724   0.0000 416.4560 783.3684 428.3043
##  (between_SS / total_SS =  74.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
loc_clusters <- pca_kclust %>% augment(juice_df)
loc_clusters
```

```
## # A tibble: 206 x 11
##    iso_code continent  location date  tests_units     PC1    PC2     PC3     PC4
##    <fct>    <fct>      <fct>    <fct> <fct>         <dbl>  <dbl>   <dbl>   <dbl>
##  1 AFG      Asia       Afghani~ 2021~ ""           -3.12   -1.08   0.681   0.924
##  2 ALB      Europe     Albania  2021~ ""            0.0215  1.03  -0.390  -0.167
##  3 DZA      Africa     Algeria  2021~ ""           -2.15   -0.431  0.280   0.412
##  4 AND      Europe     Andorra  2021~ "people te~   1.28    2.49  -0.596   0.217
##  5 AGO      Africa     Angola   2021~ ""           -2.93   -1.11   0.651   0.0526
##  6 AIA      North Ame~ Anguilla 2021~ ""           -3.31   -1.21   0.934  -0.969
##  7 ATG      North Ame~ Antigua~ 2021~ ""           -0.735   0.544 -0.0448 -1.41
##  8 ARG      South Ame~ Argenti~ 2021~ ""            2.09    0.328 -0.427  -1.35
##  9 ARM      Asia       Armenia  2021~ "tests per~  0.239   1.29  -0.570   2.37
## 10 ABW      North Ame~ Aruba    2021~ ""           -2.02   -0.221  0.417  -1.09
## # ... with 196 more rows, and 2 more variables: PC5 <dbl>, .cluster <fct>
```

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

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

combining long-lat data

```
long_lat_clusters <- left_join(loc_clusters, lats_long, by="location")
head(long_lat_clusters)
```

```
## # A tibble: 6 x 14
##   iso_code continent  location  date   tests_units     PC1    PC2    PC3     PC4
##   <fct>    <fct>      <chr>     <fct>  <fct>         <dbl>  <dbl>  <dbl>   <dbl>
## 1 AFG      Asia       Afghanis~ 2021-~ ""           -3.12   -1.08   0.681  0.924
## 2 ALB      Europe     Albania   2021-~ ""            0.0215  1.03  -0.390 -0.167
## 3 DZA      Africa     Algeria   2021-~ ""           -2.15   -0.431  0.280  0.412
## 4 AND      Europe     Andorra   2021-~ "people te~   1.28    2.49  -0.596  0.217
## 5 AGO      Africa     Angola    2021-~ ""           -2.93   -1.11   0.651  0.0526
## 6 AIA      North Ame~ Anguilla  2021-~ ""           -3.31   -1.21   0.934 -0.969
## # ... with 5 more variables: PC5 <dbl>, .cluster <fct>,
## #   ISO.3166.Country.Code <chr>, Latitude <dbl>, Longitude <dbl>
```

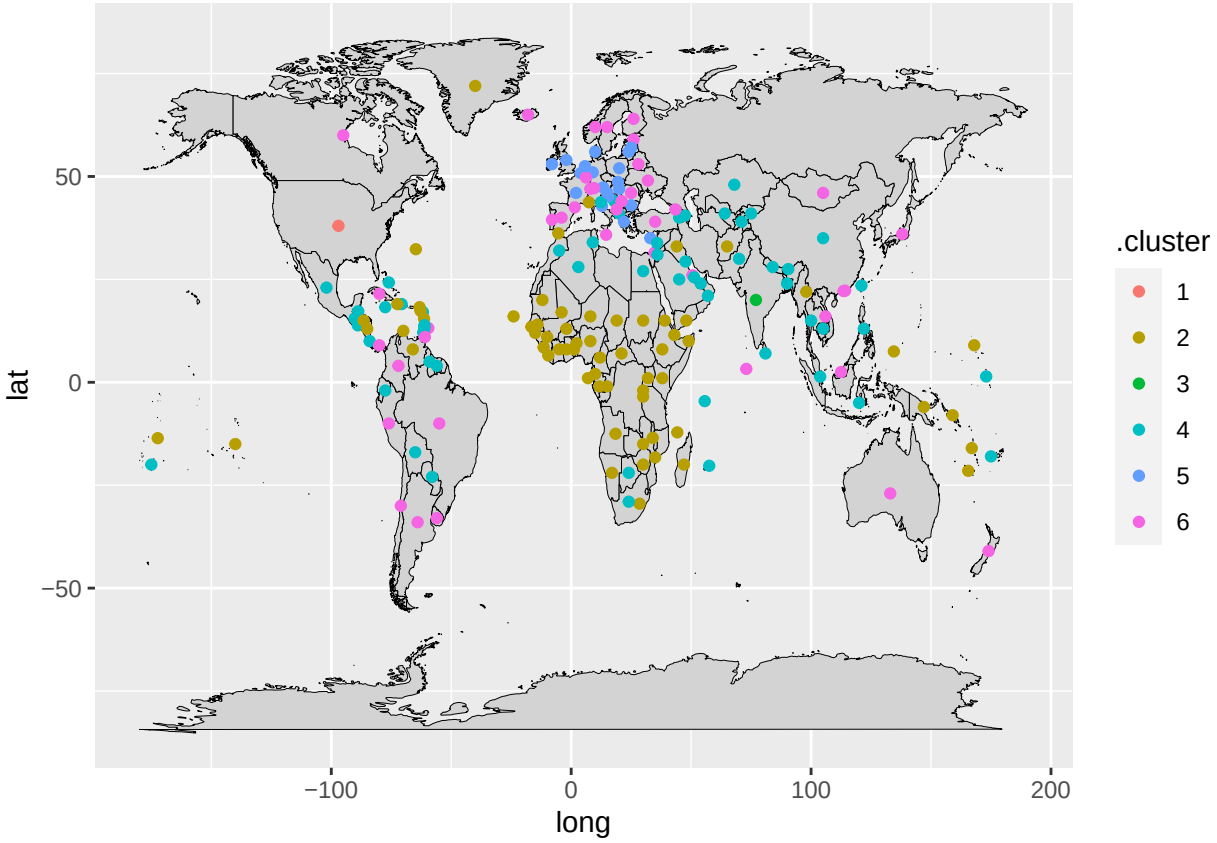
Now, we plot everything on a map, this time coloring the locations based on their cluster

```
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 = long_lat_clusters, aes(x=Longitude, y=Latitude, color=.cluster))
```

```
## Warning: Ignoring unknown aesthetics: x, y
```

```
map
```

```
## Warning: Removed 23 rows containing missing values (geom_point).
```



PCA-Clustering by Day

vaccinations started 2020-12-16

We want to loop through the days of the data since 2020-12-16. To do this, we first convert the `date` variable from a factor to a type `Date`, and increment for each loop. We maintain some global dataframe that consists of the dates we have already iterated through. For each iteration, we call rbind with the global dataframe

and the local dataframe that contains only information from a specific data. To create the local dataframe, we will be performing PCA on the data for a given day, and then we will perform clustering based on the PCAs and augment the cluster labels with the non-PCA. After iterating through all the days, we will have a dataframe with columns `Location`, `date`, `cluster`. At this point, we will cbind this data with the longitude and latitude data. The resultant dataframe will then allow us to plot on a map of the world the cluster labels, so that we can try to gain a sense of whether there is a spatial relationship between clusters. We will create an animation that shows the clusters based on the PCAs over time. Our hypothesis is that data points in the same location will tend to be in the same cluster, even as time progresses. Furthermore, we predict that changes in groupings will also occur based on spatial relations. Note that clusters will have no relation at all between timesteps. Therefore, we will not measure whether or not the color of a certain location is different between frames, but whether the (arbitrary) color of a point is the same as the color of points near it even as time progresses.

converting date data into date type

```
cleaned_covid <- covid %>%
  replace_all_na() %>%
  filter_continents() %>%
  filter_world() %>%
  mutate(date = as.Date(date))
```

```
cleaned_covid %>% head()
```

```
##   iso_code continent    location       date total_cases new_cases
## 1      AFG      Asia Afghanistan 2020-02-24           5         5
## 2      AFG      Asia Afghanistan 2020-02-25           5         0
## 3      AFG      Asia Afghanistan 2020-02-26           5         0
## 4      AFG      Asia Afghanistan 2020-02-27           5         0
## 5      AFG      Asia Afghanistan 2020-02-28           5         0
## 6      AFG      Asia Afghanistan 2020-02-29           5         0
##   new_cases_smoothed total_deaths new_deaths new_deaths_smoothed
## 1              0.000            0          0                   0
## 2              0.000            0          0                   0
## 3              0.000            0          0                   0
## 4              0.000            0          0                   0
## 5              0.000            0          0                   0
## 6              0.714            0          0                   0
##   total_cases_per_million new_cases_per_million new_cases_smoothed_per_million
## 1                   0.126                 0.126                           0.000
## 2                   0.126                 0.000                           0.000
## 3                   0.126                 0.000                           0.000
## 4                   0.126                 0.000                           0.000
## 5                   0.126                 0.000                           0.000
## 6                   0.126                 0.000                           0.018
##   total_deaths_per_million new_deaths_per_million
## 1                        0                      0
## 2                        0                      0
## 3                        0                      0
## 4                        0                      0
## 5                        0                      0
## 6                        0                      0
##   new_deaths_smoothed_per_million reproduction_rate icu_patients
```

```
## 1                             0                0           0
## 2                             0                0           0
## 3                             0                0           0
## 4                             0                0           0
## 5                             0                0           0
## 6                             0                0           0
##   icu_patients_per_million hosp_patients hosp_patients_per_million
## 1                        0            0                         0
## 2                        0            0                         0
## 3                        0            0                         0
## 4                        0            0                         0
## 5                        0            0                         0
## 6                        0            0                         0
##   weekly_icu_admissions weekly_icu_admissions_per_million
## 1                     0                                 0
## 2                     0                                 0
## 3                     0                                 0
## 4                     0                                 0
## 5                     0                                 0
## 6                     0                                 0
##   weekly_hosp_admissions weekly_hosp_admissions_per_million new_tests
## 1                      0                                 0         0
## 2                      0                                 0         0
## 3                      0                                 0         0
## 4                      0                                 0         0
## 5                      0                                 0         0
## 6                      0                                 0         0
##   total_tests total_tests_per_thousand new_tests_per_thousand
## 1           0                        0                     0
## 2           0                        0                     0
## 3           0                        0                     0
## 4           0                        0                     0
## 5           0                        0                     0
## 6           0                        0                     0
##   new_tests_smoothed new_tests_smoothed_per_thousand positive_rate
## 1                  0                               0             0
## 2                  0                               0             0
## 3                  0                               0             0
## 4                  0                               0             0
## 5                  0                               0             0
## 6                  0                               0             0
##   tests_per_case tests_units total_vaccinations people_vaccinated
## 1              0                              0                 0
## 2              0                              0                 0
## 3              0                              0                 0
## 4              0                              0                 0
## 5              0                              0                 0
## 6              0                              0                 0
##   people_fully_vaccinated total_boosters new_vaccinations
## 1                       0             0                0
## 2                       0             0                0
## 3                       0             0                0
## 4                       0             0                0
## 5                       0             0                0
```

```
## 6                       0             0                0
##   new_vaccinations_smoothed total_vaccinations_per_hundred
## 1                         0                              0
## 2                         0                              0
## 3                         0                              0
## 4                         0                              0
## 5                         0                              0
## 6                         0                              0
##   people_vaccinated_per_hundred people_fully_vaccinated_per_hundred
## 1                             0                                    0
## 2                             0                                    0
## 3                             0                                    0
## 4                             0                                    0
## 5                             0                                    0
## 6                             0                                    0
##   total_boosters_per_hundred new_vaccinations_smoothed_per_million
## 1                          0                                     0
## 2                          0                                     0
## 3                          0                                     0
## 4                          0                                     0
## 5                          0                                     0
## 6                          0                                     0
##   new_people_vaccinated_smoothed new_people_vaccinated_smoothed_per_hundred
## 1                              0                                           0
## 2                              0                                           0
## 3                              0                                           0
## 4                              0                                           0
## 5                              0                                           0
## 6                              0                                           0
##   stringency_index population population_density median_age aged_65_older
## 1             8.33   39835428             54.422       18.6         2.581
## 2             8.33   39835428             54.422       18.6         2.581
## 3             8.33   39835428             54.422       18.6         2.581
## 4             8.33   39835428             54.422       18.6         2.581
## 5             8.33   39835428             54.422       18.6         2.581
## 6             8.33   39835428             54.422       18.6         2.581
##   aged_70_older gdp_per_capita extreme_poverty cardiovasc_death_rate
## 1         1.337       1803.987              0               597.029
## 2         1.337       1803.987              0               597.029
## 3         1.337       1803.987              0               597.029
## 4         1.337       1803.987              0               597.029
## 5         1.337       1803.987              0               597.029
## 6         1.337       1803.987              0               597.029
##   diabetes_prevalence female_smokers male_smokers handwashing_facilities
## 1                9.59              0            0                 37.746
## 2                9.59              0            0                 37.746
## 3                9.59              0            0                 37.746
## 4                9.59              0            0                 37.746
## 5                9.59              0            0                 37.746
## 6                9.59              0            0                 37.746
##   hospital_beds_per_thousand life_expectancy human_development_index
## 1                        0.5           64.83                   0.511
## 2                        0.5           64.83                   0.511
## 3                        0.5           64.83                   0.511
```

```
## 4                        0.5          64.83                   0.511
## 5                        0.5          64.83                   0.511
## 6                        0.5          64.83                   0.511
##   excess_mortality_cumulative_absolute excess_mortality_cumulative
## 1                                    0                           0
## 2                                    0                           0
## 3                                    0                           0
## 4                                    0                           0
## 5                                    0                           0
## 6                                    0                           0
##   excess_mortality excess_mortality_cumulative_per_million
## 1                0                                       0
## 2                0                                       0
## 3                0                                       0
## 4                0                                       0
## 5                0                                       0
## 6                0                                       0
```

Need to check dates such that the range for none of the variables is

```
test_cleaned <- cleaned_covid %>%
  select(!is.character) %>%
  select(!c(excess_mortality_cumulative, excess_mortality, excess_mortality_cumulative_per_million, exc
  filter(date > as.Date("2020-12-28")) %>%
  group_by(date) %>%
  summarize_all(sd) %>%
  filter_all(all_vars(. != 0))
```

```
## Warning: Predicate functions must be wrapped in `where()`.
##
##   # Bad
##   data %>% select(is.character)
##
##   # Good
##   data %>% select(where(is.character))
##
## i Please update your code.
## This message is displayed once per session.
```

```
head(test_cleaned)
```

```
## # A tibble: 6 x 52
##   date       total_cases new_cases new_cases_smoothed total_deaths new_deaths
##   <date>           <dbl>     <dbl>              <dbl>        <dbl>      <dbl>
## 1 2020-12-29    1735033.    16076.             14106.       33824.       310.
## 2 2020-12-30    1751155.    17487.             14119.       34124.       332.
## 3 2020-12-31    1769364.    19890.             14661.       34382.       281.
## 4 2021-01-01    1781283.    13226.             15261.       34551.       189.
## 5 2021-01-02    1800458.    21945.             16048.       34726.       201.
## 6 2021-01-03    1814231.    15490.             16563.       34840.       129.
## # ... with 46 more variables: 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>,
## #   total_tests_per_thousand <dbl>, new_tests_per_thousand <dbl>, ...
```

Above, we can see the bad variables that we need to exclude for our analysis. We save this cleaner dataset:

```
cleaner_covid <- cleaned_covid %>%
  select(!c(excess_mortality_cumulative, excess_mortality, excess_mortality_cumulative_per_million, exc

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

Note that 2020-12-28 may be too late a date. We may want to view the data even before this.

```
head(cleaner_covid)
```

```
##   iso_code continent    location       date total_cases new_cases
## 1      AFG      Asia Afghanistan 2020-02-24           5         5
## 2      AFG      Asia Afghanistan 2020-02-25           5         0
## 3      AFG      Asia Afghanistan 2020-02-26           5         0
## 4      AFG      Asia Afghanistan 2020-02-27           5         0
## 5      AFG      Asia Afghanistan 2020-02-28           5         0
## 6      AFG      Asia Afghanistan 2020-02-29           5         0
##   new_cases_smoothed total_deaths new_deaths new_deaths_smoothed
## 1              0.000            0          0                   0
## 2              0.000            0          0                   0
## 3              0.000            0          0                   0
## 4              0.000            0          0                   0
## 5              0.000            0          0                   0
## 6              0.714            0          0                   0
##   total_cases_per_million new_cases_per_million new_cases_smoothed_per_million
## 1                   0.126                 0.126                           0.000
## 2                   0.126                 0.000                           0.000
## 3                   0.126                 0.000                           0.000
## 4                   0.126                 0.000                           0.000
## 5                   0.126                 0.000                           0.000
## 6                   0.126                 0.000                           0.018
##   total_deaths_per_million new_deaths_per_million
## 1                        0                     0
## 2                        0                     0
## 3                        0                     0
## 4                        0                     0
## 5                        0                     0
## 6                        0                     0
##   new_deaths_smoothed_per_million icu_patients icu_patients_per_million
## 1                               0           0                        0
## 2                               0           0                        0
## 3                               0           0                        0
## 4                               0           0                        0
## 5                               0           0                        0
## 6                               0           0                        0
##   hosp_patients hosp_patients_per_million new_tests total_tests
```

```
## 1             0                        0         0           0
## 2             0                        0         0           0
## 3             0                        0         0           0
## 4             0                        0         0           0
## 5             0                        0         0           0
## 6             0                        0         0           0
##   total_tests_per_thousand new_tests_per_thousand new_tests_smoothed
## 1                        0                       0                  0
## 2                        0                       0                  0
## 3                        0                       0                  0
## 4                        0                       0                  0
## 5                        0                       0                  0
## 6                        0                       0                  0
##   new_tests_smoothed_per_thousand positive_rate tests_per_case
## 1                               0             0              0
## 2                               0             0              0
## 3                               0             0              0
## 4                               0             0              0
## 5                               0             0              0
## 6                               0             0              0
##   total_vaccinations people_vaccinated people_fully_vaccinated new_vaccinations
## 1                  0                 0                       0                0
## 2                  0                 0                       0                0
## 3                  0                 0                       0                0
## 4                  0                 0                       0                0
## 5                  0                 0                       0                0
## 6                  0                 0                       0                0
##   new_vaccinations_smoothed total_vaccinations_per_hundred
## 1                         0                              0
## 2                         0                              0
## 3                         0                              0
## 4                         0                              0
## 5                         0                              0
## 6                         0                              0
##   people_vaccinated_per_hundred people_fully_vaccinated_per_hundred
## 1                             0                                   0
## 2                             0                                   0
## 3                             0                                   0
## 4                             0                                   0
## 5                             0                                   0
## 6                             0                                   0
##   new_vaccinations_smoothed_per_million new_people_vaccinated_smoothed
## 1                                     0                              0
## 2                                     0                              0
## 3                                     0                              0
## 4                                     0                              0
## 5                                     0                              0
## 6                                     0                              0
##   new_people_vaccinated_smoothed_per_hundred stringency_index population
## 1                                          0            8.33   39835428
## 2                                          0            8.33   39835428
## 3                                          0            8.33   39835428
## 4                                          0            8.33   39835428
## 5                                          0            8.33   39835428
```

```
## 6                                        0             8.33   39835428
##   population_density median_age aged_65_older aged_70_older gdp_per_capita
## 1             54.422       18.6         2.581         1.337       1803.987
## 2             54.422       18.6         2.581         1.337       1803.987
## 3             54.422       18.6         2.581         1.337       1803.987
## 4             54.422       18.6         2.581         1.337       1803.987
## 5             54.422       18.6         2.581         1.337       1803.987
## 6             54.422       18.6         2.581         1.337       1803.987
##   extreme_poverty cardiovasc_death_rate diabetes_prevalence female_smokers
## 1               0               597.029                9.59              0
## 2               0               597.029                9.59              0
## 3               0               597.029                9.59              0
## 4               0               597.029                9.59              0
## 5               0               597.029                9.59              0
## 6               0               597.029                9.59              0
##   male_smokers handwashing_facilities hospital_beds_per_thousand
## 1            0                37.746                         0.5
## 2            0                37.746                         0.5
## 3            0                37.746                         0.5
## 4            0                37.746                         0.5
## 5            0                37.746                         0.5
## 6            0                37.746                         0.5
##   life_expectancy human_development_index
## 1           64.83                   0.511
## 2           64.83                   0.511
## 3           64.83                   0.511
## 4           64.83                   0.511
## 5           64.83                   0.511
## 6           64.83                   0.511
```

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

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

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

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion

## # A tibble: 1 x 55
##   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 48 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>, ...
```

(Yes, I know this is messy, inefficient code. It only gets worse from here)

```
# function to perform PCA and cluster based
get_cluster_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)

  return(loc_clusters)
}
```

```
set.seed(4700)
# the first date
all_loc_clusters <- get_cluster_df(first_day_df)
head(all_loc_clusters)
```

```
## # A tibble: 6 x 10
##   iso_code continent     location  date          PC1    PC2    PC3    PC4    PC5
##   <fct>    <fct>         <fct>     <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AFG      Asia          Afghanis~ 2020-12-28 -2.00  -2.81  -0.189 -0.480 -0.268
## 2 ALB      Europe        Albania   2020-12-28 -0.198  1.44   0.811  0.357 -1.27
## 3 DZA      Africa        Algeria   2020-12-28 -1.25  -1.05   0.261  1.09  -0.734
## 4 AND      Europe        Andorra   2020-12-28  0.188  1.85  -0.357 -3.60   0.785
## 5 AGO      Africa        Angola    2020-12-28 -1.82  -2.38  -0.318 -0.632  0.444
## 6 ATG      North America Antigua ~ 2020-12-28 -1.30  -0.958 -0.121  1.18   0.696
## # ... with 1 more variable: .cluster <fct>
```

CHECKPOINT

```
set.seed(4747)
```

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

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

lets gooooooooo

```
head(all_loc_clusters)
```

```
## # A tibble: 6 x 10
##   iso_code continent     location  date          PC1    PC2    PC3    PC4    PC5
##   <fct>    <fct>         <fct>     <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AFG      Asia          Afghanis~ 2020-12-28 -2.00  -2.81  -0.189 -0.480 -0.268
## 2 ALB      Europe        Albania   2020-12-28 -0.198  1.44   0.811  0.357 -1.27
## 3 DZA      Africa        Algeria   2020-12-28 -1.25  -1.05   0.261  1.09  -0.734
## 4 AND      Europe        Andorra   2020-12-28  0.188  1.85  -0.357 -3.60   0.785
## 5 AGO      Africa        Angola    2020-12-28 -1.82  -2.38  -0.318 -0.632  0.444
## 6 ATG      North America Antigua ~ 2020-12-28 -1.30  -0.958 -0.121  1.18   0.696
## # ... with 1 more variable: .cluster <fct>
```

```
all_long_lat_clusters <- left_join(all_loc_clusters, lats_long, by="location")
head(all_long_lat_clusters)
```

```
## # A tibble: 6 x 13
##   iso_code continent     location  date          PC1    PC2    PC3    PC4    PC5
##   <fct>    <fct>         <chr>     <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AFG      Asia          Afghanis~ 2020-12-28 -2.00  -2.81  -0.189 -0.480 -0.268
## 2 ALB      Europe        Albania   2020-12-28 -0.198  1.44   0.811  0.357 -1.27
## 3 DZA      Africa        Algeria   2020-12-28 -1.25  -1.05   0.261  1.09  -0.734
## 4 AND      Europe        Andorra   2020-12-28  0.188  1.85  -0.357 -3.60   0.785
## 5 AGO      Africa        Angola    2020-12-28 -1.82  -2.38  -0.318 -0.632  0.444
## 6 ATG      North America Antigua ~ 2020-12-28 -1.30  -0.958 -0.121  1.18   0.696
## # ... with 4 more variables: .cluster <fct>, ISO.3166.Country.Code <chr>,
## #   Latitude <dbl>, Longitude <dbl>
```

```
library(gapminder)
library(gganimate)

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=.cluster)) +
  transition_time(date)

animate(map_anim, duration = 20)
```

```
anim_save("/Users/jaredmejia/Documents/Pomona/fall21/compstats/GroupJ-COVID/workbooks/pca-cluster-map.g
```

TODO: Okay, clearly we need to do something to make sure the colors are consistent throughout. One idea: Identify certain countries that are always in the "same" cluster, such as the United States, Central African Republic, and Germany. Then, arbitrarily swap the names of the columns such that the cluster that contains the US is always labeled PC1, the cluster that contains the Central African Republic is always labeled PC2, and the cluster that contains Germany is always labeled PC3.

Also, consider changing cluster count to 5 from 6.

```
test_day_cluster <- get_cluster_df(first_day_df)
head(test_day_cluster)
```

```
## # A tibble: 6 x 10
##   iso_code continent     location  date          PC1    PC2    PC3    PC4    PC5
##   <fct>    <fct>         <fct>     <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AFG      Asia          Afghanis~ 2020-12-28 -2.00  -2.81  -0.189 -0.480 -0.268
## 2 ALB      Europe        Albania   2020-12-28 -0.198  1.44   0.811  0.357 -1.27
## 3 DZA      Africa        Algeria   2020-12-28 -1.25  -1.05   0.261  1.09  -0.734
## 4 AND      Europe        Andorra   2020-12-28  0.188  1.85  -0.357 -3.60   0.785
## 5 AGO      Africa        Angola    2020-12-28 -1.82  -2.38  -0.318 -0.632  0.444
## 6 ATG      North America Antigua ~ 2020-12-28 -1.30  -0.958 -0.121  1.18   0.696
## # ... with 1 more variable: .cluster <fct>
```

```
test_long_lat_clusters <- left_join(test_day_cluster, lats_long, by="location")
head(test_long_lat_clusters)
```

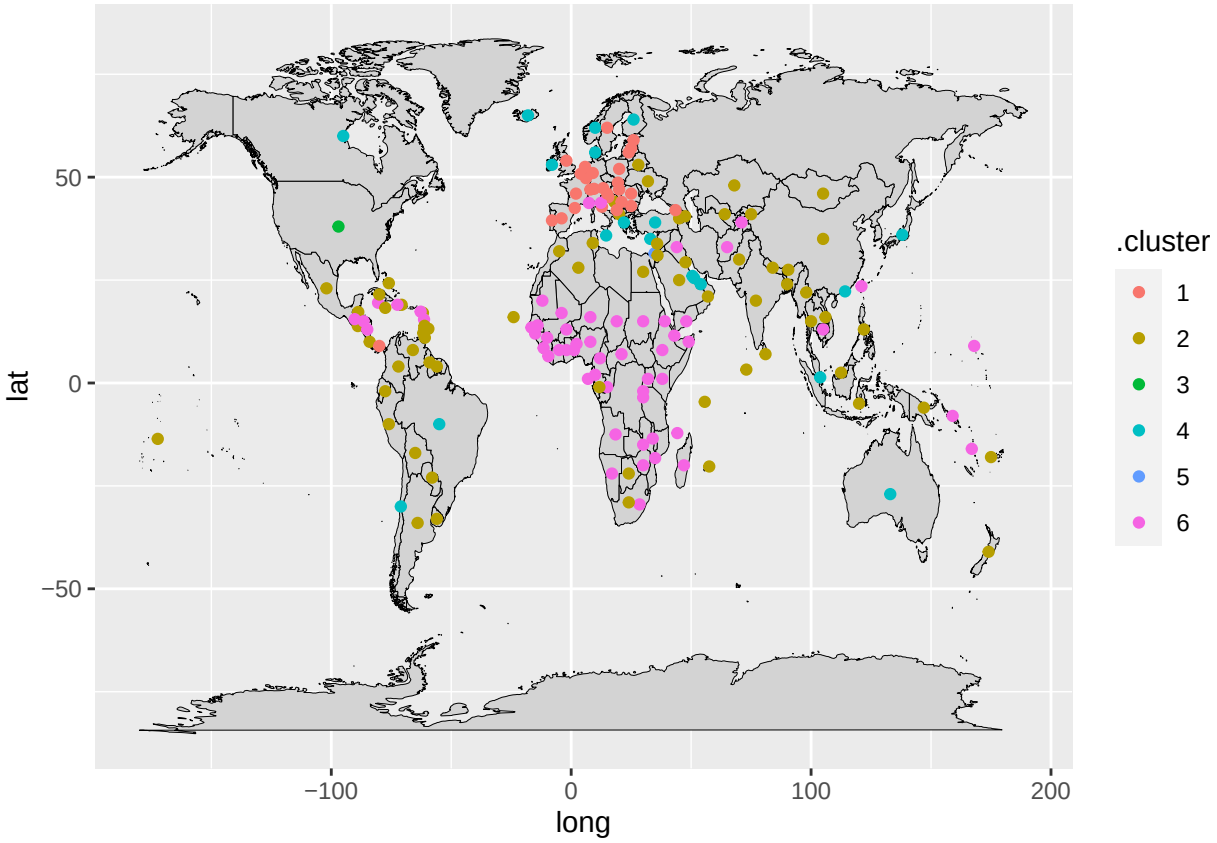
```
## # A tibble: 6 x 13
##   iso_code continent     location  date          PC1    PC2    PC3    PC4    PC5
##   <fct>    <fct>         <chr>     <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AFG      Asia          Afghanis~ 2020-12-28 -2.00  -2.81  -0.189 -0.480 -0.268
## 2 ALB      Europe        Albania   2020-12-28 -0.198  1.44   0.811  0.357 -1.27
## 3 DZA      Africa        Algeria   2020-12-28 -1.25  -1.05   0.261  1.09  -0.734
## 4 AND      Europe        Andorra   2020-12-28  0.188  1.85  -0.357 -3.60   0.785
## 5 AGO      Africa        Angola    2020-12-28 -1.82  -2.38  -0.318 -0.632  0.444
## 6 ATG      North America Antigua ~ 2020-12-28 -1.30  -0.958 -0.121  1.18   0.696
## # ... with 4 more variables: .cluster <fct>, ISO.3166.Country.Code <chr>,
## #   Latitude <dbl>, Longitude <dbl>
```

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

map1 <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "black", fill = "lightgray", size = 0.01
  ) +
  geom_point(data = test_long_lat_clusters, aes(x=Longitude, y=Latitude, color=.cluster))
```

```
## Warning: Ignoring unknown aesthetics: x, y
```

```
map1
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```



We can create a new column that sets the color of all clusters with the US as red, sets all clusters with CAR as blue, sets all clusters with Germany as green, sets the three other clusters as purple and pink and yellow.

```
a <- test_day_cluster %>%
  filter(location == "United States")

a
```

```
## # A tibble: 1 x 10
##   iso_code continent location date         PC1   PC2   PC3    PC4   PC5 .cluster
##   <fct>    <fct>     <fct>    <date>     <dbl> <dbl> <dbl>  <dbl> <dbl> <fct>
## 1 USA      North Am~ United ~ 2020-12-28  49.6 -15.4 0.755 -0.760 0.620 3
```

```
us_clust = a$.cluster
us_clust
```

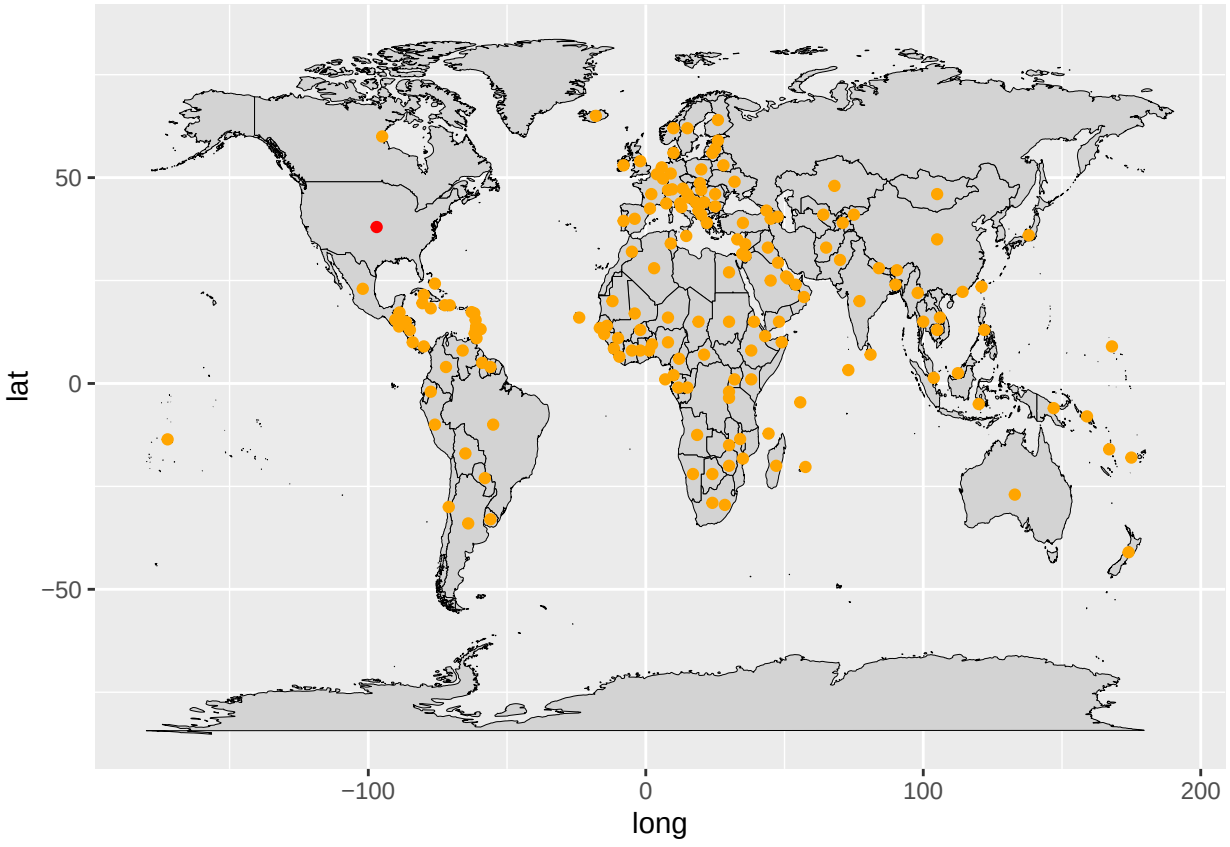
```
## [1] 3
## Levels: 1 2 3 4 5 6
```

```
test_long_lat_clusters <- test_long_lat_clusters %>%
  mutate(color = ifelse(.cluster == us_clust, "red", "orange1"))

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

```
## Warning: Ignoring unknown aesthetics: x, y
```

```
map
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```



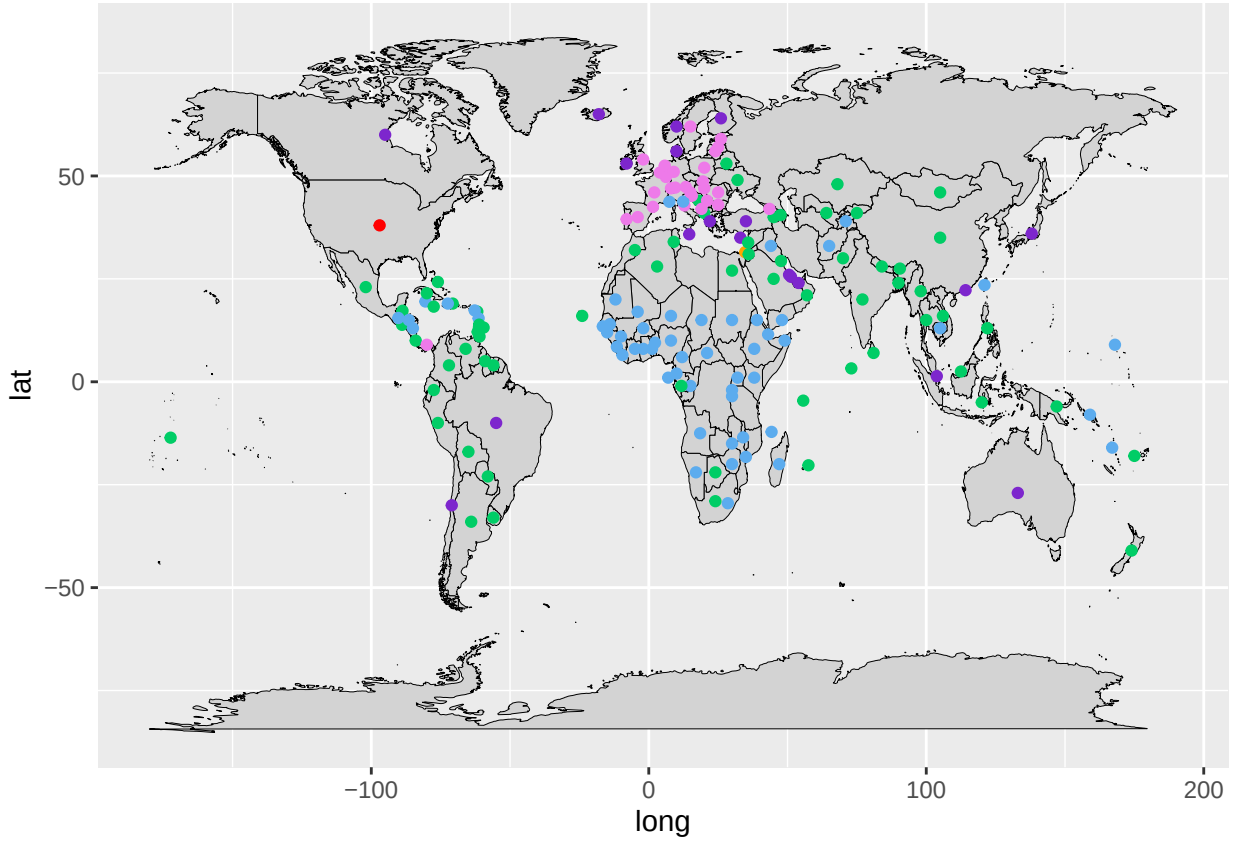

```
us_row <- test_day_cluster %>%
  dplyr::filter(location == "United States")
us_clust = us_row$.cluster

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

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

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

test_long_lat_clusters <- test_long_lat_clusters %>%
  mutate(color = ifelse(.cluster == us_clust, "red", "orange1")) %>%
  mutate(color = ifelse(.cluster == car_clust, "steelblue2", color)) %>%
  mutate(color =  ifelse(.cluster == ger_clust, "orchid2", color)) %>%
  mutate(color = ifelse(.cluster == remaining_clusts[1], "springgreen3", color)) %>%
  mutate(color = ifelse(.cluster == remaining_clusts[2], "purple3", color))

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

```
## Warning: Ignoring unknown aesthetics: x, y
```

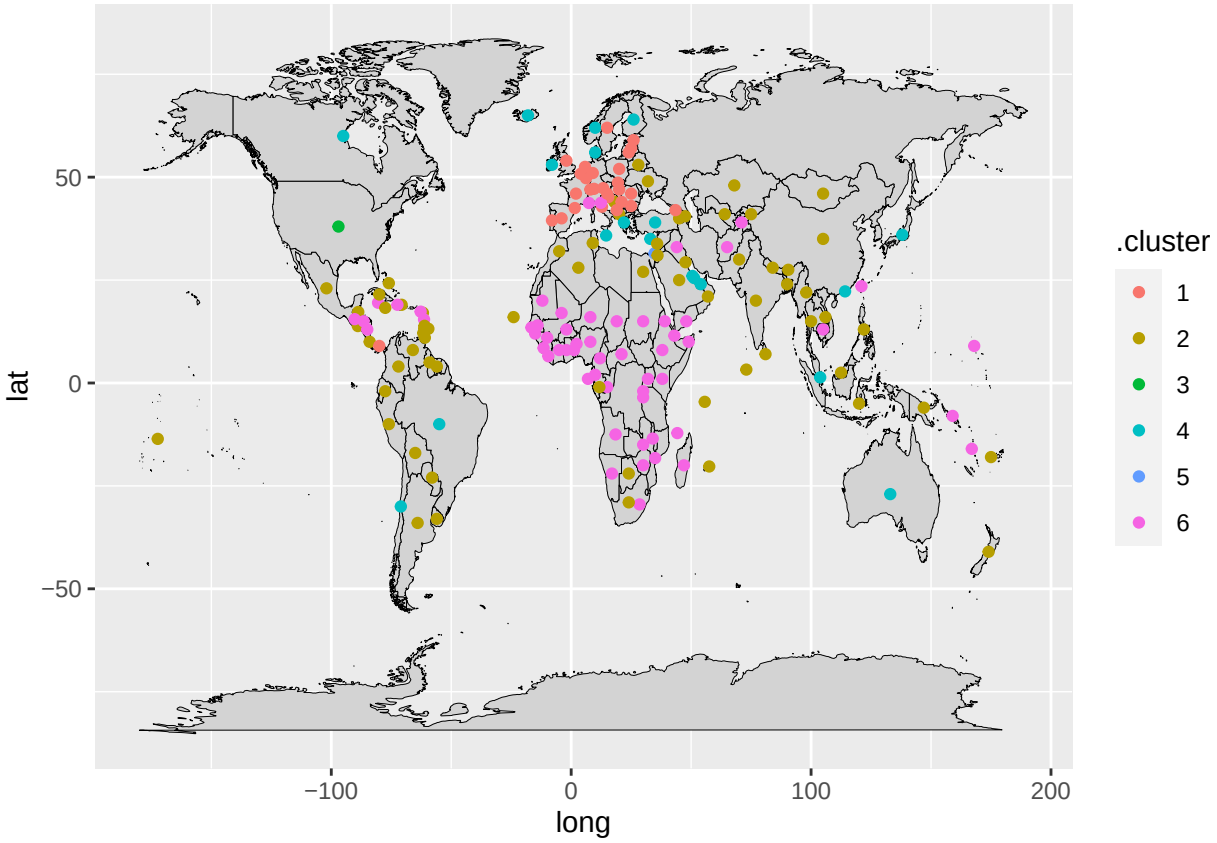
```
map
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```



```
map1
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```




```
test_day_cluster %>%
  select(iso_code) %>%
  dplyr::filter(iso_code == "ALB")
```

```
## # A tibble: 1 x 1
##   iso_code
##   <fct>
## 1 ALB
```

New function with color setting

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

  return(loc_clusters)
}
```

```
set.seed(4700)
# the first date
all_loc_clusters_colored <- get_cluster_colored_df(first_day_df)
head(all_loc_clusters_colored)
```

```
## # A tibble: 6 x 11
##   iso_code continent     location  date          PC1    PC2    PC3    PC4    PC5
##   <fct>    <fct>         <fct>     <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AFG      Asia          Afghanis~ 2020-12-28 -2.00  -2.81  -0.189 -0.480 -0.268
## 2 ALB      Europe        Albania   2020-12-28 -0.198  1.44   0.811  0.357 -1.27
## 3 DZA      Africa        Algeria   2020-12-28 -1.25  -1.05   0.261  1.09  -0.734
## 4 AND      Europe        Andorra   2020-12-28  0.188  1.85  -0.357 -3.60   0.785
## 5 AGO      Africa        Angola    2020-12-28 -1.82  -2.38  -0.318 -0.632  0.444
## 6 ATG      North America Antigua ~ 2020-12-28 -1.30  -0.958 -0.121  1.18   0.696
## # ... with 2 more variables: .cluster <fct>, color <chr>
```

CHECKPOINT

```
set.seed(4747)

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

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

```
all_loc_clusters_colored %>%
  dplyr::filter(color != "red", color != "steelblue2", continent=="Europe") %>%
  dplyr::count(location)
```

```
## # A tibble: 51 x 2
##    location                   n
##    <fct>                  <int>
##  1 Albania                  335
##  2 Andorra                  308
##  3 Austria                  347
##  4 Belarus                  337
##  5 Belgium                  347
##  6 Bosnia and Herzegovina   343
##  7 Bulgaria                 347
##  8 Croatia                  347
##  9 Cyprus                   345
## 10 Czechia                  347
## # ... with 41 more rows
```

```
all_long_lat_clusters_colored <- left_join(all_loc_clusters_colored, lats_long, by="location")
head(all_long_lat_clusters_colored)
```

```
## # A tibble: 6 x 14
##   iso_code continent     location  date          PC1    PC2    PC3    PC4    PC5
##   <fct>    <fct>         <chr>     <date>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AFG      Asia          Afghanis~ 2020-12-28 -2.00  -2.81  -0.189 -0.480 -0.268
## 2 ALB      Europe        Albania   2020-12-28 -0.198  1.44   0.811  0.357 -1.27
## 3 DZA      Africa        Algeria   2020-12-28 -1.25  -1.05   0.261  1.09  -0.734
## 4 AND      Europe        Andorra   2020-12-28  0.188  1.85  -0.357 -3.60   0.785
## 5 AGO      Africa        Angola    2020-12-28 -1.82  -2.38  -0.318 -0.632  0.444
## 6 ATG      North America Antigua ~ 2020-12-28 -1.30  -0.958 -0.121  1.18   0.696
## # ... with 5 more variables: .cluster <fct>, color <chr>,
## #   ISO.3166.Country.Code <chr>, Latitude <dbl>, Longitude <dbl>
```

```
map_anim_colored <- 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_colored, aes(x=Longitude, y=Latitude), color=all_long_lat_clus
  transition_time(date) +
  labs(title = 'PCA Clusters on Date: {frame_time}', x = '', y = '')

animate(map_anim_colored, duration = 20)
```

```
anim_save("/Users/jaredmejia/Documents/Pomona/fall21/compstats/GroupJ-COVID/workbooks/pca-cluster-map-c
```

Now, we shall repeat this process, but getting rid of all the variables relating to population size

```
cleaner_covid_2 <- cleaned_covid %>%
  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"))
```

```
head(cleaner_covid_2)
```

```
##   total_cases_per_million new_cases_per_million new_cases_smoothed_per_million
## 1                   0.126                 0.126                           0.000
## 2                   0.126                 0.000                           0.000
## 3                   0.126                 0.000                           0.000
## 4                   0.126                 0.000                           0.000
## 5                   0.126                 0.000                           0.000
## 6                   0.126                 0.000                           0.018
##   total_deaths_per_million new_deaths_per_million
## 1                        0                      0
## 2                        0                      0
## 3                        0                      0
## 4                        0                      0
## 5                        0                      0
## 6                        0                      0
##   new_deaths_smoothed_per_million icu_patients_per_million
## 1                               0                        0
## 2                               0                        0
## 3                               0                        0
## 4                               0                        0
## 5                               0                        0
## 6                               0                        0
##   hosp_patients_per_million total_tests_per_thousand new_tests_per_thousand
## 1                         0                        0                      0
## 2                         0                        0                      0
## 3                         0                        0                      0
## 4                         0                        0                      0
## 5                         0                        0                      0
## 6                         0                        0                      0
##   new_tests_smoothed_per_thousand tests_per_case total_vaccinations_per_hundred
## 1                               0              0                             0
```

```
## 2                              0             0                              0
## 3                              0             0                              0
## 4                              0             0                              0
## 5                              0             0                              0
## 6                              0             0                              0
##   people_vaccinated_per_hundred people_fully_vaccinated_per_hundred
## 1                             0                                    0
## 2                             0                                    0
## 3                             0                                    0
## 4                             0                                    0
## 5                             0                                    0
## 6                             0                                    0
##   new_vaccinations_smoothed_per_million
## 1                                     0
## 2                                     0
## 3                                     0
## 4                                     0
## 5                                     0
## 6                                     0
##   new_people_vaccinated_smoothed_per_hundred gdp_per_capita
## 1                                          0       1803.987
## 2                                          0       1803.987
## 3                                          0       1803.987
## 4                                          0       1803.987
## 5                                          0       1803.987
## 6                                          0       1803.987
##   hospital_beds_per_thousand continent    location       date aged_65_older
## 1                        0.5      Asia Afghanistan 2020-02-24         2.581
## 2                        0.5      Asia Afghanistan 2020-02-25         2.581
## 3                        0.5      Asia Afghanistan 2020-02-26         2.581
## 4                        0.5      Asia Afghanistan 2020-02-27         2.581
## 5                        0.5      Asia Afghanistan 2020-02-28         2.581
## 6                        0.5      Asia Afghanistan 2020-02-29         2.581
##   aged_70_older extreme_poverty cardiovasc_death_rate diabetes_prevalence
## 1         1.337               0               597.029                9.59
## 2         1.337               0               597.029                9.59
## 3         1.337               0               597.029                9.59
## 4         1.337               0               597.029                9.59
## 5         1.337               0               597.029                9.59
## 6         1.337               0               597.029                9.59
##   female_smokers male_smokers handwashing_facilities life_expectancy
## 1              0            0                 37.746           64.83
## 2              0            0                 37.746           64.83
## 3              0            0                 37.746           64.83
## 4              0            0                 37.746           64.83
## 5              0            0                 37.746           64.83
## 6              0            0                 37.746           64.83
##   human_development_index
## 1                   0.511
## 2                   0.511
## 3                   0.511
## 4                   0.511
## 5                   0.511
## 6                   0.511
```

CHECKPOINT

```
# function to perform PCA and cluster based and colors!
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(color = ifelse(.cluster == remaining_clusts[1], "red", "orange1")) %>%
    mutate(color = ifelse(.cluster == uae_clust, "purple3", color)) %>%
        mutate(color = ifelse(.cluster == tunisia_clust, "springgreen3", color)) %>%
        mutate(color =  ifelse(.cluster == ger_clust, "orchid2", color)) %>%
        mutate(color = ifelse(.cluster == car_clust, "steelblue2", color))

  return(loc_clusters)
}
```

```
set.seed(4700)
# the first date
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)
head(all_loc_clusters_colored2)
```

```
## # A tibble: 6 x 10
##   continent  location  date          PC1     PC2    PC3     PC4     PC5 .cluster
##   <fct>      <fct>     <date>      <dbl>   <dbl>  <dbl>   <dbl>   <dbl> <fct>
## 1 Asia       Afghanis~ 2020-12-28 -2.98   0.269   0.458  0.340   0.653  3
## 2 Europe     Albania   2020-12-28  0.761 -0.788  -0.668  0.985   0.268  2
## 3 Africa     Algeria   2020-12-28 -1.63  -0.125  -0.918  0.645   0.683  4
## 4 Europe     Andorra   2020-12-28  2.60  -0.558   2.75  -0.0326  2.93   2
## 5 Africa     Angola    2020-12-28 -2.75   0.416   0.720 -0.602  -0.0812 3
## 6 North Ame~ Antigua ~ 2020-12-28 -1.17   0.0241 -1.29  -0.363   0.987  4
## # ... with 1 more variable: color <chr>
```

```
set.seed(4747)

# 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_loc_clusters_colored2 %>%
  dplyr::filter(color!="red", color != "orchid2", color != "steelblue2", color != "purple3") %>%
  dplyr::count(location) %>%
  arrange(desc(n))
```

```
## # A tibble: 217 x 2
##    location               n
##    <fct>              <int>
##  1 Tunisia              343
##  2 Kazakhstan           303
##  3 Paraguay             294
##  4 Fiji                 292
##  5 Suriname             290
##  6 Trinidad and Tobago  290
##  7 Bahamas              289
##  8 Iran                 288
##  9 Costa Rica           286
## 10 Azerbaijan           284
## # ... with 207 more rows
```

```
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  3
## 2 Europe     Albania   2020-12-28  0.761 -0.788  -0.668  0.985   0.268  2
## 3 Africa     Algeria   2020-12-28 -1.63  -0.125  -0.918  0.645   0.683  4
## 4 Europe     Andorra   2020-12-28  2.60  -0.558   2.75  -0.0326  2.93   2
```

```
## 5 Africa     Angola    2020-12-28 -2.75   0.416   0.720 -0.602  -0.0812 3
## 6 North Ame~ Antigua ~ 2020-12-28 -1.17   0.0241 -1.29  -0.363   0.987  4
## # ... with 4 more variables: color <chr>, ISO.3166.Country.Code <chr>,
## #   Latitude <dbl>, Longitude <dbl>
```

```
library(gganimate)
library(gapminder)

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=all_long_lat_clu
  transition_time(date) +
  labs(title = 'PCA Clusters on Date: {frame_time}', x = '', y = '')

animate(map_anim_colored2, duration = 20)
```

```
anim_save("/Users/jaredmejia/Documents/Pomona/fall21/compstats/GroupJ-COVID/workbooks/pca-cluster-map-c
```