The Global Influence of COVID-19 on Happiness

1. Introduction

COVID-19 is an illness caused by a virus that has taken a toll in everyone's families. Not only that, but one's mental health as well due to the numerous restrictions from losing in-person connections to even family members. Moreover, the question here is whether COVID-19 impacted global happiness levels. The datasets we chose are World Happiness Reports up to 2022 focusing on the years 2018, 2020, and 2021 by Mathurin Ache and Covid-19 Global Summary Dataset by Joseph Assaker, both sets are from Kaggle and can be found at the following links: https://www.kaggle.com/datasets/josephassaker/covid19-global-dataset?select=worldometer_coronavirus_daily_data.csv (https://www.kaggle.com/datasets/josephassaker/covid19-global-dataset?select=worldometer_coronavirus_daily_data.csv) https://www.kaggle.com/datasets/mathurinache/world-happiness-report?resource=download&select=2021.csv (https://www.kaggle.com/datasets/mathurinache/world-happiness-report?resource=download&select=2021.csv) Contained within these two data sets are more data sets that focus on different variables. The Covid-19 data set includes an overall summary of global Covid-19 statistics from the beginning of the pandemic to May of 2022, it also contains global daily Covid-19 statistics from February of 2020 to May of 2022. The Happiness data set includes separate data sets for each year, we chose to focus on the years 2018, 2020, and 2021. These years help to represent a pre-covid era, during the pandemic, and towards the back end of the pandemic. We chose these data sets because it would be interesting to see how the Covid-19 pandemic affected the overall happiness levels across the globe in addition to other factors examined by the happiness data set including: Trust in the government, and Perception of Freedom.

A unique row in the Covid Summary data set would represent a country (categorical), continent (categorical), total confirmed covid cases (numeric), total covid deaths (numeric), total recovered, total active cases (numerical), serious or critical cases (numeric), total cases per one million in the population (numeric), total deaths per one million in the population (numeric), total number of covid tests (numeric), total covid tests per one million in the population (numeric), and total population (numeric).

A unique row in the Covid Daily data set would represent the date (categorical), country (categorical), cumulative total cases (numeric), daily new cases (numeric), active cases (numeric), cumulative total deaths (numeric), and daily new deaths (numeric).

A unique row in the Happiness 2018 data set would represent Rank (categorical), Happiness Score (numeric), Country (categorical), GDP per capital (numeric), Healthy Life Expectancy (numeric), Social support (numeric), Freedom to make life choices (numeric), Generosity (numeric), Corruption Perception (numeric).

A unique row in the Happiness 2020 data set would represent Happiness Score (numeric), Standard error of Happiness Score (numeric), Upper whisker (numeric), lower whisker (numeric), Country (categorical), Region (categorical), GDP per capital (numeric), Healthy Life Expectancy (numeric), Social support (numeric), Freedom to make life choices (numeric), Generosity (numeric), Corruption Perception (numeric), Dystopia (numeric), then Explained by: GDP per capital (numeric), Healthy Life Expectancy (numeric), Social support (numeric), Freedom to make life choices (numeric), Generosity (numeric), Corruption Perception (numeric), Dystopia (numeric), and finally Dystopia + residual (numeric).

A unique row in the Happiness 2020 data set would represent Happiness Score (numeric), Standard error of Happiness Score (numeric), Upper whisker (numeric), lower whisker (numeric), Country (categorical), Region (categorical), GDP per capital (numeric), Healthy Life Expectancy (numeric), Social support (numeric), Freedom to make life choices (numeric), Generosity (numeric), Corruption Perception (numeric), Dystopia (numeric), then Explained by: GDP per capital (numeric), Healthy Life Expectancy (numeric), Social support (numeric), Freedom to make life choices (numeric), Generosity (numeric), Corruption Perception (numeric), Dystopia (numeric), and finally Dystopia + residual (numeric).

All of these data sets can be joined by the Country (categorical) variable. The three happiness data sets can be joined by Happiness Score (numeric), Country (categorical), GDP per capital (numeric), Healthy Life Expectancy (numeric), Social support (numeric), Freedom to make life choices (numeric), Generosity (numeric), Corruption Perception (numeric). The happiness data sets can be joined to the Covid Daily data set by year (categorical).

The following data sets were cleaned and created in the previous project by standardizing all of the original data sets described above and joining them according to desired relationships/comparisons containing information about geography, COVID, and happiness. Cleaning and tidying the data involved changing many of the country names and other variables to match across datasets and ensuring that all of the variables are consistent and contained across all years.

Our research questions are: Can geographical location be predicted based on Covid-19 and Happiness data using PCA?

Can the “deadliness” of a country be predicted based on Covid-19 and Happiness data using cross-validation?

2. Exploratory Data Analysis

Cleaning to make a Correlation Matrix

```
#Remove variables
X2018_2020_2021_by_yearly_covid_num <- X2018_2020_2021_by_yearly_covid%>%
  select(-1,-2)
#Make Year numeric
X2018_2020_2021_by_yearly_covid_num$Year <- as.numeric(X2018_2020_2021_by_yearly_covid_num
$Year)
#Make Corruption numeric
X2018_2020_2021_by_yearly_covid_num$Corruption <- as.numeric(X2018_2020_2021_by_yearly_cov
id_num$Corruption)

#Check it
X2018_2020_2021_by_yearly_covid_num
```

```
## # A tibble: 423 × 10
##    Score   GDP Social_su…¹ Life_…² Freedom Gener…³ Corru…⁴  Year new_c…⁵ new_d…⁶
##    <dbl> <dbl>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
##  1  3.63 0.332       0.537   0.255   0.085   0.191   0.036  2018      NA      NA
##  2  4.59 0.916       0.817   0.79    0.419   0.149   0.032  2018      NA      NA
##  3  5.30 0.979       1.15    0.687   0.077   0.055   0.135  2018      NA      NA
##  4  6.39 1.07        1.47    0.744   0.57    0.062   0.054  2018      NA      NA
##  5  4.32 0.816       0.99    0.666   0.26    0.077   0.028  2018      NA      NA
##  6  7.27 1.34        1.57    0.91    0.647   0.361   0.302  2018      NA      NA
##  7  7.14 1.34        1.50    0.891   0.617   0.242   0.224  2018      NA      NA
##  8  5.20 1.02        1.16    0.603   0.43    0.031   0.176  2018      NA      NA
##  9  6.10 1.34        1.37    0.698   0.594   0.243   0.123  2018      NA      NA
## 10  4.5  0.532       0.85    0.579   0.58    0.153   0.144  2018      NA      NA
## # … with 413 more rows, and abbreviated variable names ¹Social_support,
## #   ²Life_expectancy, ³Generosity, ⁴Corruption, ⁵new_cases_per_year,
## #   ⁶new_deaths_per_year
```

```
#Remove variables
happiness_sumcovid_yearlycovid_num <- happiness_sumcovid_yearlycovid%>%
  select(total_cases_per_1m_population, total_deaths_per_1m_population, Score, GDP, Social
_support, Life_expectancy, Freedom, Generosity, Corruption, fatality_rate)%>%
  #Make numeric
  mutate(Corruption=as.numeric(Corruption))

happiness_sumcovid_yearlycovid_num
```

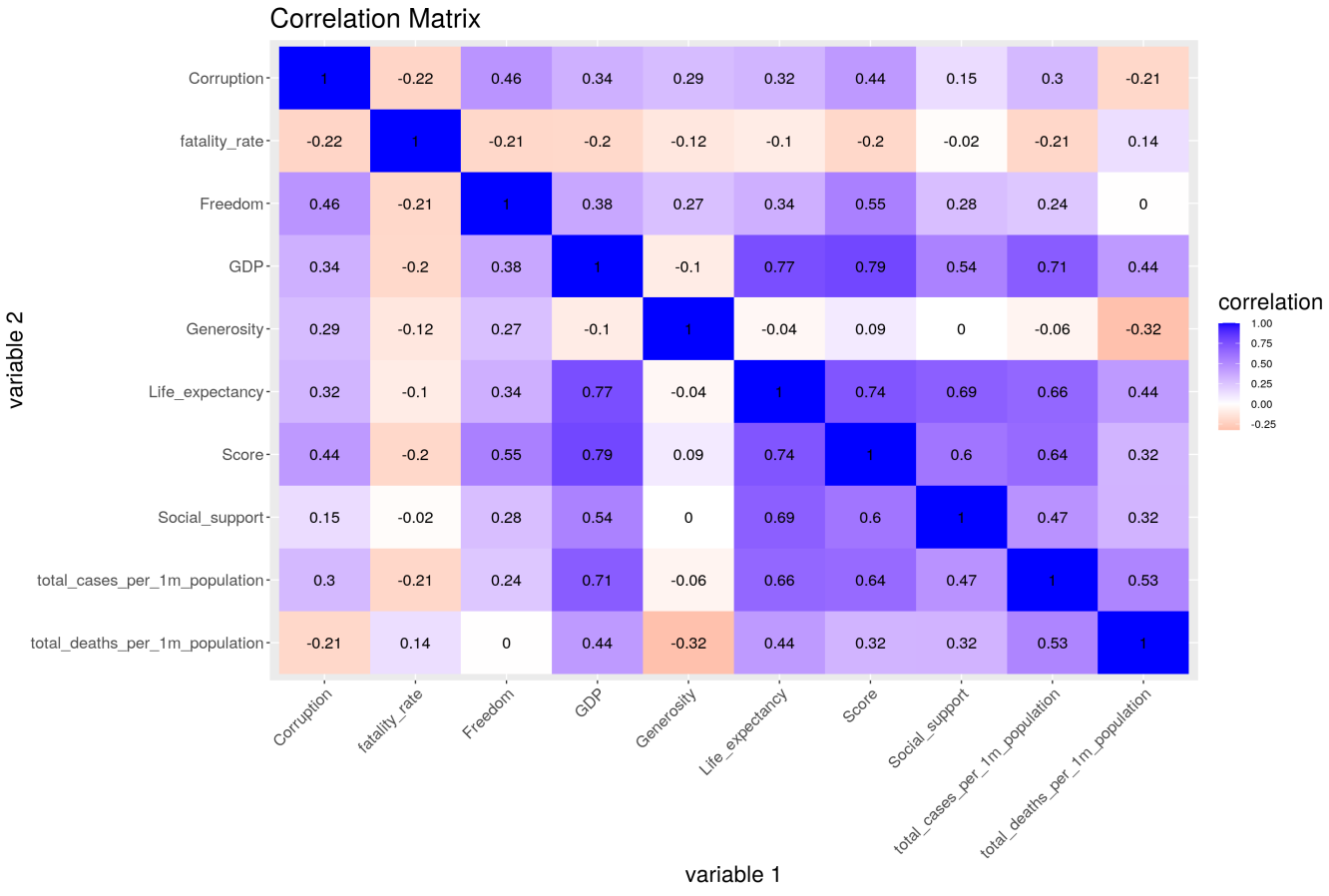
```
## # A tibble: 423 × 10
##    total_c…¹ total…² Score   GDP Socia…³ Life_…⁴ Freedom Gener…⁵ Corru…⁶ fatal…⁷
##        <dbl>   <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1      4420     190  3.63 0.332   0.537   0.255  0.085    0.191 0.036   NA
##  2      4420     190  2.57 0.301   0.356   0.266  0        0.135 0.00123  0.0419
##  3      4420     190  2.52 0.37    0       0.126  0        0.122 0.01     0.0488
##  4     95954    1218  4.59 0.916   0.817   0.79   0.419    0.149 0.032   NA
##  5     95954    1218  4.88 0.907   0.830   0.846  0.462    0.171 0.0254   0.0202
##  6     95954    1218  5.12 1.01    0.529   0.646  0.491    0.168 0.024    0.0134
##  7      5865     152  5.30 0.979   1.15    0.687  0.077    0.055 0.135   NA
##  8      5865     152  5.01 0.944   1.14    0.745  0.0839   0.119 0.129    0.0277
##  9      5865     152  4.89 0.946   0.765   0.552  0.119    0.144 0.12     0.0296
## 10    197992    2800  6.39 1.07    1.47    0.744  0.57     0.062 0.054   NA
## # … with 413 more rows, and abbreviated variable names
## #   ¹total_cases_per_1m_population, ²total_deaths_per_1m_population,
## #   ³Social_support, ⁴Life_expectancy, ⁵Generosity, ⁶Corruption, ⁷fatality_rate
```

Correlation Matrix

Create Correlation Matrix

The correlation matrix above represents the different correlation values that exist for each pair of variables. Some of the most positively correlated variables include a country's GDP in relation to its happiness score, its GDP in relation to its overall life expectancy, as well as, surprisingly, it's GDP in relation to its total amount of Covid-19 cases per 1 million people. Overall, it seems that the higher a country's GDP is, the higher the citizen's quality of life may be in terms of life expectancy and happiness. On the other hand, the most negatively correlated variables appear to be as follows: total deaths & generosity, corruption & Covid-19 fatality rate, sense of freedom & fatality rate, and amount of cases per 1 million people & fatality rate. Overall, it seems that the most negatively correlated pairs of variables often include the fatality rate.

```
# Find the correlations among all variables
cor(happiness_sumcovid_yearlycovid_num, use = "pairwise.complete.obs") %>%
  # Save as a data frame
  as.data.frame %>%
  # Convert row names to an explicit variable
  rownames_to_column %>%
  # Pivot so that all correlations appear in the same column
  pivot_longer(-1,
               names_to = "other_var",
               values_to = "correlation") %>%
  # Define ggplot (reorder values on y-axis)
  ggplot(aes(x = rowname,
             y = ordered(other_var, levels = rev(sort(unique(other_var)))),
             fill = correlation)) +
  # Heat map with geom_tile
  geom_tile() +
  # Change the scale to make the middle appear neutral
  scale_fill_gradient2(low = "red", mid = "white", high = "blue") +
  # Overlay values
  geom_text(aes(label = round(correlation,2)), color = "black", size = 4.5) +
  # Angle the x-axis label to 45 degrees
  theme(axis.text.x = element_text(angle = 45, hjust = 1,size=13),
        axis.text.y = element_text(size=13),
        title = element_text(size=18)) +
  # Give title and labels
  labs(title = "Correlation Matrix",
       x = "variable 1", y = "variable 2")
```



Visualizations ### One-Variable Graphs

Wrangling for the one-variable graphs

The map_data data set is used and joined to the Average Happiness by Country data set using anti_join to check which values are present in the Happiness data set and not present in the map data set. The spelling of these countries are checked to see if they can be corrected, and are corrected using mutate() to rename. The map data is then joined to the general happiness data and the average happiness per country data using left_join() to create two new data sets for visualization. Macedonia was left out because it was not present in the map data set.

```
mapWorld <- map_data("world") #Rename
mapWorld #Check
```

```
##         long      lat group order      region subregion
## 1  -69.89912 12.45200     1     1       Aruba      <NA>
## 2  -69.89571 12.42300     1     2       Aruba      <NA>
## 3  -69.94219 12.43853     1     3       Aruba      <NA>
## 4  -70.00415 12.50049     1     4       Aruba      <NA>
## 5  -70.06612 12.54697     1     5       Aruba      <NA>
## 6  -70.05088 12.59707     1     6       Aruba      <NA>
## 7  -70.03511 12.61411     1     7       Aruba      <NA>
## 8  -69.97314 12.56763     1     8       Aruba      <NA>
## 9  -69.91181 12.48047     1     9       Aruba      <NA>
## 10 -69.89912 12.45200     1    10       Aruba      <NA>
## 12  74.89131 37.23164     2    12 Afghanistan      <NA>
## 13  74.84023 37.22505     2    13 Afghanistan      <NA>
## 14  74.76738 37.24917     2    14 Afghanistan      <NA>
## 15  74.73896 37.28564     2    15 Afghanistan      <NA>
## 16  74.72666 37.29072     2    16 Afghanistan      <NA>
## 17  74.66895 37.26670     2    17 Afghanistan      <NA>
##  [ reached 'max' / getOption("max.print") -- omitted 99322 rows ]
```

```
mapWorld_long_lat <- mapWorld%>% #Rename
  select(1,2,5,3) #Select the columns of interest
mapWorld_long_lat #Check
```

```
##         long      lat      region group
## 1  -69.89912 12.45200       Aruba     1
## 2  -69.89571 12.42300       Aruba     1
## 3  -69.94219 12.43853       Aruba     1
## 4  -70.00415 12.50049       Aruba     1
## 5  -70.06612 12.54697       Aruba     1
## 6  -70.05088 12.59707       Aruba     1
## 7  -70.03511 12.61411       Aruba     1
## 8  -69.97314 12.56763       Aruba     1
## 9  -69.91181 12.48047       Aruba     1
## 10 -69.89912 12.45200       Aruba     1
## 12  74.89131 37.23164 Afghanistan     2
## 13  74.84023 37.22505 Afghanistan     2
## 14  74.76738 37.24917 Afghanistan     2
## 15  74.73896 37.28564 Afghanistan     2
## 16  74.72666 37.29072 Afghanistan     2
## 17  74.66895 37.26670 Afghanistan     2
## 18  74.55899 37.23662 Afghanistan     2
## 19  74.37217 37.15771 Afghanistan     2
## 20  74.37617 37.13735 Afghanistan     2
## 21  74.49796 37.05722 Afghanistan     2
## 22  74.52646 37.03066 Afghanistan     2
## 23  74.54140 37.02217 Afghanistan     2
## 24  74.43106 36.98369 Afghanistan     2
## 25  74.19473 36.89688 Afghanistan     2
## 26  74.03887 36.82573 Afghanistan     2
##  [ reached 'max' / getOption("max.print") -- omitted 99313 rows ]
```

```
anti_join(X2018_2020_2021_avg_happiness_by_country, mapWorld,
          by = c("Country" = "region")) #Anti join to check for discrepencies
```

```
## # A tibble: 6 × 2
##   Country                 mean_Score
##   <chr>                        <dbl>
## 1 Bosnia And Herzegovina        5.54
## 2 Congo                         5.03
## 3 Macedonia                     5.15
## 4 Palestinian Territories       4.60
## 5 United Kingdom                7.14
## 6 United States                 6.93
```

```
mapWorld_long_lat%>% #Checking for spelling
  distinct(region)%>% #Finds distinct regions
  arrange(region) #Arrange in alphabetical order
```

```
##                              region
## 1                       Afghanistan
## 2                           Albania
## 3                           Algeria
## 4                    American Samoa
## 5                           Andorra
## 6                            Angola
## 7                          Anguilla
## 8                        Antarctica
## 9                           Antigua
## 10                        Argentina
## 11                          Armenia
## 12                            Aruba
## 13                Ascension Island
## 14                       Australia
## 15                          Austria
## 16                       Azerbaijan
## 17                           Azores
## 18                          Bahamas
## 19                          Bahrain
## 20                       Bangladesh
## 21                         Barbados
## 22                          Barbuda
## 23                          Belarus
## 24                          Belgium
## 25                           Belize
## 26                            Benin
## 27                          Bermuda
## 28                           Bhutan
## 29                          Bolivia
## 30                          Bonaire
## 31            Bosnia and Herzegovina
## 32                         Botswana
## 33                           Brazil
## 34                           Brunei
## 35                         Bulgaria
## 36                     Burkina Faso
## 37                          Burundi
## 38                         Cambodia
## 39                         Cameroon
## 40                           Canada
## 41                   Canary Islands
## 42                       Cape Verde
## 43                   Cayman Islands
## 44           Central African Republic
## 45                             Chad
## 46                Chagos Archipelago
## 47                             Chile
## 48                             China
## 49                 Christmas Island
## 50                    Cocos Islands
## 51                          Colombia
```

```
## 52                            Comoros
## 53                       Cook Islands
## 54                         Costa Rica
## 55                            Croatia
## 56                               Cuba
## 57                            Curacao
## 58                             Cyprus
## 59                     Czech Republic
## 60     Democratic Republic of the Congo
## 61                            Denmark
## 62                           Djibouti
## 63                           Dominica
## 64                 Dominican Republic
## 65                            Ecuador
## 66                              Egypt
## 67                        El Salvador
## 68                  Equatorial Guinea
## 69                            Eritrea
## 70                            Estonia
## 71                           Ethiopia
## 72                   Falkland Islands
## 73                      Faroe Islands
## 74                               Fiji
## 75                            Finland
## 76                             France
## 77                      French Guiana
## 78                    French Polynesia
## 79  French Southern and Antarctic Lands
## 80                              Gabon
## 81                             Gambia
## 82                            Georgia
## 83                            Germany
## 84                              Ghana
## 85                             Greece
## 86                          Greenland
## 87                            Grenada
## 88                         Grenadines
## 89                         Guadeloupe
## 90                               Guam
## 91                          Guatemala
## 92                           Guernsey
## 93                             Guinea
## 94                      Guinea-Bissau
## 95                             Guyana
## 96                              Haiti
## 97                       Heard Island
## 98                           Honduras
## 99                            Hungary
## 100                           Iceland
##  [ reached 'max' / getOption("max.print") -- omitted 152 rows ]
```

```
mapWorld_long_lat_clean1 <- mapWorld_long_lat%>% #Rename data
mutate(region=recode(region, 'Bosnia and Herzegovina'='Bosnia And Herzegovina', 'Democrati
c Republic of the Congo'='Congo','Palestine'='Palestinian Territories','UK'='United Kingdo
m', 'USA'='United States')) #Rename values

mapWorld_long_lat_clean1 #Check
```

```
##         long      lat      region group
## 1  -69.89912 12.45200       Aruba     1
## 2  -69.89571 12.42300       Aruba     1
## 3  -69.94219 12.43853       Aruba     1
## 4  -70.00415 12.50049       Aruba     1
## 5  -70.06612 12.54697       Aruba     1
## 6  -70.05088 12.59707       Aruba     1
## 7  -70.03511 12.61411       Aruba     1
## 8  -69.97314 12.56763       Aruba     1
## 9  -69.91181 12.48047       Aruba     1
## 10 -69.89912 12.45200       Aruba     1
## 12  74.89131 37.23164 Afghanistan     2
## 13  74.84023 37.22505 Afghanistan     2
## 14  74.76738 37.24917 Afghanistan     2
## 15  74.73896 37.28564 Afghanistan     2
## 16  74.72666 37.29072 Afghanistan     2
## 17  74.66895 37.26670 Afghanistan     2
## 18  74.55899 37.23662 Afghanistan     2
## 19  74.37217 37.15771 Afghanistan     2
## 20  74.37617 37.13735 Afghanistan     2
## 21  74.49796 37.05722 Afghanistan     2
## 22  74.52646 37.03066 Afghanistan     2
## 23  74.54140 37.02217 Afghanistan     2
## 24  74.43106 36.98369 Afghanistan     2
## 25  74.19473 36.89688 Afghanistan     2
## 26  74.03887 36.82573 Afghanistan     2
##  [ reached 'max' / getOption("max.print") -- omitted 99313 rows ]
```

```
anti_join(X2018_2020_2021_avg_happiness_by_country,
          mapWorld_long_lat_clean1, by = c("Country" = "region")) #Check that it worked
```

```
## # A tibble: 1 × 2
##   Country   mean_Score
##   <chr>          <dbl>
## 1 Macedonia       5.15
```

```
X2018_2020_2021_map_avg_happy <- X2018_2020_2021_avg_happiness_by_country%>% #Rename
  left_join(mapWorld_long_lat_clean1, by = c("Country"="region")) #Join

X2018_2020_2021_map <- X2018_2020_2021%>% #Rename
  left_join(mapWorld_long_lat_clean1, by = c("Country"="region")) #Join

X2018_2020_2021_map #Check
```

```
## # A tibble: 249,033 × 13
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵  Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <dbl>
##  1 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  2 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  3 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  4 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  5 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  6 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  7 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  8 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
##  9 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
## 10 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036    2018
## # … with 249,023 more rows, 3 more variables: long <dbl>, lat <dbl>,
## #   group <dbl>, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
```

```
X2018_2020_2021_map_avg_happy #Check
```

```
## # A tibble: 83,011 × 5
##    Country     mean_Score  long   lat group
##    <chr>            <dbl> <dbl> <dbl> <dbl>
##  1 Afghanistan       2.91  74.9  37.2     2
##  2 Afghanistan       2.91  74.8  37.2     2
##  3 Afghanistan       2.91  74.8  37.2     2
##  4 Afghanistan       2.91  74.7  37.3     2
##  5 Afghanistan       2.91  74.7  37.3     2
##  6 Afghanistan       2.91  74.7  37.3     2
##  7 Afghanistan       2.91  74.6  37.2     2
##  8 Afghanistan       2.91  74.4  37.2     2
##  9 Afghanistan       2.91  74.4  37.1     2
## 10 Afghanistan       2.91  74.5  37.1     2
## # … with 83,001 more rows
```

Average Global Happiness Scores

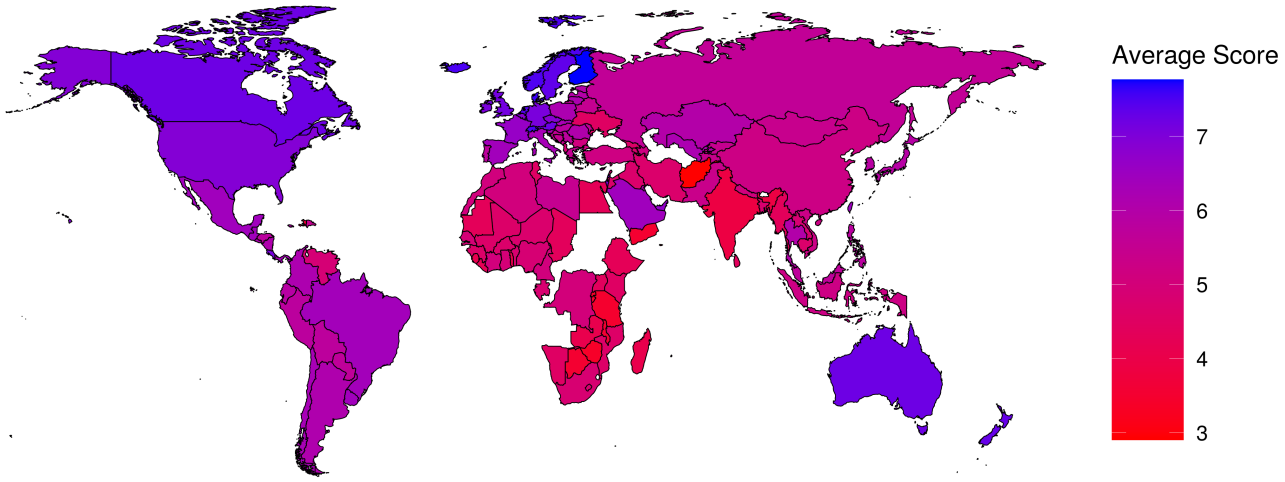
This plot depicts the Happiness Score for each country contained in the Happiness data sets, an average happiness score taken from the years 2018, 2020, and 2021 was used to find the average happiness score per country. This is displayed on the graph using two colors, the lower the happiness score the more red the color will be and the higher the happiness score the more blue the color will be. This allows us to easily identify which countries are the most happy or least happy. This showed that much of Africa and Western Asia were the least happy, and North America and Western Europe were the most happy.

```
#Change the figure size, this is the link I used: https://www.andrewheiss.com/blog/2022/06/23/long-labels-ggplot/

X2018_2020_2021_map_avg_happy #Look at data
```

```
## # A tibble: 83,011 × 5
##    Country     mean_Score  long   lat group
##    <chr>            <dbl> <dbl> <dbl> <dbl>
##  1 Afghanistan       2.91  74.9  37.2     2
##  2 Afghanistan       2.91  74.8  37.2     2
##  3 Afghanistan       2.91  74.8  37.2     2
##  4 Afghanistan       2.91  74.7  37.3     2
##  5 Afghanistan       2.91  74.7  37.3     2
##  6 Afghanistan       2.91  74.7  37.3     2
##  7 Afghanistan       2.91  74.6  37.2     2
##  8 Afghanistan       2.91  74.4  37.2     2
##  9 Afghanistan       2.91  74.4  37.1     2
## 10 Afghanistan       2.91  74.5  37.1     2
## # … with 83,001 more rows
```

```
X2018_2020_2021_map_avg_happy%>%
  ggplot(aes(x = long, y = lat, group = group, fill = mean_Score)) + #Set aesthetics
  geom_polygon(colour = "black") + # Display the country borders in black
  scale_fill_gradient(low = "red", high = "blue")+ #Color from red to blue
  labs(title = "Average Global Happiness Scores" , #Label title
       fill="Average Score")+ #Label fill
  theme_classic()+ #Change theme
  theme(legend.key.size = unit(3, 'cm'), #Change legend size
        legend.title = element_text(size=30), #Change legend title size
        legend.text = element_text(size=25), #Change legend text size
        title = element_text(size=80), #Change title text size
        axis.text.x=element_blank(), #remove x axis labels
        axis.ticks.x=element_blank(), #remove x axis ticks
        axis.text.y=element_blank(),  #remove y axis labels
        axis.ticks.y=element_blank(), #remove y axis ticks
        axis.title.x=element_blank(), #remove x axis title
        axis.title.y=element_blank(), #remove x axis title
        axis.line = element_blank()) #remove axis lines
```

Average Global Happiness Scores




```
#This is the source I used: https://www.statology.org/remove-axis-labels-ggplot2/
```

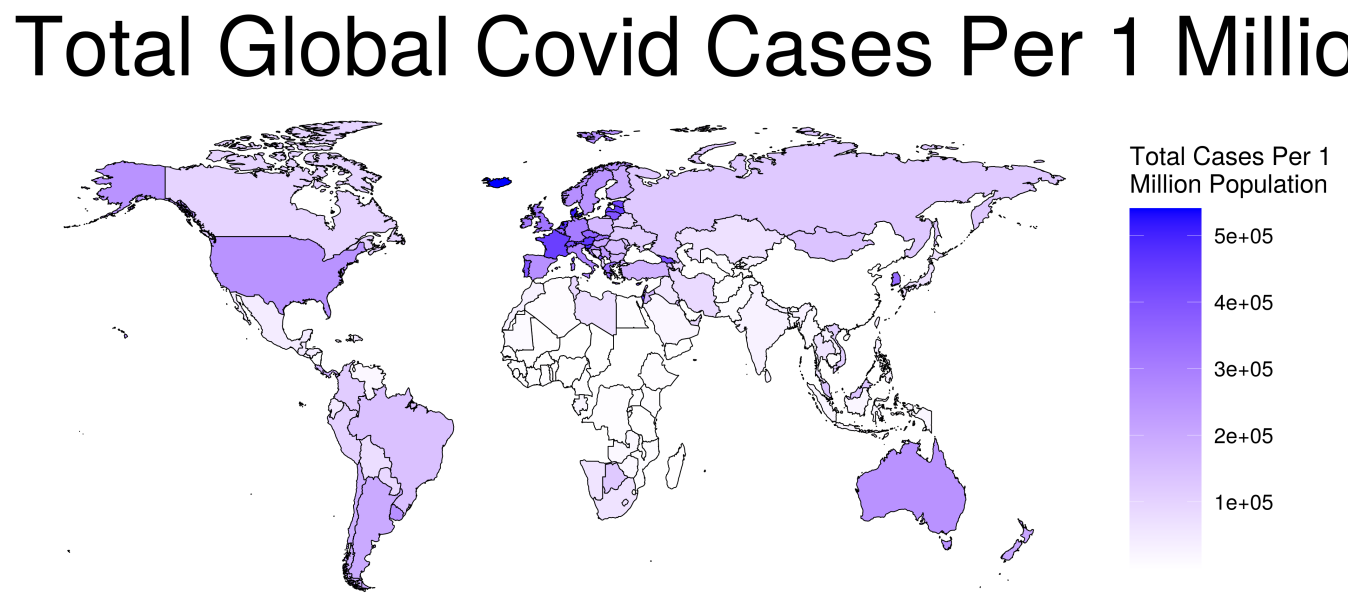
Total Global Covid Cases Per 1 Million Population

This plot depicts the Total Covid Cases per 1 Million of the Population of each country contained in the Happiness and Covid data sets. Using the total cases per 1 million of population helps us to get a better idea of how greatly each country was impacted. This is displayed on the graph using the color blue, the more blue a country is the higher the number of covid cases per 1 million population.

```
#Change the figure size, this is the link I used: https://www.andrewheiss.com/blog/2022/06/23/long-labels-ggplot/

covid_sum_clean1 #Look at data set
```

```
## # A tibble: 226 × 12
##    country       conti…¹ total…² total…³ total…⁴ activ…⁵ serio…⁶ total…⁷ total…⁸
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Afghanistan   Asia     179267    7690  162202    9375    1124    4420     190
##  2 Albania       Europe   275574    3497  271826     251       2   95954    1218
##  3 Algeria       Africa   265816    6875  178371   80570       6    5865     152
##  4 Andorra       Europe    42156     153   41021     982      14  543983    1974
##  5 Angola        Africa    99194    1900   97149     145      NA    2853      55
##  6 Anguilla      North …    2984       9    2916      59       4  195646     590
##  7 Antigua And … North …    7721     137    7511      73       1   77646    1378
##  8 Argentina     South … 9101319  128729 8895999   76591     372  197992    2800
##  9 Armenia       Asia     422896    8623  412048    2225      NA  142219    2900
## 10 Aruba         North …   35693     213   35199     281      NA  331689    1979
## # … with 216 more rows, 3 more variables: total_tests <dbl>,
## #   total_tests_per_1m_population <dbl>, population <dbl>, and abbreviated
## #   variable names ¹continent, ²total_confirmed, ³total_deaths,
## #   ⁴total_recovered, ⁵active_cases, ⁶serious_or_critical,
## #   ⁷total_cases_per_1m_population, ⁸total_deaths_per_1m_population
```

```
X2018_2020_2021_map_avg_happy%>%
  left_join(covid_sum_clean1, by = c("Country"="country"))%>%
  ggplot(aes(x=long, y=lat, group=group, fill=total_cases_per_1m_population)) + #Set aesth
etics
  geom_polygon(colour = "black") + # Display the country borders in black
  scale_fill_gradient(low = "white", high = "blue")+ #Color from white to blue
  labs(title = "Total Global Covid Cases Per 1 Million Population" , #Label title
       fill="Total Cases Per 1 \nMillion Population")+ #Label fill
  theme_classic()+ #Change theme
  theme(legend.key.size = unit(3, 'cm'), #Change legend size
        legend.title = element_text(size=30), #Change legend title size
        legend.text = element_text(size=25), #Change legend text size
        title = element_text(size=80), #Change title text size
        axis.text.x=element_blank(), #remove x axis labels
        axis.ticks.x=element_blank(), #remove x axis ticks
        axis.text.y=element_blank(),  #remove y axis labels
        axis.ticks.y=element_blank(), #remove y axis ticks
        axis.title.x=element_blank(), #remove x axis title
        axis.title.y=element_blank(), #remove x axis title
        axis.line = element_blank()) #remove axis lines
```




```
#This is the source I used: https://www.statology.org/remove-axis-labels-ggplot2/
```

Two-Variable Graphs:

The Relationship Between Happiness Scores and Covid-19 Cases (2021)

In this graph, we focused on the year 2021 and compared the overall happiness score of each country to their amount of Covid-19 cases. We choose to visualize data for 2021 since the global population experienced the full effects of the pandemic during this year, in terms of climbing death rates, accessibility to a vaccine, and new emerging strains. We found that there is, surprisingly, a positive correlation between a country's happiness score and its rate of covid cases in relation to its population size. This could be

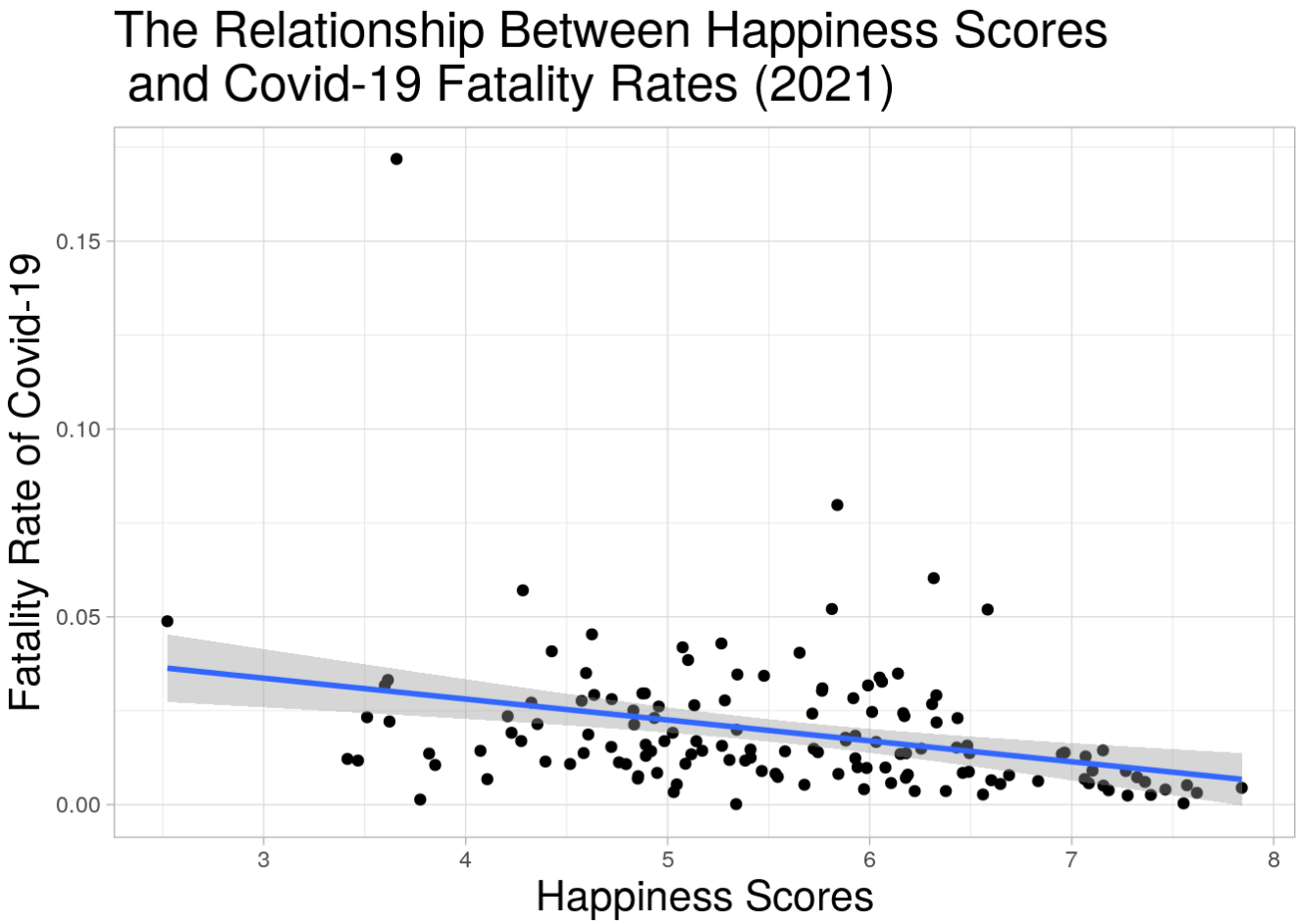
explained by the fact that more industrial and globalized countries that have higher standards of living made themselves most susceptible to the contacting the virus through higher levels of tourism, trading, and general day-to-day activity.

```
# Score vs cases 2021
happiness_sumcovid_yearlycovid %>%
  filter(Year == 2021) %>% #filter only the year 2021
  ggplot(aes(x = Score, y = new_cases_to_popsize)) + #Set aesthetics
  geom_point() + #Create scatterplot
  geom_smooth(method = 'lm') + #Create a smooth trend line
  ylim(0,0.2) + #Adjust y limit
  scale_x_continuous(breaks = seq(0,10,1)) + #Adjust x limit
  labs(title = 'The Relationship Between Happiness Scores \n and Covid-19 Cases (2021)',
       x = 'Happiness Score',
       y = 'Proportion of Cases to Pop. Size') + #Add labels
  theme_light()+ #Change theme
  theme(title=element_text(size=18)) #Adjust font size
```

Happiness Score vs. Fatality Rate for 2021

Rather than comparing a country's happiness score to its number of Covid-19 cases, this graph visualizes the relationship between happiness scores and Covid-19 fatality rates. We found that there is a slight negative correlation between the two, which was the outcome that we expected. Countries with higher fatality rates indicate a poorer ability to care for Covid-19 patients, which could be explained by a multitude of factors including low income rates and poor quality healthcare, both of which directly affect a population's happiness.

```
# Score vs fatality rate 2021
happiness_sumcovid_yearlycovid %>%
  filter(Year == 2021, #filter only the year 2021
         fatality_rate < .2) %>% #Remove outlier
  ggplot(aes(x = Score, y = fatality_rate)) + #Set aesthetics
  geom_point() + #Create scatterplot
  geom_smooth(method = 'lm') + #Create a smooth trend line
  scale_x_continuous(breaks = seq(0,10,1)) + #Adjust x limit
  labs(title = 'The Relationship Between Happiness Scores \n and Covid-19 Fatality Rates
 (2021)',
       x = 'Happiness Scores',
       y = 'Fatality Rate of Covid-19') + #Add labels
  theme_light()+ #Change theme
  theme(title=element_text(size=16)) #Adjust font size
```

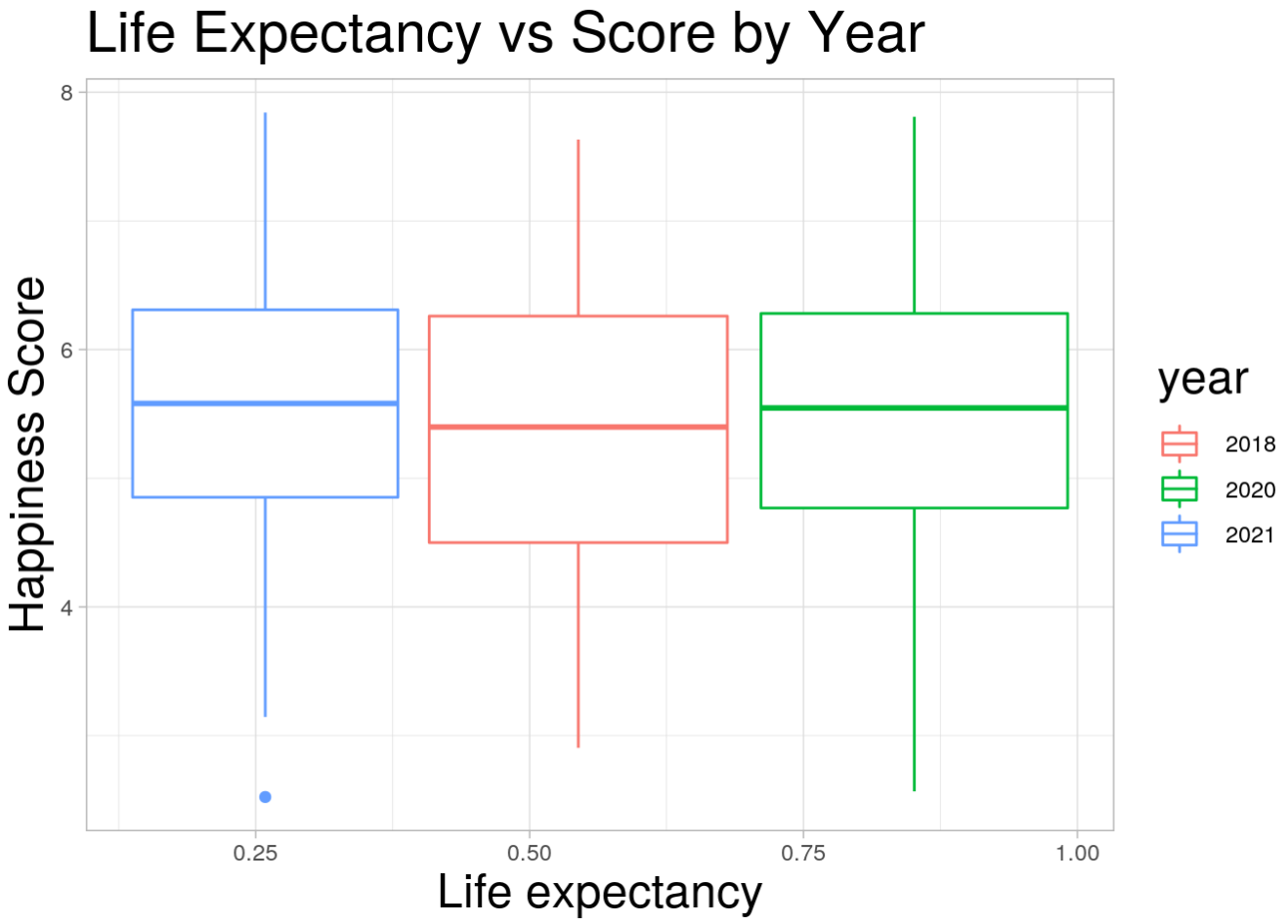


Three-Variable Graphs:

Life Expectancy vs Score by Year

Figure 1 demonstrates box plots of the life expectancy versus score by year distributions. In general, the year 2021 has a higher happiness score due to having a higher mean compared to the other years, yet it has the lowest life expectancy. The outlier indicates how it has an unordinary lower value. However, it would not be because that year is when COVID-19 cases skyrocketed, thus, reducing life expectancy. Therefore, people were most happy if there was more life expectancy in a time period where the chances were less.

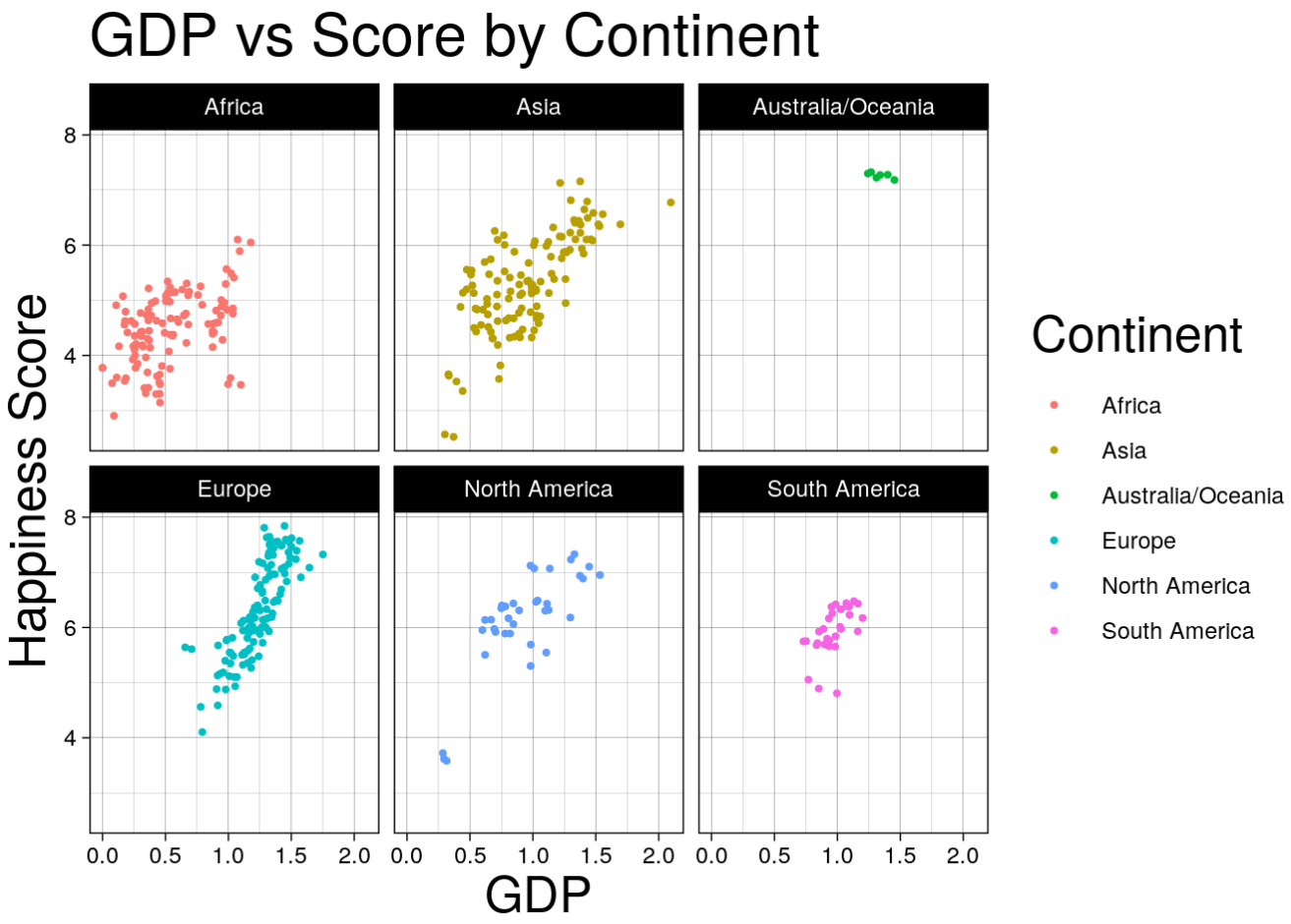
```
#Make the Year a Character
happiness_sumcovid_yearlycovid$year <- as.character(happiness_sumcovid_yearlycovid$Year)

happiness_sumcovid_yearlycovid%>%
  #Make a plot
  ggplot()+
  #Add labels
  labs(title = 'Life Expectancy vs Score by Year', x = 'Life expectancy', y = 'Happiness S
core') +
  #Make a boxplot
  geom_boxplot(aes(x = Life_expectancy, y =Score , color = year))+
  #Adjust scale
  scale_x_continuous(breaks = c(0.25, 0.5, 0.75,1))+
  #Adjust theme
  theme_light()+
  theme(title=element_text(size=18)) #Adjust font size
```



GDP vs Score by Continent

Figure 2 shows scatterplots of the gross domestic product (GDP) versus happiness score by continent distributions. Overall, the plots portray that the higher the GDP, the more happy the continent is. Europe has a higher happiness level compared to Africa, being the overall lowest in both areas. That is due to poor governance and low agricultural productivity, affecting their economy.

```
happiness_sumcovid_yearlycovid %>%
  ggplot(aes(x = GDP, y = Score, color = continent))+ #Set aesthetics
  labs(title = 'GDP vs Score by Continent', x = 'GDP', y = 'Happiness Score',
       color = 'Continent') + #Add labels
  geom_point(stat = "identity", size=0.75) + #Create scatterplot
  facet_wrap(~ continent, nrow = 2) + #Facet by continent
  theme_linedraw()+ #Change theme
  theme(title=element_text(size=19)) #Adjust font size
```

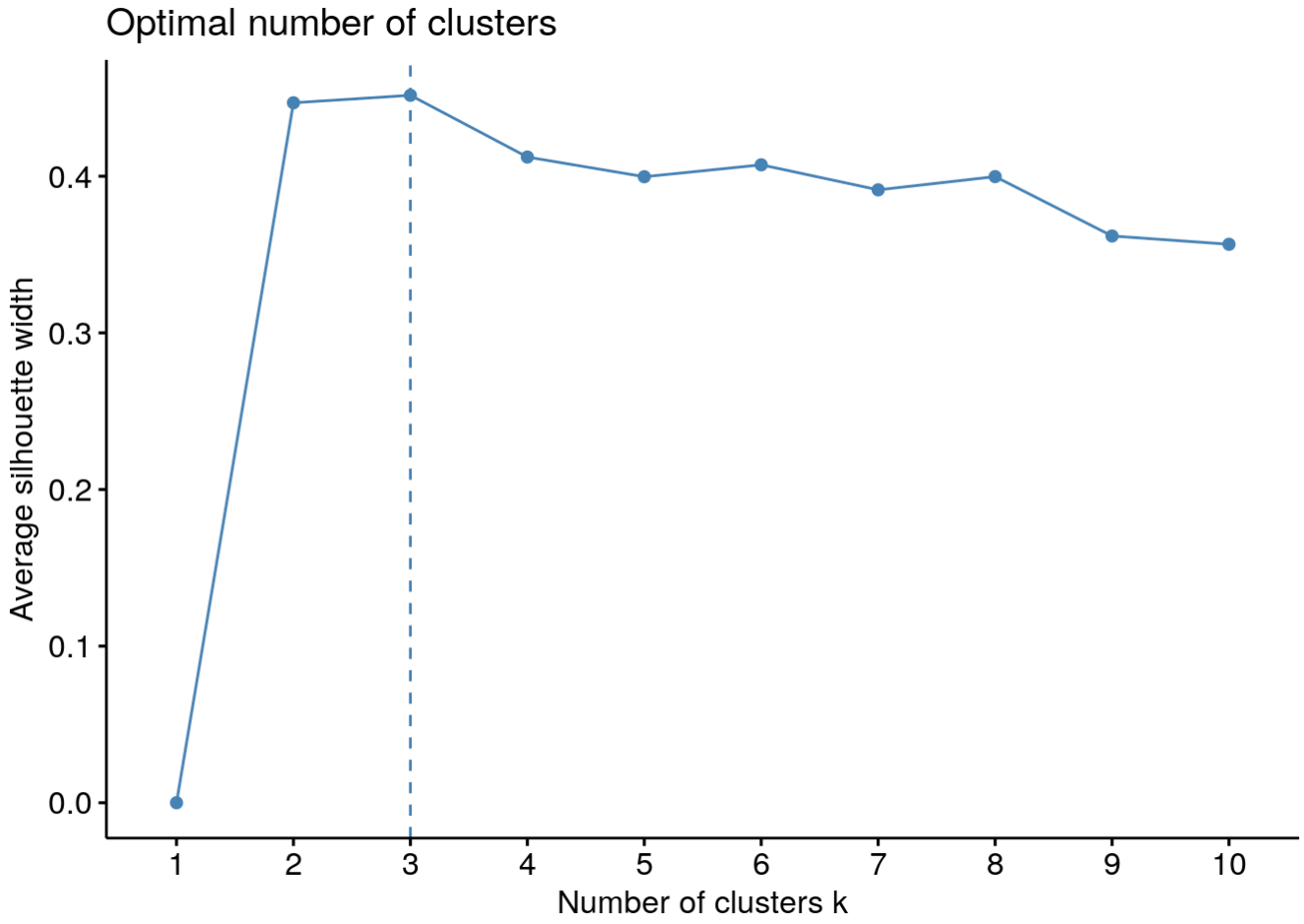


3. Clustering

PAM

PAM Clustering

```
hap_sumcov_yearlycov_vars <- happiness_sumcovid_yearlycovid %>%
  # variables
  select(country, continent, Score, total_cases_per_1m_population, total_deaths_per_1m_pop
ulation,    GDP, Life_expectancy, fatality_rate) %>%
  # Consider categorical variables as factors
  mutate_if(is.character, as.factor) %>%
  # Ignore missing values
  drop_na

# Calculate Gower distances between observations
hap_sumcov_yearlycov_vars %>%
  daisy(metric = "gower") %>%
  as.matrix -> hap_sumcov_yearlycov_gower

# Visualize clusters (optimal number = 3)
fviz_nbclust(hap_sumcov_yearlycov_gower, pam, method = "silhouette")
```




```
# Apply PAM
pam_results <- pam(hap_sumcov_yearlycov_gower, k = 3, diss = TRUE)
pam_results
```

```
## Medoids:
##      ID
## [1,] "276" "276"
## [2,] "236" "236"
## [3,] "80"  "80"
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   1   2   2   3   3   2   2   1   1   2   2   2   2   1   1   1   1   1   1
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   2   3   3   1   1   2   2   3   3   2   2   2   2   3   3   3   3
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   3   3   2   2   3   3   2   2   1   1   2   1   3   3   2   2   3   3
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   2   1   1   2   2   2   2   1   1   1   1   3   3   1   1   2   2   3   3
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   2   2   3   3   1   1   2   2   3   3   2   2   1   1   3   3   3   3
##  [ reached getOption("max.print") -- omitted 181 entries ]
## Objective function:
##     build      swap
## 0.2415923 0.2249695
##
## Available components:
## [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
## [6] "clusinfo"   "silinfo"    "diss"       "call"
```

```
# Add cluster column to reduced dataset
vars_pam <- hap_sumcov_yearlycov_vars %>%
  mutate(cluster = as.factor(pam_results$clustering))
vars_pam
```

```
## # A tibble: 281 × 9
##    country     continent     Score total…¹ total…²   GDP Life_…³ fatal…⁴ cluster
##    <fct>       <fct>         <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl> <fct>
##  1 Afghanistan Asia           2.57    4420     190 0.301   0.266  0.0419 1
##  2 Afghanistan Asia           2.52    4420     190 0.37    0.126  0.0488 1
##  3 Albania     Europe         4.88   95954    1218 0.907   0.846  0.0202 2
##  4 Albania     Europe         5.12   95954    1218 1.01    0.646  0.0134 2
##  5 Algeria     Africa         5.01    5865     152 0.944   0.745  0.0277 3
##  6 Algeria     Africa         4.89    5865     152 0.946   0.552  0.0296 3
##  7 Argentina   South America  5.97  197992    2800 1.03    0.850  0.0266 2
##  8 Argentina   South America  5.93  197992    2800 1.16    0.646  0.0183 2
##  9 Armenia     Asia           4.68  142219    2900 0.808   0.776  0.0177 1
## 10 Armenia     Asia           5.28  142219    2900 0.996   0.585  0.0278 1
## # … with 271 more rows, and abbreviated variable names
## #   ¹total_cases_per_1m_population, ²total_deaths_per_1m_population,
## #   ³Life_expectancy, ⁴fatality_rate
```

```
# Provides summary statistics for numeric variables in each cluster
vars_pam %>%
  group_by(cluster) %>%
  summarize_if(is.numeric, mean, na.rm = T)
```

```
## # A tibble: 3 × 7
##   cluster Score total_cases_per_1m_population total_deat…¹   GDP Life_…² fatal…³
##   <fct>   <dbl>                         <dbl>        <dbl> <dbl>   <dbl>   <dbl>
## 1 1        5.44                        93586.         896. 0.929   0.640  0.0231
## 2 2        6.45                       269251.        2405. 1.24    0.795  0.0191
## 3 3        4.51                        16404.         297. 0.530   0.346  0.0206
## # … with abbreviated variable names ¹total_deaths_per_1m_population,
## #   ²Life_expectancy, ³fatality_rate
```

```
# Distribution of continents within in each cluster
vars_pam %>%
  group_by(cluster, continent) %>%
  summarize(freq = n())
```

```
## # A tibble: 9 × 3
## # Groups:   cluster [3]
##   cluster continent          freq
##   <fct>   <fct>             <int>
## 1 1       Asia                 80
## 2 1       North America        12
## 3 1       South America        11
## 4 2       Australia/Oceania     4
## 5 2       Europe               78
## 6 2       North America        10
## 7 2       South America         9
## 8 3       Africa               75
## 9 3       North America         2
```

```
# Average silhouette width (weak structure ;-;)
pam_results$silinfo$avg.width
```

```
## [1] 0.3045656
```

After performing PAM clustering on our dataset based on eight different variables, each observation in our data was grouped into one of three separate clusters. First, we decided to analyze the clusters based on continent. We found that every Asian country was located within cluster 1, as well as a few North and South American countries. In cluster 2, we found every European country located here, as well as other small amounts of Australian, North American, and South American countries. Lastly, cluster 3 was comprised of every African country and only two North American countries. When analyzing numeric variable statistics for each cluster, such as mean happiness scores, mean deaths, etc., the results were a bit more mixed. Cluster 2 exhibited the highest average happiness scores (6.45), the highest amount of cases per 1 mil population, the highest amount of deaths per 1 mil population, highest GDP, highest life expectancy, and lowest fatality rate. Cluster 1 had the next highest summary statistics for each variable, and cluster 3 had the lowest

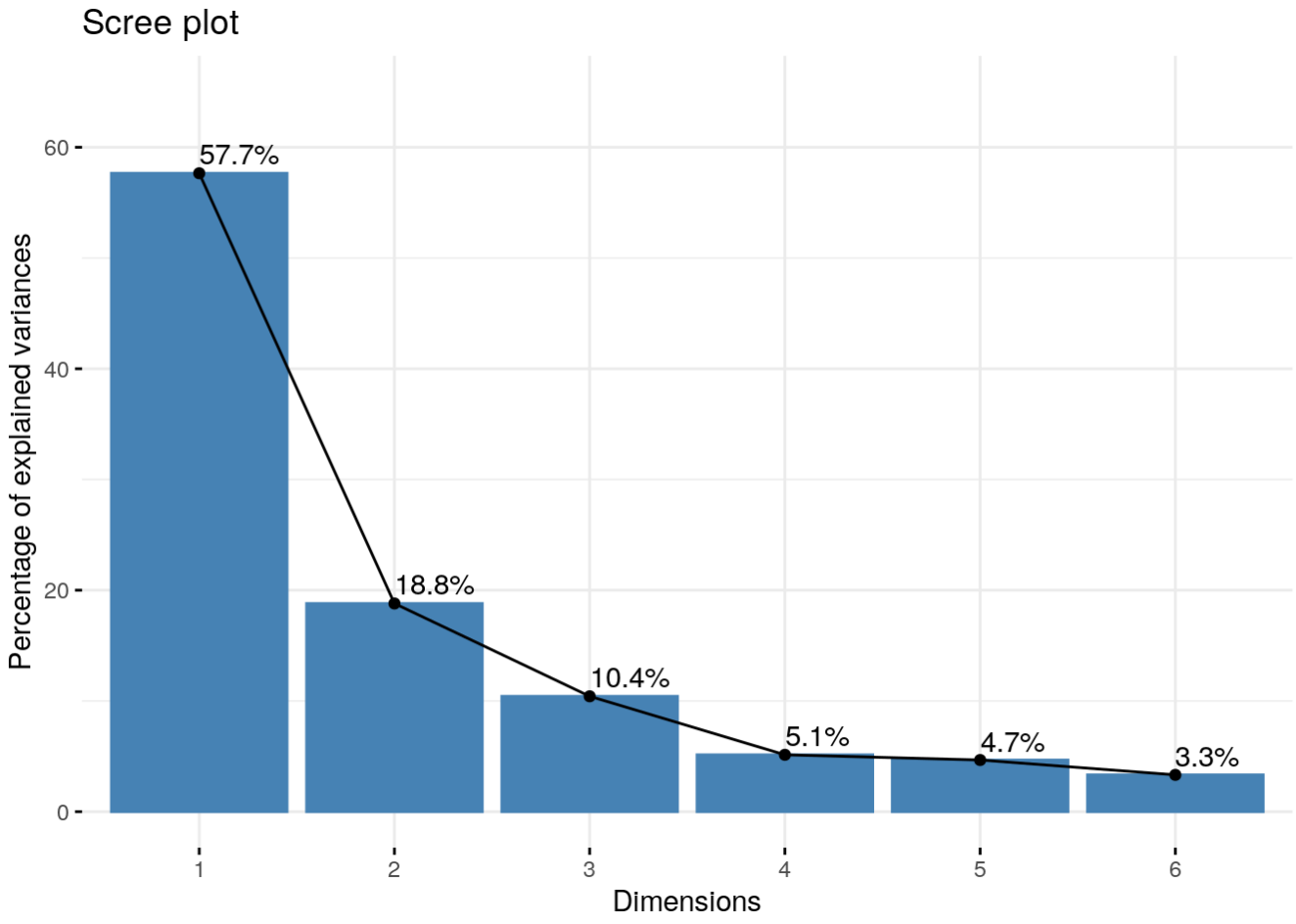
summary statistics. These trends make sense, since cluster 2 is comprised of wealthier, more developed European countries, cluster 1 is comprised of overall less wealthy Asian countries, and cluster 3 is made up of developing African countries. The average silhouette width is .305, indicating an overall weak structure.

4. Dimensionality Reduction

PCA

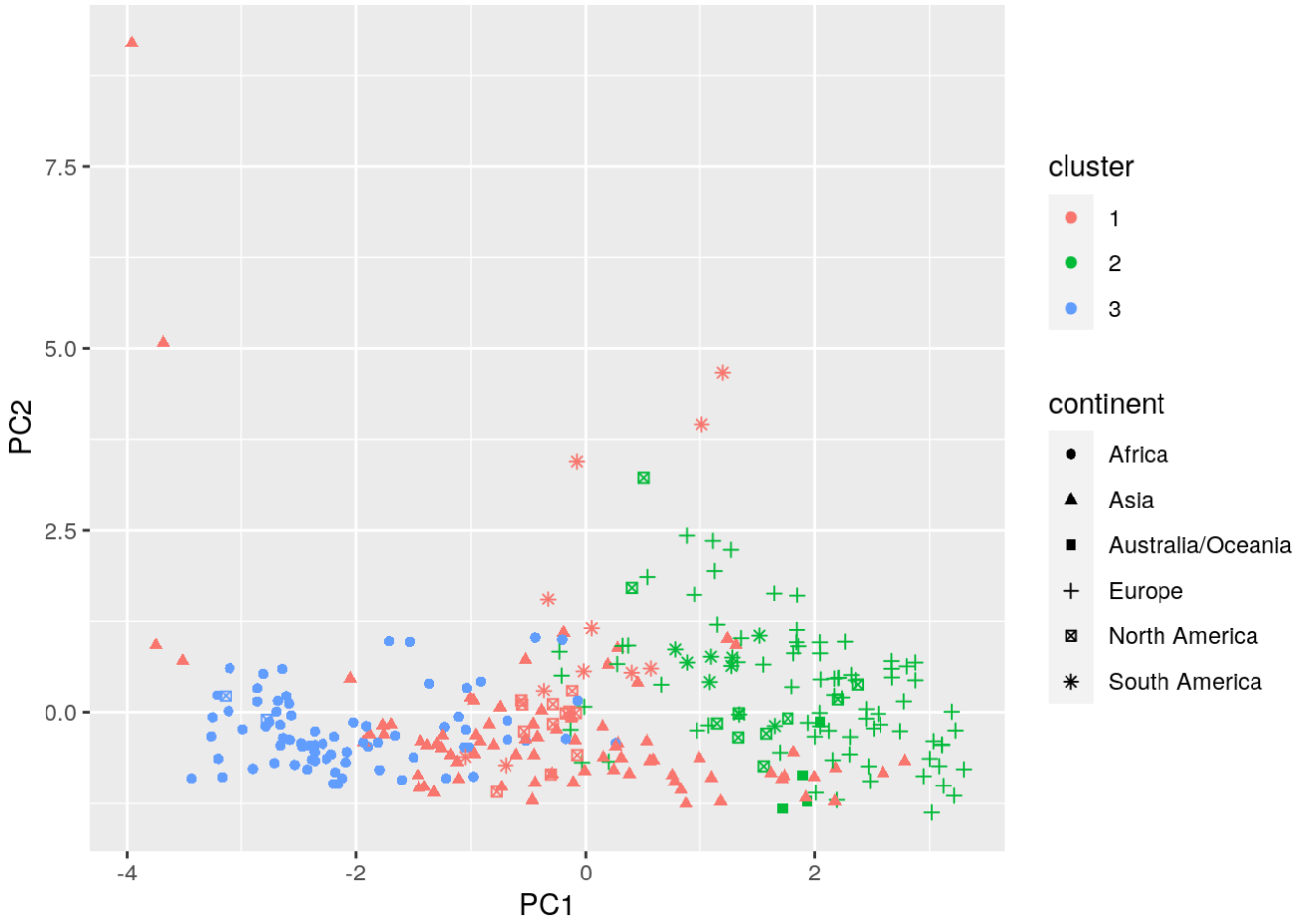
Dimensionality Reduction (PCA)

```
# Apply PCA
pca <- hap_sumcov_yearlycov_vars %>%
  select_if(is.numeric) %>%
  scale %>% # remember to scale
  prcomp

# Variation explained by each PC
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 65))
```



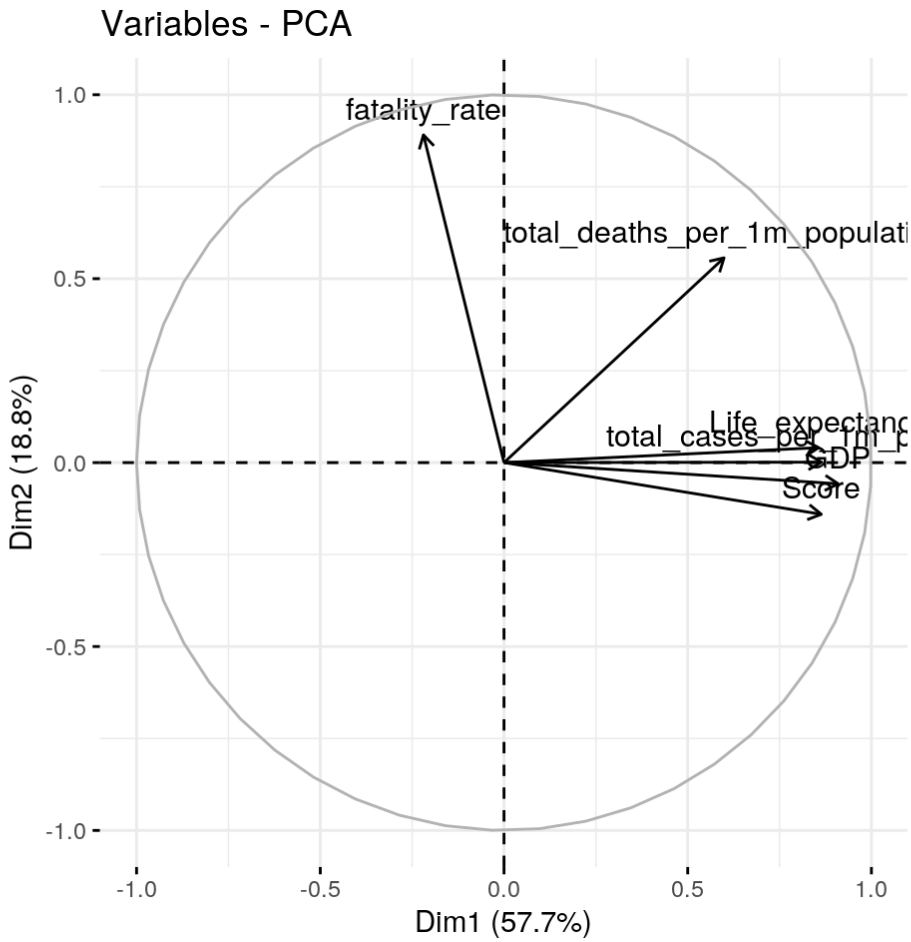
Visualizing clusters based on PC1 and PC2

```
#Make Visualization
pca$x %>%
  as.data.frame %>%
  mutate(cluster = as.factor(pam_results$clustering),
         continent = hap_sumcov_yearlycov_vars$continent) %>% # also add vore
  ggplot(aes(x = PC1, PC2,
             color = cluster, shape = continent)) +
  geom_point()
```



Visualizing PC1 and PC2

```
fviz_pca_var(pca) #Make visualization
```



Scoring high on PC1 indicates a high likelihood that the observation belongs in cluster 2, while scoring low likely places the observation in cluster 3. To be more specific, scoring high on PC1 indicates an observation has high values for (happiness) Score, GDP, Life expectancy, and total cases, a moderate value for total deaths, and a lower value for fatality rate. Scoring high on PC2 likely places the observation in cluster 1 or 2, but there is not a significant trend here. It may also indicate that the observation has a high value for fatality rate or a moderate value for total deaths. Scoring low of PC2 cannot very accurately predict which cluster an observation may belong to. PC1 accounts for 57.7% of the total variation in the dataset, and PC2 accounts for 18.8% of the total variation.

5. Classification and Cross-Validation

Make a Binary Variable

There were no preexisting binary variables in our datasets so we needed to create one. We decided to create a variable about the deadliness of a country with the cut off determined by the average deaths per 1 million of the population, if the country has deaths per 1 million below the average it is ‘safe’ and if it is above it is ‘deadly’.

```
happiness_sumcovid_yearlycovid%>%
  #Find the average deaths per 1 million of the population
  summarize(avg_deaths_per_1m_population = mean(total_deaths_per_1m_population))
```

```
## # A tibble: 1 × 1
##   avg_deaths_per_1m_population
##                          <dbl>
## 1                        1271.
```

```
#Rename
happiness_sumcovid_yearlycovid_deadliness <-happiness_sumcovid_yearlycovid%>%
  #Pick one year
  filter(Year=="2020")%>%
  #New binary variable based on average
  mutate(Deadliness = ifelse(total_deaths_per_1m_population >= 1271.191, "Deadly","Safe"))

#Check dataset
happiness_sumcovid_yearlycovid_deadliness
```

```
## # A tibble: 141 × 27
##    country     continent total…¹ total…² total…³ activ…⁴ serio…⁵ total…⁶ total…⁷
##    <chr>       <chr>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Afghanistan Asia       179267    7690  162202    9375    1124    4420     190
##  2 Albania     Europe     275574    3497  271826     251       2   95954    1218
##  3 Algeria     Africa     265816    6875  178371   80570       6    5865     152
##  4 Argentina   South Am… 9101319  128729 8895999   76591     372  197992    2800
##  5 Armenia     Asia       422896    8623  412048    2225      NA  142219    2900
##  6 Australia   Australi… 6593795    7794 6199822  386179     129  253112     299
##  7 Austria     Europe    4212492   18303 4135885   58304      58  462804    2011
##  8 Azerbaijan  Asia       792638    9709  782869      60      NA   76885     942
##  9 Bahrain     Asia       576997    1479  569758    5760       4  318491     816
## 10 Bangladesh  Asia      1953012   29127 1899419   24466    1273   11643     174
## # … with 131 more rows, 18 more variables: total_tests <dbl>,
## #   total_tests_per_1m_population <dbl>, population <dbl>,
## #   Regional_indicator <chr>, Score <dbl>, GDP <dbl>, Social_support <dbl>,
## #   Life_expectancy <dbl>, Freedom <dbl>, Generosity <dbl>, Corruption <chr>,
## #   Year <dbl>, new_cases_per_year <dbl>, new_deaths_per_year <dbl>,
## #   new_cases_to_popsize <dbl>, fatality_rate <dbl>, year <chr>,
## #   Deadliness <chr>, and abbreviated variable names ¹total_confirmed, …
```

```
#Number of deadly countries
sum(happiness_sumcovid_yearlycovid_deadliness$Deadliness=="Deadly")
```

```
## [1] 54
```

```
#Number of safe countries
sum(happiness_sumcovid_yearlycovid_deadliness$Deadliness=="Safe")
```

```
## [1] 87
```

kNN

Perform a kNN on the entire dataset

```
#Rename and kNN
deadliness_kNN <- knn3(Deadliness ~ total_cases_per_1m_population+GDP+Life_expectancy+Scor
e,
                data = happiness_sumcovid_yearlycovid_deadliness,
                k = 5) # number of neighbors
deadliness_kNN #Check
```

```
## 5-nearest neighbor model
## Training set outcome distribution:
##    Length     Class      Mode
##       141 character character
```

Make Predictions

```
predict(deadliness_kNN, happiness_sumcovid_yearlycovid_deadliness)%>% as.data.frame %>% he
ad #Predict and look at sample of predictions
```

```
##   Deadly Safe
## 1    0.0  1.0
## 2    0.2  0.8
## 3    0.0  1.0
## 4    0.4  0.6
## 5    0.6  0.4
## 6    0.6  0.4
```

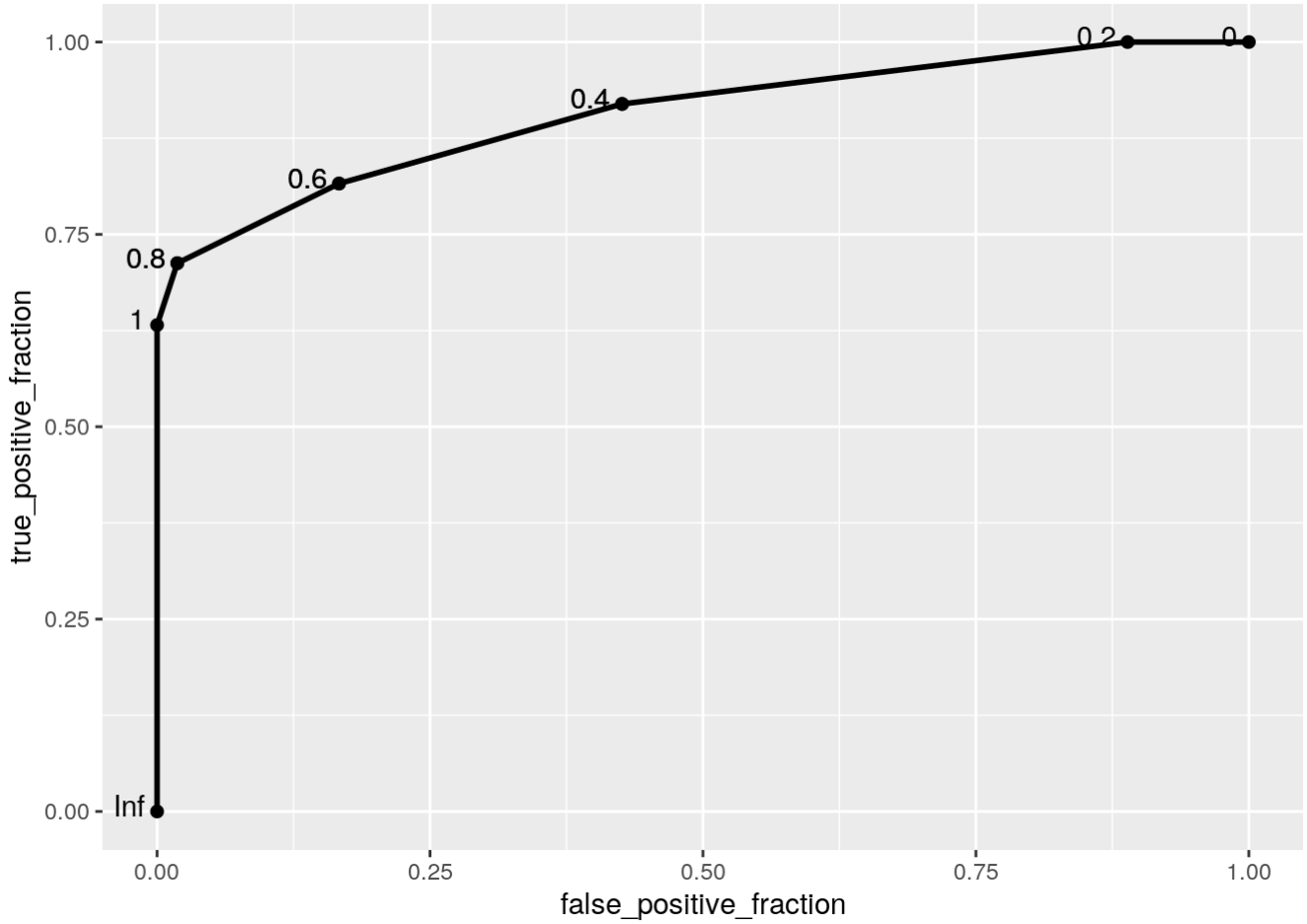
```
deadliness_pred<-happiness_sumcovid_yearlycovid_deadliness%>% #Rename
  mutate(predictions=predict(deadliness_kNN, happiness_sumcovid_yearlycovid_deadliness)[,2
],
         predicted = ifelse(predictions > 0.5, "Deadly","Safe")) #Make predictions

deadliness_pred #Check
```

```
## # A tibble: 141 × 29
##    country     continent total…¹ total…² total…³ activ…⁴ serio…⁵ total…⁶ total…⁷
##    <chr>       <chr>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Afghanistan Asia       179267    7690  162202    9375    1124    4420     190
##  2 Albania     Europe     275574    3497  271826     251       2   95954    1218
##  3 Algeria     Africa     265816    6875  178371   80570       6    5865     152
##  4 Argentina   South Am… 9101319  128729 8895999   76591     372  197992    2800
##  5 Armenia     Asia       422896    8623  412048    2225      NA  142219    2900
##  6 Australia   Australi… 6593795    7794 6199822  386179     129  253112     299
##  7 Austria     Europe    4212492   18303 4135885   58304      58  462804    2011
##  8 Azerbaijan  Asia       792638    9709  782869      60      NA   76885     942
##  9 Bahrain     Asia       576997    1479  569758    5760       4  318491     816
## 10 Bangladesh  Asia      1953012   29127 1899419   24466    1273   11643     174
## # … with 131 more rows, 20 more variables: total_tests <dbl>,
## #   total_tests_per_1m_population <dbl>, population <dbl>,
## #   Regional_indicator <chr>, Score <dbl>, GDP <dbl>, Social_support <dbl>,
## #   Life_expectancy <dbl>, Freedom <dbl>, Generosity <dbl>, Corruption <chr>,
## #   Year <dbl>, new_cases_per_year <dbl>, new_deaths_per_year <dbl>,
## #   new_cases_to_popsize <dbl>, fatality_rate <dbl>, year <chr>,
## #   Deadliness <chr>, predictions <dbl>, predicted <chr>, and abbreviated …
```

ROC and AUC

ROC Curve

```
ROC <- ggplot(deadliness_pred) +
  geom_roc(aes(d = Deadliness, m = predictions), n.cuts = 10) #Make ROC curve
ROC #Check
```



AUC

```
calc_auc(ROC)$AUC #Find AUC
```

```
## [1] 0.9061303
```

An AUC of 0.9061303 indicates that the model predicts the data fairly well, which is reflected by the ROC curve above. This AUC indicates that the true positive rate of the data is about 91% meaning only 9% of the time there is a false positive, this means the model works fairly well.

Cross-validation

```
# Choose number of folds
k = 10

# Randomly order rows in the dataset
data <- happiness_sumcovid_yearlycovid_deadliness[sample(nrow(happiness_sumcovid_yearlycov
id_deadliness)), ]

# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Initialize a vector to keep track of the performance
perf_k <- NULL

# Use a for loop to get diagnostics for each test set
for(i in 1:k){
  # Create train and test sets
  train <- data[folds != i, ] # all observations except in fold i
  test <- data[folds == i, ]  # observations in fold i

  # Train model on train set (all but fold i)
  deadliness_kNN_cross <- knn3(Deadliness ~ total_cases_per_1m_population+GDP+Life_expecta
ncy+Score,
                data = train,
                k = 5) # number of neighbors

  # Test model on test set (fold i)
   df <- data.frame(
    predictions = predict(deadliness_kNN_cross, test)[,2],
    Deadliness = test$Deadliness)

  # Consider the ROC curve for the test dataset
  ROC <- ggplot(df) +
    geom_roc(aes(d = Deadliness, m = predictions))

  # Get diagnostics for fold i (AUC)
  perf_k[i] <- calc_auc(ROC)$AUC
}

# Average performance
mean(perf_k)
```

```
## [1] 0.8251073
```

The average AUC for the model trained on only one fold from the cross-validation was 0.8251073 and the AUC for the model trained on the entire data set was 0.9061303. The decreased average AUC from the cross-validation indicates over-fitting of the data.