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).

2. Tidying

Cleaning the Data Sets

Cleaning the 2018 Happiness Report to the Covid Summary

Anti-join was used to see which variables from the 2018 Happiness data set were missing from the Covid Summary data set. The countries from the Happiness data set that did not match were checked for spelling differences and recoded to make them consistent.

```
#Cleaning 2018 to Covid

  X2018%>% #Check which values from happiness report 2018 do not match the covid summary
    #Manually check which values
    anti_join(covid_sum, by = c("Country or region"="country"))
```

```
## # A tibble: 13 × 9
##    `Overall rank` Countr…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##             <dbl> <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1             11 United …  7.19   1.24    1.43    0.888   0.464   0.262 0.082
##  2             18 United …  6.89   1.40    1.47    0.819   0.547   0.291 0.133
##  3             38 Trinida…  6.19   1.22    1.49    0.564   0.575   0.171 0.019
##  4             58 Norther…  5.84   1.23    1.21    0.909   0.495   0.179 0.154
##  5             66 Kosovo    5.66   0.855   1.23    0.578   0.448   0.274 0.023
##  6             68 Turkmen…  5.64   1.02    1.53    0.517   0.417   0.199 0.037
##  7             76 Hong Ko…  5.43   1.40    1.29    1.03    0.524   0.246 0.291
##  8             93 Bosnia …  5.13   0.915   1.08    0.758   0.28    0.216 0.000
##  9             95 Vietnam   5.10   0.715   1.36    0.702   0.618   0.177 0.079
## 10            104 Palesti…  4.74   0.642   1.22    0.602   0.266   0.086 0.076
## 11            107 Ivory C…  4.67   0.541   0.872   0.08    0.467   0.146 0.103
## 12            114 Congo (…  4.56   0.682   0.811   0.343   0.514   0.091 0.077
## 13            132 Congo (…  4.24   0.069   1.14    0.204   0.312   0.197 0.052
## # … with abbreviated variable names ¹`Country or region`, ²`GDP per capita`,
## #   ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
```

```
X2018_clean1 <- X2018%>% #Clean up so they match and rename
  mutate(`Country or region`=recode(`Country or region`, 'Trinidad & Tobago'='Trinidad And Tobago',
          'Bosnia and Herzegovina'='Bosnia And Herzegovina',
          'Congo (Kinshasa)'='Democratic Republic Of The Congo',
          'Congo (Brazzaville)'='Congo')) #This code renames countries so that they match

X2018_clean1 #Check that it works
```

```
## # A tibble: 156 × 9
##    `Overall rank` Countr…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##             <dbl> <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1              1 Finland   7.63    1.30    1.59   0.874   0.681   0.202 0.393
##  2              2 Norway    7.59    1.46    1.58   0.861   0.686   0.286 0.340
##  3              3 Denmark   7.56    1.35    1.59   0.868   0.683   0.284 0.408
##  4              4 Iceland   7.50    1.34    1.64   0.914   0.677   0.353 0.138
##  5              5 Switzer…  7.49    1.42    1.55   0.927   0.66    0.256 0.357
##  6              6 Netherl…  7.44    1.36    1.49   0.878   0.638   0.333 0.295
##  7              7 Canada    7.33    1.33    1.53   0.896   0.653   0.321 0.291
##  8              8 New Zea…  7.32    1.27    1.60   0.876   0.669   0.365 0.389
##  9              9 Sweden    7.31    1.36    1.50   0.913   0.659   0.285 0.383
## 10             10 Austral…  7.27    1.34    1.57   0.91    0.647   0.361 0.302
## # … with 146 more rows, and abbreviated variable names ¹`Country or region`,
## #   ²`GDP per capita`, ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows
```

Cleaning up the Covid Summary

Using the information from the code chunk above about which countries spelling did not match, values from the Covid Summary data set were recoded to match the 2018 Happiness Report.

```
covid_sum_clean1 <- covid_sum%>% #Clean up so they match and rename
  mutate(country=recode(country, 'USA'='United States', 'UK'='United Kingdom', 'Viet Nam'='Vietnam',
          'State Of Palestine'='Palestinian Territories','Cote D Ivoire'='Ivory Coast'))
          #This code renames countries so they match

covid_sum_clean1 #Check it
```

```
## # 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
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
  X2018_clean1%>% #Check that it worked
    anti_join(covid_sum_clean1, by = c("Country or region"="country"))
```

```
## # A tibble: 4 × 9
##   `Overall rank` Country…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##            <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 1             58 Northern…  5.84   1.23     1.21   0.909   0.495   0.179 0.154
## 2             66 Kosovo     5.66   0.855    1.23   0.578   0.448   0.274 0.023
## 3             68 Turkmeni…  5.64   1.02     1.53   0.517   0.417   0.199 0.037
## 4             76 Hong Kong  5.43   1.40     1.29   1.03    0.524   0.246 0.291
## # … with abbreviated variable names ¹`Country or region`, ²`GDP per capita`,
## #   ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
```

Cleaning Up the 2020 Happiness Report

Anti-join was used to see which variables from the 2020 Happiness data set were missing from the Covid Summary data set. The countries from the Happiness data set that did not match were checked for spelling differences and recoded to make them consistent.

```
#Cleaning 2020 to Covid

  X2020%>% #Check which values from happiness report 2020 do not match cleaned covid data
    anti_join(covid_sum_clean1, by = c("Country name"="country"))
```

```
## # A tibble: 9 × 20
##   `Country name` Regio…¹ Ladde…² Stand…³ upper…⁴ lower…⁵ Logge…⁶ Socia…⁷ Healt…⁸
##   <chr>          <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Taiwan Provin… East A…    6.46  0.0391    6.53    6.38   10.8    0.894    69.6
## 2 Kosovo         Centra…    6.33  0.0522    6.43    6.22    9.20   0.821    63.9
## 3 Trinidad and … Latin …    6.19  0.114     6.42    5.97   10.3    0.915    63.5
## 4 Bosnia and He… Centra…    5.67  0.0464    5.77    5.58    9.46   0.829    67.8
## 5 North Cyprus   Wester…    5.54  0.0510    5.64    5.44   10.4    0.820    73.7
## 6 Hong Kong S.A… East A…    5.51  0.0460    5.60    5.42   10.9    0.846    76.8
## 7 Congo (Brazza… Sub-Sa…    5.19  0.0770    5.35    5.04    8.54   0.640    57.9
## 8 Turkmenistan   Common…    5.12  0.0294    5.18    5.06    9.75   0.959    62.2
## 9 Congo (Kinsha… Sub-Sa…    4.31  0.109     4.52    4.10    6.69   0.672    52.9
## # … with 11 more variables: `Freedom to make life choices` <dbl>,
## #   Generosity <dbl>, `Perceptions of corruption` <dbl>,
## #   `Ladder score in Dystopia` <dbl>, `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>,
## #   `Explained by: Generosity` <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

```
  X2020_clean1 <- X2020%>% #Clean up so they match and rename
    mutate(`Country name`=recode(`Country name`,
            'Trinidad and Tobago'='Trinidad And Tobago',
            'Bosnia and Herzegovina'='Bosnia And Herzegovina',
            'Congo (Kinshasa)'='Democratic Republic Of The Congo',
            'Congo (Brazzaville)'='Congo',
            'Taiwan Province of China'='Taiwan')) #This code renames countries so that they match

  X2020_clean1 #Check it
```

```
## # A tibble: 153 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.81  0.0312    7.87    7.75    10.6   0.954    71.9
##  2 Denmark       Wester…    7.65  0.0335    7.71    7.58    10.8   0.956    72.4
##  3 Switzerland   Wester…    7.56  0.0350    7.63    7.49    11.0   0.943    74.1
##  4 Iceland       Wester…    7.50  0.0596    7.62    7.39    10.8   0.975    73
##  5 Norway        Wester…    7.49  0.0348    7.56    7.42    11.1   0.952    73.2
##  6 Netherlands   Wester…    7.45  0.0278    7.50    7.39    10.8   0.939    72.3
##  7 Sweden        Wester…    7.35  0.0362    7.42    7.28    10.8   0.926    72.6
##  8 New Zealand   North …    7.30  0.0395    7.38    7.22    10.5   0.949    73.2
##  9 Austria       Wester…    7.29  0.0334    7.36    7.23    10.7   0.928    73.0
## 10 Luxembourg    Wester…    7.24  0.0309    7.30    7.18    11.5   0.907    72.6
## # … with 143 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
  X2020_clean1%>% #Check that it worked
    anti_join(covid_sum_clean1, by = c("Country name"="country"))
```

```
## # A tibble: 4 × 20
##   `Country name` Regio…¹ Ladde…² Stand…³ upper…⁴ lower…⁵ Logge…⁶ Socia…⁷ Healt…⁸
##   <chr>          <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Kosovo         Centra…    6.33  0.0522    6.43    6.22    9.20   0.821    63.9
## 2 North Cyprus   Wester…    5.54  0.0510    5.64    5.44   10.4    0.820    73.7
## 3 Hong Kong S.A… East A…    5.51  0.0460    5.60    5.42   10.9    0.846    76.8
## 4 Turkmenistan   Common…    5.12  0.0294    5.18    5.06    9.75   0.959    62.2
## # … with 11 more variables: `Freedom to make life choices` <dbl>,
## #   Generosity <dbl>, `Perceptions of corruption` <dbl>,
## #   `Ladder score in Dystopia` <dbl>, `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>,
## #   `Explained by: Generosity` <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

Cleaning 2021 Happiness Report

Anti-join was used to see which variables from the 2021 Happiness data set were missing from the Covid Summary data set. The countries from the Happiness data set that did not match were checked for spelling differences and recoded to make them consistent.

```
#Cleaning 2021 to Covid

  X2021%>% #Check which values from hapiness report 2018 do not match cleaned covid data
    anti_join(covid_sum_clean1, by = c("Country name"="country"))
```

```
## # A tibble: 8 × 20
##   `Country name` Regio…¹ Ladde…² Stand…³ upper…⁴ lower…⁵ Logge…⁶ Socia…⁷ Healt…⁸
##   <chr>          <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Taiwan Provin… East A…    6.58   0.038    6.66    6.51   10.9    0.898    69.6
## 2 Kosovo         Centra…    6.37   0.059    6.49    6.26    9.32   0.821    63.8
## 3 Bosnia and He… Centra…    5.81   0.05     5.91    5.72    9.59   0.87     68.1
## 4 North Cyprus   Wester…    5.54   0.051    5.64    5.44   10.6    0.82     73.9
## 5 Hong Kong S.A… East A…    5.48   0.049    5.57    5.38   11      0.836    76.8
## 6 Congo (Brazza… Sub-Sa…    5.34   0.097    5.53    5.15    8.12   0.636    58.2
## 7 North Macedon… Centra…    5.10   0.051    5.20    5.00    9.69   0.805    65.5
## 8 Turkmenistan   Common…    5.07   0.036    5.14    5.00    9.63   0.983    62.4
## # … with 11 more variables: `Freedom to make life choices` <dbl>,
## #   Generosity <dbl>, `Perceptions of corruption` <dbl>,
## #   `Ladder score in Dystopia` <dbl>, `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>,
## #   `Explained by: Generosity` <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

```
X2021_clean1 <- X2021%>% #Clean up so they match and rename
  mutate(`Country name`=recode(`Country name`,'Bosnia and Herzegovina'='Bosnia And Herzegovina',
          'Congo (Brazzaville)'='Congo','Taiwan Province of China'='Taiwan',
          'North Macedonia'='Macedonia'))
          #This code renames countries so that they match

X2021_clean1 #Check it
```

```
## # A tibble: 149 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.84   0.032    7.90    7.78    10.8   0.954    72
##  2 Denmark       Wester…    7.62   0.035    7.69    7.55    10.9   0.954    72.7
##  3 Switzerland   Wester…    7.57   0.036    7.64    7.5     11.1   0.942    74.4
##  4 Iceland       Wester…    7.55   0.059    7.67    7.44    10.9   0.983    73
##  5 Netherlands   Wester…    7.46   0.027    7.52    7.41    10.9   0.942    72.4
##  6 Norway        Wester…    7.39   0.035    7.46    7.32    11.1   0.954    73.3
##  7 Sweden        Wester…    7.36   0.036    7.43    7.29    10.9   0.934    72.7
##  8 Luxembourg    Wester…    7.32   0.037    7.40    7.25    11.6   0.908    72.6
##  9 New Zealand   North …    7.28   0.04     7.36    7.20    10.6   0.948    73.4
## 10 Austria       Wester…    7.27   0.036    7.34    7.20    10.9   0.934    73.3
## # … with 139 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2021_clean1%>% #Check that it worked
  anti_join(covid_sum_clean1, by = c("Country name"="country"))
```

```
## # A tibble: 4 × 20
##   `Country name` Regio…¹ Ladde…² Stand…³ upper…⁴ lower…⁵ Logge…⁶ Socia…⁷ Healt…⁸
##   <chr>          <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Kosovo         Centra…    6.37   0.059    6.49    6.26    9.32   0.821    63.8
## 2 North Cyprus   Wester…    5.54   0.051    5.64    5.44   10.6    0.82     73.9
## 3 Hong Kong S.A… East A…    5.48   0.049    5.57    5.38   11      0.836    76.8
## 4 Turkmenistan   Common…    5.07   0.036    5.14    5.00    9.63   0.983    62.4
## # … with 11 more variables: `Freedom to make life choices` <dbl>,
## #   Generosity <dbl>, `Perceptions of corruption` <dbl>,
## #   `Ladder score in Dystopia` <dbl>, `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>,
## #   `Explained by: Generosity` <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

Cleaning the Happiness Reports

Cleaning 2018 against 2020

Anti-join was used to see which variables from the 2018 Happiness data set were missing from the 2020 Happiness data set. The countries from the 2018 Happiness data set that did not match were checked for spelling differences and recoded to make them consistent.

```
X2018_clean1%>% #Check which values are in 2018 and not in 2020
  anti_join(X2020_clean1, by = c("Country or region"="Country name"))
```

```
## # A tibble: 9 × 9
##   `Overall rank` Country…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##            <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 1             32 Qatar      6.37   1.65    1.30    0.748   0.654   0.256 0.171
## 2             49 Belize     5.96   0.807   1.10    0.474   0.593   0.183 0.089
## 3             58 Northern…  5.84   1.23    1.21    0.909   0.495   0.179 0.154
## 4             76 Hong Kong  5.43   1.40    1.29    1.03    0.524   0.246 0.291
## 5             97 Bhutan     5.08   0.796   1.34    0.527   0.541   0.364 0.171
## 6             98 Somalia    4.98   0       0.712   0.115   0.674   0.238 0.282
## 7            137 Sudan      4.14   0.605   1.24    0.312   0.016   0.134 0.082
## 8            142 Angola     3.80   0.73    1.12    0.269   0       0.079 0.061
## 9            150 Syria      3.46   0.689   0.382   0.539   0.088   0.376 0.144
## # … with abbreviated variable names ¹`Country or region`, ²`GDP per capita`,
## #   ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
```

```
X2018_clean2<- X2018_clean1%>% #Remove the values that do not match
  filter(!`Country or region`=="Qatar",
         !`Country or region`=="Belize",
         !`Country or region`=="Hong Kong",
         !`Country or region`=="Northern Cyprus",
         !`Country or region`=="Bhutan",
         !`Country or region`=="Somalia",
         !`Country or region`=="Angola",
         !`Country or region`=="Syria",
         !`Country or region`=="Sudan")

X2018_clean2 #Check it
```

```
## # A tibble: 147 × 9
##    `Overall rank` Countr…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##             <dbl> <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1              1 Finland   7.63    1.30    1.59   0.874   0.681   0.202 0.393
##  2              2 Norway    7.59    1.46    1.58   0.861   0.686   0.286 0.340
##  3              3 Denmark   7.56    1.35    1.59   0.868   0.683   0.284 0.408
##  4              4 Iceland   7.50    1.34    1.64   0.914   0.677   0.353 0.138
##  5              5 Switzer…  7.49    1.42    1.55   0.927   0.66    0.256 0.357
##  6              6 Netherl…  7.44    1.36    1.49   0.878   0.638   0.333 0.295
##  7              7 Canada    7.33    1.33    1.53   0.896   0.653   0.321 0.291
##  8              8 New Zea…  7.32    1.27    1.60   0.876   0.669   0.365 0.389
##  9              9 Sweden    7.31    1.36    1.50   0.913   0.659   0.285 0.383
## 10             10 Austral…  7.27    1.34    1.57   0.91    0.647   0.361 0.302
## # … with 137 more rows, and abbreviated variable names ¹`Country or region`,
## #   ²`GDP per capita`, ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows
```

```
  X2018_clean2%>% #Check that it worked
    anti_join(X2020_clean1, by = c("Country or region"="Country name"))
```

```
## # A tibble: 0 × 9
## # … with 9 variables: Overall rank <dbl>, Country or region <chr>, Score <dbl>,
## #   GDP per capita <dbl>, Social support <dbl>, Healthy life expectancy <dbl>,
## #   Freedom to make life choices <dbl>, Generosity <dbl>,
## #   Perceptions of corruption <chr>
## # ℹ Use `colnames()` to see all variable names
```

Cleaning 2018 against 2021

Anti-join was used to see which variables from the 2018 Happiness data set were missing from the 2021 Happiness data set. The countries from the 2018 Happiness data set that did not match were checked for spelling differences and recoded to make them consistent.

```
  #Cleaning 2018 to 2021
  X2018_clean2%>% #Check which values are in 2018 and not in 2021
    anti_join(X2021_clean1, by = c("Country or region"="Country name"))
```

```
## # A tibble: 4 × 9
##   `Overall rank` Country…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##            <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 1             38 Trinidad…  6.19   1.22    1.49    0.564   0.575   0.171 0.019
## 2            132 Democrat…  4.24   0.069   1.14    0.204   0.312   0.197 0.052
## 3            154 South Su…  3.25   0.337   0.608   0.177   0.112   0.224 0.106
## 4            155 Central …  3.08   0.024   0       0.01    0.305   0.218 0.038
## # … with abbreviated variable names ¹`Country or region`, ²`GDP per capita`,
## #   ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
```

```
X2018_clean3<- X2018_clean2%>% #Remove the values that do not match
    filter(!`Country or region`=="Trinidad And Tobago",
           !`Country or region`=="Democratic Republic Of The Congo",
           !`Country or region`=="South Sudan",
           !`Country or region`=="Central African Republic")

X2018_clean3 #Check it
```

```
## # A tibble: 143 × 9
##    `Overall rank` Countr…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##             <dbl> <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1              1 Finland   7.63    1.30    1.59   0.874   0.681   0.202 0.393
##  2              2 Norway    7.59    1.46    1.58   0.861   0.686   0.286 0.340
##  3              3 Denmark   7.56    1.35    1.59   0.868   0.683   0.284 0.408
##  4              4 Iceland   7.50    1.34    1.64   0.914   0.677   0.353 0.138
##  5              5 Switzer…  7.49    1.42    1.55   0.927   0.66    0.256 0.357
##  6              6 Netherl…  7.44    1.36    1.49   0.878   0.638   0.333 0.295
##  7              7 Canada    7.33    1.33    1.53   0.896   0.653   0.321 0.291
##  8              8 New Zea…  7.32    1.27    1.60   0.876   0.669   0.365 0.389
##  9              9 Sweden    7.31    1.36    1.50   0.913   0.659   0.285 0.383
## 10             10 Austral…  7.27    1.34    1.57   0.91    0.647   0.361 0.302
## # … with 133 more rows, and abbreviated variable names ¹`Country or region`,
## #   ²`GDP per capita`, ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows
```

```
  X2018_clean3%>% #Check that it worked
    anti_join(X2020_clean1, by = c("Country or region"="Country name"))
```

```
## # A tibble: 0 × 9
## # … with 9 variables: Overall rank <dbl>, Country or region <chr>, Score <dbl>,
## #   GDP per capita <dbl>, Social support <dbl>, Healthy life expectancy <dbl>,
## #   Freedom to make life choices <dbl>, Generosity <dbl>,
## #   Perceptions of corruption <chr>
## # ℹ Use `colnames()` to see all variable names
```

Cleaning 2020 against 2018

Anti-join was used to see which variables from the 2020 Happiness data set were missing from the 2018 Happiness data set. The countries from the 2020 Happiness data set that did not match were checked for spelling differences and recoded to make them consistent.

```
#Cleaning 2020 against 2018
  X2020_clean1%>% #Check which values are in 2020 and not in 2018
    anti_join(X2018_clean3, by = c("Country name"="Country or region"))
```

```
## # A tibble: 10 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Trinidad And… Latin …    6.19  0.114     6.42    5.97   10.3    0.915    63.5
##  2 North Cyprus  Wester…    5.54  0.0510    5.64    5.44   10.4    0.820    73.7
##  3 Hong Kong S.… East A…    5.51  0.0460    5.60    5.42   10.9    0.846    76.8
##  4 Maldives      South …    5.20  0.0720    5.34    5.06    9.52   0.913    70.6
##  5 Gambia        Sub-Sa…    4.75  0.0672    4.88    4.62    7.32   0.693    55.0
##  6 Democratic R… Sub-Sa…    4.31  0.109     4.52    4.10    6.69   0.672    52.9
##  7 Swaziland     Sub-Sa…    4.31  0.0715    4.45    4.17    9.16   0.770    51.2
##  8 Comoros       Sub-Sa…    4.29  0.0843    4.45    4.12    7.83   0.626    57.3
##  9 Central Afri… Sub-Sa…    3.48  0.115     3.70    3.25    6.63   0.319    45.2
## 10 South Sudan   Sub-Sa…    2.82  0.108     3.03    2.61    7.43   0.554    51
## # … with 11 more variables: `Freedom to make life choices` <dbl>,
## #   Generosity <dbl>, `Perceptions of corruption` <dbl>,
## #   `Ladder score in Dystopia` <dbl>, `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>,
## #   `Explained by: Generosity` <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

```
X2020_clean2 <- X2020_clean1%>%
    filter(!`Country name`=="Trinidad And Tobago",
           !`Country name`=="Democratic Republic Of The Congo",
           !`Country name`=="South Sudan",
           !`Country name`=="North Cyprus",
           !`Country name`=="Hong Kong S.A.R. of China",
           !`Country name`=="Maldives",
           !`Country name`=="Gambia",
           !`Country name`=="Central African Republic",
           !`Country name`=="Swaziland",
           !`Country name`=="Comoros")

  X2020_clean2 #Check it
```

```
## # A tibble: 143 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.81  0.0312    7.87    7.75    10.6   0.954    71.9
##  2 Denmark       Wester…    7.65  0.0335    7.71    7.58    10.8   0.956    72.4
##  3 Switzerland   Wester…    7.56  0.0350    7.63    7.49    11.0   0.943    74.1
##  4 Iceland       Wester…    7.50  0.0596    7.62    7.39    10.8   0.975    73
##  5 Norway        Wester…    7.49  0.0348    7.56    7.42    11.1   0.952    73.2
##  6 Netherlands   Wester…    7.45  0.0278    7.50    7.39    10.8   0.939    72.3
##  7 Sweden        Wester…    7.35  0.0362    7.42    7.28    10.8   0.926    72.6
##  8 New Zealand   North …    7.30  0.0395    7.38    7.22    10.5   0.949    73.2
##  9 Austria       Wester…    7.29  0.0334    7.36    7.23    10.7   0.928    73.0
## 10 Luxembourg    Wester…    7.24  0.0309    7.30    7.18    11.5   0.907    72.6
## # … with 133 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
  X2020_clean2%>% #Check that it worked
    anti_join(X2018_clean3, by = c("Country name"="Country or region"))
```

```
## # A tibble: 0 × 20
## # … with 20 variables: Country name <chr>, Regional indicator <chr>,
## #   Ladder score <dbl>, Standard error of ladder score <dbl>,
## #   upperwhisker <dbl>, lowerwhisker <dbl>, Logged GDP per capita <dbl>,
## #   Social support <dbl>, Healthy life expectancy <dbl>,
## #   Freedom to make life choices <dbl>, Generosity <dbl>,
## #   Perceptions of corruption <dbl>, Ladder score in Dystopia <dbl>,
## #   Explained by: Log GDP per capita <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

Checking 2020 against 2021

Anti-join was used to see which variables from the 2020 Happiness data set were missing from the 2021 Happiness data set. The countries from the 2020 Happiness data set that did not match were checked for spelling differences and recoded to make them consistent. In this case no renaming was needed!

```
#CLeaning 2020 against 2021
  X2020_clean2%>% #Check which values are in 2020 and not in 2021
    anti_join(X2021_clean1, by = "Country name")
```

```
## # A tibble: 0 × 20
## # … with 20 variables: Country name <chr>, Regional indicator <chr>,
## #   Ladder score <dbl>, Standard error of ladder score <dbl>,
## #   upperwhisker <dbl>, lowerwhisker <dbl>, Logged GDP per capita <dbl>,
## #   Social support <dbl>, Healthy life expectancy <dbl>,
## #   Freedom to make life choices <dbl>, Generosity <dbl>,
## #   Perceptions of corruption <dbl>, Ladder score in Dystopia <dbl>,
## #   Explained by: Log GDP per capita <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

```
#It's already clean!

#Make sure both data sets actually work and are not broken

  X2020_clean2
```

```
## # A tibble: 143 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.81  0.0312    7.87    7.75    10.6   0.954    71.9
##  2 Denmark       Wester…    7.65  0.0335    7.71    7.58    10.8   0.956    72.4
##  3 Switzerland   Wester…    7.56  0.0350    7.63    7.49    11.0   0.943    74.1
##  4 Iceland       Wester…    7.50  0.0596    7.62    7.39    10.8   0.975    73
##  5 Norway        Wester…    7.49  0.0348    7.56    7.42    11.1   0.952    73.2
##  6 Netherlands   Wester…    7.45  0.0278    7.50    7.39    10.8   0.939    72.3
##  7 Sweden        Wester…    7.35  0.0362    7.42    7.28    10.8   0.926    72.6
##  8 New Zealand   North …    7.30  0.0395    7.38    7.22    10.5   0.949    73.2
##  9 Austria       Wester…    7.29  0.0334    7.36    7.23    10.7   0.928    73.0
## 10 Luxembourg    Wester…    7.24  0.0309    7.30    7.18    11.5   0.907    72.6
## # … with 133 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
  X2021_clean1
```

```
## # A tibble: 149 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.84   0.032    7.90    7.78    10.8   0.954    72
##  2 Denmark       Wester…    7.62   0.035    7.69    7.55    10.9   0.954    72.7
##  3 Switzerland   Wester…    7.57   0.036    7.64    7.5     11.1   0.942    74.4
##  4 Iceland       Wester…    7.55   0.059    7.67    7.44    10.9   0.983    73
##  5 Netherlands   Wester…    7.46   0.027    7.52    7.41    10.9   0.942    72.4
##  6 Norway        Wester…    7.39   0.035    7.46    7.32    11.1   0.954    73.3
##  7 Sweden        Wester…    7.36   0.036    7.43    7.29    10.9   0.934    72.7
##  8 Luxembourg    Wester…    7.32   0.037    7.40    7.25    11.6   0.908    72.6
##  9 New Zealand   North …    7.28   0.04     7.36    7.20    10.6   0.948    73.4
## 10 Austria       Wester…    7.27   0.036    7.34    7.20    10.9   0.934    73.3
## # … with 139 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
#They work! Yay! Moving on...
```

Cleaning 2021 against 2018

Anti-join was used to see which variables from the 2021 Happiness data set were missing from the 2018 Happiness data set. The countries from the 2021 Happiness data set that did not match were checked for spelling differences and recoded to make them consistent.

```
#Cleaning 2021 against 2018
  X2021_clean1%>% #Check which values are in 2021 and not in 2018
    anti_join(X2018_clean3, by = c("Country name"="Country or region"))
```

```
## # A tibble: 6 × 20
##   `Country name` Regio…¹ Ladde…² Stand…³ upper…⁴ lower…⁵ Logge…⁶ Socia…⁷ Healt…⁸
##   <chr>          <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 North Cyprus   Wester…    5.54   0.051    5.64    5.44   10.6    0.82     73.9
## 2 Hong Kong S.A… East A…    5.48   0.049    5.57    5.38   11      0.836    76.8
## 3 Maldives       South …    5.20   0.072    5.34    5.06    9.83   0.913    70.6
## 4 Gambia         Sub-Sa…    5.05   0.089    5.22    4.88    7.69   0.69     55.2
## 5 Swaziland      Sub-Sa…    4.31   0.071    4.45    4.17    9.06   0.77     50.8
## 6 Comoros        Sub-Sa…    4.29   0.084    4.45    4.12    8.03   0.626    57.3
## # … with 11 more variables: `Freedom to make life choices` <dbl>,
## #   Generosity <dbl>, `Perceptions of corruption` <dbl>,
## #   `Ladder score in Dystopia` <dbl>, `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>,
## #   `Explained by: Generosity` <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

```
X2021_clean2 <- X2021_clean1%>%
  filter(!`Country name`=="North Cyprus",
         !`Country name`=="Hong Kong S.A.R. of China",
         !`Country name`=="Maldives",
         !`Country name`=="Gambia",
         !`Country name`=="Swaziland",
         !`Country name`=="Comoros")

X2021_clean2 #Check it
```

```
## # A tibble: 143 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.84   0.032    7.90    7.78    10.8   0.954    72
##  2 Denmark       Wester…    7.62   0.035    7.69    7.55    10.9   0.954    72.7
##  3 Switzerland   Wester…    7.57   0.036    7.64    7.5     11.1   0.942    74.4
##  4 Iceland       Wester…    7.55   0.059    7.67    7.44    10.9   0.983    73
##  5 Netherlands   Wester…    7.46   0.027    7.52    7.41    10.9   0.942    72.4
##  6 Norway        Wester…    7.39   0.035    7.46    7.32    11.1   0.954    73.3
##  7 Sweden        Wester…    7.36   0.036    7.43    7.29    10.9   0.934    72.7
##  8 Luxembourg    Wester…    7.32   0.037    7.40    7.25    11.6   0.908    72.6
##  9 New Zealand   North …    7.28   0.04     7.36    7.20    10.6   0.948    73.4
## 10 Austria       Wester…    7.27   0.036    7.34    7.20    10.9   0.934    73.3
## # … with 133 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2021_clean2%>% #Check that it worked
  anti_join(X2018_clean3, by = c("Country name"="Country or region"))
```

```
## # A tibble: 0 × 20
## # … with 20 variables: Country name <chr>, Regional indicator <chr>,
## #   Ladder score <dbl>, Standard error of ladder score <dbl>,
## #   upperwhisker <dbl>, lowerwhisker <dbl>, Logged GDP per capita <dbl>,
## #   Social support <dbl>, Healthy life expectancy <dbl>,
## #   Freedom to make life choices <dbl>, Generosity <dbl>,
## #   Perceptions of corruption <dbl>, Ladder score in Dystopia <dbl>,
## #   Explained by: Log GDP per capita <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

Cleaning 2021 against 2020

Anti-join was used to see which variables from the 2021 Happiness data set were missing from the 2020 Happiness data set. The countries from the 2021 Happiness data set that did not match were checked for spelling differences and recoded to make them consistent. In this case no renaming was needed!

```
#Cleaning 2021 against 2018
  X2021_clean2%>% #Check which values are in 2021 and not in 2020
    anti_join(X2020_clean2, by = "Country name")
```

```
## # A tibble: 0 × 20
## # … with 20 variables: Country name <chr>, Regional indicator <chr>,
## #   Ladder score <dbl>, Standard error of ladder score <dbl>,
## #   upperwhisker <dbl>, lowerwhisker <dbl>, Logged GDP per capita <dbl>,
## #   Social support <dbl>, Healthy life expectancy <dbl>,
## #   Freedom to make life choices <dbl>, Generosity <dbl>,
## #   Perceptions of corruption <dbl>, Ladder score in Dystopia <dbl>,
## #   Explained by: Log GDP per capita <dbl>, …
## # ℹ Use `colnames()` to see all variable names
```

```
#It's already clean!

#Make sure both data sets actually work and are not broken

  X2020_clean2
```

```
## # A tibble: 143 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.81  0.0312    7.87    7.75    10.6   0.954    71.9
##  2 Denmark       Wester…    7.65  0.0335    7.71    7.58    10.8   0.956    72.4
##  3 Switzerland   Wester…    7.56  0.0350    7.63    7.49    11.0   0.943    74.1
##  4 Iceland       Wester…    7.50  0.0596    7.62    7.39    10.8   0.975    73
##  5 Norway        Wester…    7.49  0.0348    7.56    7.42    11.1   0.952    73.2
##  6 Netherlands   Wester…    7.45  0.0278    7.50    7.39    10.8   0.939    72.3
##  7 Sweden        Wester…    7.35  0.0362    7.42    7.28    10.8   0.926    72.6
##  8 New Zealand   North …    7.30  0.0395    7.38    7.22    10.5   0.949    73.2
##  9 Austria       Wester…    7.29  0.0334    7.36    7.23    10.7   0.928    73.0
## 10 Luxembourg    Wester…    7.24  0.0309    7.30    7.18    11.5   0.907    72.6
## # … with 133 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2021_clean2
```

```
## # A tibble: 143 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.84   0.032    7.90    7.78    10.8   0.954    72
##  2 Denmark       Wester…    7.62   0.035    7.69    7.55    10.9   0.954    72.7
##  3 Switzerland   Wester…    7.57   0.036    7.64    7.5     11.1   0.942    74.4
##  4 Iceland       Wester…    7.55   0.059    7.67    7.44    10.9   0.983    73
##  5 Netherlands   Wester…    7.46   0.027    7.52    7.41    10.9   0.942    72.4
##  6 Norway        Wester…    7.39   0.035    7.46    7.32    11.1   0.954    73.3
##  7 Sweden        Wester…    7.36   0.036    7.43    7.29    10.9   0.934    72.7
##  8 Luxembourg    Wester…    7.32   0.037    7.40    7.25    11.6   0.908    72.6
##  9 New Zealand   North …    7.28   0.04     7.36    7.20    10.6   0.948    73.4
## 10 Austria       Wester…    7.27   0.036    7.34    7.20    10.9   0.934    73.3
## # … with 133 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
#They work! Yay! Moving on...
```

Let's take a look at our clean data sets. Notice they all have the same number of rows but not the same columns. Let's fix that.

```
X2018_clean3
```

```
## # A tibble: 143 × 9
##    `Overall rank` Countr…¹ Score GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##             <dbl> <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1              1 Finland   7.63    1.30    1.59   0.874   0.681   0.202 0.393
##  2              2 Norway    7.59    1.46    1.58   0.861   0.686   0.286 0.340
##  3              3 Denmark   7.56    1.35    1.59   0.868   0.683   0.284 0.408
##  4              4 Iceland   7.50    1.34    1.64   0.914   0.677   0.353 0.138
##  5              5 Switzer…  7.49    1.42    1.55   0.927   0.66    0.256 0.357
##  6              6 Netherl…  7.44    1.36    1.49   0.878   0.638   0.333 0.295
##  7              7 Canada    7.33    1.33    1.53   0.896   0.653   0.321 0.291
##  8              8 New Zea…  7.32    1.27    1.60   0.876   0.669   0.365 0.389
##  9              9 Sweden    7.31    1.36    1.50   0.913   0.659   0.285 0.383
## 10             10 Austral…  7.27    1.34    1.57   0.91    0.647   0.361 0.302
## # … with 133 more rows, and abbreviated variable names ¹`Country or region`,
## #   ²`GDP per capita`, ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows
```

```
X2020_clean2
```

```
## # A tibble: 143 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.81  0.0312    7.87    7.75    10.6   0.954    71.9
##  2 Denmark       Wester…    7.65  0.0335    7.71    7.58    10.8   0.956    72.4
##  3 Switzerland   Wester…    7.56  0.0350    7.63    7.49    11.0   0.943    74.1
##  4 Iceland       Wester…    7.50  0.0596    7.62    7.39    10.8   0.975    73
##  5 Norway        Wester…    7.49  0.0348    7.56    7.42    11.1   0.952    73.2
##  6 Netherlands   Wester…    7.45  0.0278    7.50    7.39    10.8   0.939    72.3
##  7 Sweden        Wester…    7.35  0.0362    7.42    7.28    10.8   0.926    72.6
##  8 New Zealand   North …    7.30  0.0395    7.38    7.22    10.5   0.949    73.2
##  9 Austria       Wester…    7.29  0.0334    7.36    7.23    10.7   0.928    73.0
## 10 Luxembourg    Wester…    7.24  0.0309    7.30    7.18    11.5   0.907    72.6
## # … with 133 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2021_clean2
```

```
## # A tibble: 143 × 20
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.84   0.032    7.90    7.78    10.8   0.954    72
##  2 Denmark       Wester…    7.62   0.035    7.69    7.55    10.9   0.954    72.7
##  3 Switzerland   Wester…    7.57   0.036    7.64    7.5     11.1   0.942    74.4
##  4 Iceland       Wester…    7.55   0.059    7.67    7.44    10.9   0.983    73
##  5 Netherlands   Wester…    7.46   0.027    7.52    7.41    10.9   0.942    72.4
##  6 Norway        Wester…    7.39   0.035    7.46    7.32    11.1   0.954    73.3
##  7 Sweden        Wester…    7.36   0.036    7.43    7.29    10.9   0.934    72.7
##  8 Luxembourg    Wester…    7.32   0.037    7.40    7.25    11.6   0.908    72.6
##  9 New Zealand   North …    7.28   0.04     7.36    7.20    10.6   0.948    73.4
## 10 Austria       Wester…    7.27   0.036    7.34    7.20    10.9   0.934    73.3
## # … with 133 more rows, 11 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Cleaning the columns

Adding the 'Year' column

The variable year was added to each cleaned Happiness data set in order to specify which year it came from.

```
#Add the variable year (This is useful later)
  X2018_clean3$Year <- c("2018")
  X2020_clean2$Year <- c("2020")
  X2021_clean2$Year <- c("2021")

#Check it
  X2018_clean3
```

```
## # A tibble: 143 × 10
##    Overall…¹ Count…² Score GDP p…³ Socia…⁴ Healt…⁵ Freed…⁶ Gener…⁷ Perce…⁸ Year
##        <dbl> <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1         1 Finland  7.63    1.30    1.59   0.874   0.681   0.202 0.393   2018
##  2         2 Norway   7.59    1.46    1.58   0.861   0.686   0.286 0.340   2018
##  3         3 Denmark  7.56    1.35    1.59   0.868   0.683   0.284 0.408   2018
##  4         4 Iceland  7.50    1.34    1.64   0.914   0.677   0.353 0.138   2018
##  5         5 Switze…  7.49    1.42    1.55   0.927   0.66    0.256 0.357   2018
##  6         6 Nether…  7.44    1.36    1.49   0.878   0.638   0.333 0.295   2018
##  7         7 Canada   7.33    1.33    1.53   0.896   0.653   0.321 0.291   2018
##  8         8 New Ze…  7.32    1.27    1.60   0.876   0.669   0.365 0.389   2018
##  9         9 Sweden   7.31    1.36    1.50   0.913   0.659   0.285 0.383   2018
## 10        10 Austra…  7.27    1.34    1.57   0.91    0.647   0.361 0.302   2018
## # … with 133 more rows, and abbreviated variable names ¹`Overall rank`,
## #   ²`Country or region`, ³`GDP per capita`, ⁴`Social support`,
## #   ⁵`Healthy life expectancy`, ⁶`Freedom to make life choices`, ⁷Generosity,
## #   ⁸`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows
```

```
X2020_clean2
```

```
## # A tibble: 143 × 21
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.81  0.0312    7.87    7.75    10.6   0.954    71.9
##  2 Denmark       Wester…    7.65  0.0335    7.71    7.58    10.8   0.956    72.4
##  3 Switzerland   Wester…    7.56  0.0350    7.63    7.49    11.0   0.943    74.1
##  4 Iceland       Wester…    7.50  0.0596    7.62    7.39    10.8   0.975    73
##  5 Norway        Wester…    7.49  0.0348    7.56    7.42    11.1   0.952    73.2
##  6 Netherlands   Wester…    7.45  0.0278    7.50    7.39    10.8   0.939    72.3
##  7 Sweden        Wester…    7.35  0.0362    7.42    7.28    10.8   0.926    72.6
##  8 New Zealand   North …    7.30  0.0395    7.38    7.22    10.5   0.949    73.2
##  9 Austria       Wester…    7.29  0.0334    7.36    7.23    10.7   0.928    73.0
## 10 Luxembourg    Wester…    7.24  0.0309    7.30    7.18    11.5   0.907    72.6
## # … with 133 more rows, 12 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2021_clean2
```

```
## # A tibble: 143 × 21
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Finland       Wester…    7.84   0.032    7.90    7.78    10.8   0.954    72
##  2 Denmark       Wester…    7.62   0.035    7.69    7.55    10.9   0.954    72.7
##  3 Switzerland   Wester…    7.57   0.036    7.64    7.5     11.1   0.942    74.4
##  4 Iceland       Wester…    7.55   0.059    7.67    7.44    10.9   0.983    73
##  5 Netherlands   Wester…    7.46   0.027    7.52    7.41    10.9   0.942    72.4
##  6 Norway        Wester…    7.39   0.035    7.46    7.32    11.1   0.954    73.3
##  7 Sweden        Wester…    7.36   0.036    7.43    7.29    10.9   0.934    72.7
##  8 Luxembourg    Wester…    7.32   0.037    7.40    7.25    11.6   0.908    72.6
##  9 New Zealand   North …    7.28   0.04     7.36    7.20    10.6   0.948    73.4
## 10 Austria       Wester…    7.27   0.036    7.34    7.20    10.9   0.934    73.3
## # … with 133 more rows, 12 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Adding the column Regional Indicator to 2018

The Happiness data sets were all placed in alphabetical order using arrange() to ensure that everything lines up properly. Then the 'Regional_indicator' variable from the 2020/2021 data sets was added to the 2018 data set. Since they are in the same order all of the information should align.

```
#Put the 2020 data in alphabetical order so they align and save it as a new data set
X2020_alpha <- X2020_clean2%>%
    arrange(`Country name`)

X2020_alpha #Look at it
```

```
## # A tibble: 143 × 21
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Afghanistan   South …    2.57  0.0313    2.63    2.51    7.46   0.470    52.6
##  2 Albania       Centra…    4.88  0.0561    4.99    4.77    9.42   0.671    68.7
##  3 Algeria       Middle…    5.01  0.0442    5.09    4.92    9.54   0.803    65.9
##  4 Argentina     Latin …    5.97  0.0534    6.08    5.87    9.81   0.901    68.8
##  5 Armenia       Common…    4.68  0.0586    4.79    4.56    9.10   0.757    66.8
##  6 Australia     North …    7.22  0.0418    7.30    7.14   10.7    0.945    73.6
##  7 Austria       Wester…    7.29  0.0334    7.36    7.23   10.7    0.928    73.0
##  8 Azerbaijan    Common…    5.16  0.0342    5.23    5.10    9.69   0.819    65.5
##  9 Bahrain       Middle…    6.23  0.0819    6.39    6.07   10.7    0.876    68.5
## 10 Bangladesh    South …    4.83  0.0401    4.91    4.75    8.29   0.687    64.5
## # … with 133 more rows, 12 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2021_alpha <- X2021_clean2%>%
    arrange(`Country name`)

X2021_alpha #Look at it
```

```
## # A tibble: 143 × 21
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Afghanistan   South …    2.52   0.038    2.60    2.45    7.70   0.463    52.5
##  2 Albania       Centra…    5.12   0.059    5.23    5.00    9.52   0.697    69.0
##  3 Algeria       Middle…    4.89   0.053    4.99    4.78    9.34   0.802    66.0
##  4 Argentina     Latin …    5.93   0.056    6.04    5.82    9.96   0.898    69
##  5 Armenia       Common…    5.28   0.058    5.40    5.17    9.49   0.799    67.1
##  6 Australia     North …    7.18   0.041    7.26    7.10   10.8    0.94     73.9
##  7 Austria       Wester…    7.27   0.036    7.34    7.20   10.9    0.934    73.3
##  8 Azerbaijan    Common…    5.17   0.04     5.25    5.09    9.57   0.836    65.7
##  9 Bahrain       Middle…    6.65   0.068    6.78    6.51   10.7    0.862    69.5
## 10 Bangladesh    South …    5.03   0.046    5.12    4.93    8.45   0.693    64.8
## # … with 133 more rows, 12 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
#Put the 2018 data in alphabetical order so they align and save it as a new data set
X2018_alpha <- X2018_clean3%>%
    arrange(`Country or region`)

X2018_alpha #Check it
```

```
## # A tibble: 143 × 10
##    Overall…¹ Count…² Score GDP p…³ Socia…⁴ Healt…⁵ Freed…⁶ Gener…⁷ Perce…⁸ Year
##        <dbl> <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1       145 Afghan…  3.63   0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2       112 Albania  4.59   0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3        84 Algeria  5.30   0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4        29 Argent…  6.39   1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5       129 Armenia  4.32   0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6        10 Austra…  7.27   1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7        12 Austria  7.14   1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8        87 Azerba…  5.20   1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9        43 Bahrain  6.10   1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10       115 Bangla…  4.5    0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 133 more rows, and abbreviated variable names ¹`Overall rank`,
## #   ²`Country or region`, ³`GDP per capita`, ⁴`Social support`,
## #   ⁵`Healthy life expectancy`, ⁶`Freedom to make life choices`, ⁷Generosity,
## #   ⁸`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows
```

```
#Add the Regional Indicator variable from 2020 to 2018
X2018_regional<-X2018_alpha%>%
  mutate(`Regional indicator`=X2020_alpha$`Regional indicator`)

X2018_regional #Check it
```

```
## # A tibble: 143 × 11
##    Overall…¹ Count…² Score GDP p…³ Socia…⁴ Healt…⁵ Freed…⁶ Gener…⁷ Perce…⁸ Year
##        <dbl> <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1       145 Afghan…  3.63   0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2       112 Albania  4.59   0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3        84 Algeria  5.30   0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4        29 Argent…  6.39   1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5       129 Armenia  4.32   0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6        10 Austra…  7.27   1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7        12 Austria  7.14   1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8        87 Azerba…  5.20   1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9        43 Bahrain  6.10   1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10       115 Bangla…  4.5    0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 133 more rows, 1 more variable: `Regional indicator` <chr>, and
## #   abbreviated variable names ¹`Overall rank`, ²`Country or region`,
## #   ³`GDP per capita`, ⁴`Social support`, ⁵`Healthy life expectancy`,
## #   ⁶`Freedom to make life choices`, ⁷Generosity, ⁸`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Rename and select the columns for 2018

Using select() the variables of interest were taken from the 2018 Happiness data set to remain consistent across all of the Happiness data sets and the columns were for convenience and clarity.

```
  X2018_regional #Look at the data set for reference
```

```
## # A tibble: 143 × 11
##    Overall…¹ Count…² Score GDP p…³ Socia…⁴ Healt…⁵ Freed…⁶ Gener…⁷ Perce…⁸ Year
##        <dbl> <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1       145 Afghan…  3.63   0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2       112 Albania  4.59   0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3        84 Algeria  5.30   0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4        29 Argent…  6.39   1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5       129 Armenia  4.32   0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6        10 Austra…  7.27   1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7        12 Austria  7.14   1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8        87 Azerba…  5.20   1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9        43 Bahrain  6.10   1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10       115 Bangla…  4.5    0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 133 more rows, 1 more variable: `Regional indicator` <chr>, and
## #   abbreviated variable names ¹`Overall rank`, ²`Country or region`,
## #   ³`GDP per capita`, ⁴`Social support`, ⁵`Healthy life expectancy`,
## #   ⁶`Freedom to make life choices`, ⁷Generosity, ⁸`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2018_test1 <- X2018_regional%>% #Rename
  select(2,11,3,4,5,6,7,8,9,10)%>% #Select the columns you want
  rename(Country = `Country or region`,
       "Social_support" = `Social support`,
       "Life_expectancy" = `Healthy life expectancy`,
       Freedom = `Freedom to make life choices`,
       Corruption =`Perceptions of corruption`,
       GDP =`GDP per capita`,
       "Regional_indicator"=`Regional indicator`) #Rename the columns

X2018_test1 #Check it
```

```
## # A tibble: 143 × 10
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2 Albania     Centra…  4.59 0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3 Algeria     Middle…  5.30 0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4 Argentina   Latin …  6.39 1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5 Armenia     Common…  4.32 0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6 Australia   North …  7.27 1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7 Austria     Wester…  7.14 1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8 Azerbaijan  Common…  5.20 1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9 Bahrain     Middle…  6.10 1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10 Bangladesh  South …  4.5  0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 133 more rows, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows
```

Rename and select the columns for 2020

Using select() the variables of interest were taken from the 2020 Happiness data set to remain consistent across all of the Happiness data sets and the columns were for convenience and clarity.

```
X2020_alpha #Look at the data set for reference
```

```
## # A tibble: 143 × 21
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Afghanistan   South …    2.57  0.0313    2.63    2.51    7.46   0.470    52.6
##  2 Albania       Centra…    4.88  0.0561    4.99    4.77    9.42   0.671    68.7
##  3 Algeria       Middle…    5.01  0.0442    5.09    4.92    9.54   0.803    65.9
##  4 Argentina     Latin …    5.97  0.0534    6.08    5.87    9.81   0.901    68.8
##  5 Armenia       Common…    4.68  0.0586    4.79    4.56    9.10   0.757    66.8
##  6 Australia     North …    7.22  0.0418    7.30    7.14   10.7    0.945    73.6
##  7 Austria       Wester…    7.29  0.0334    7.36    7.23   10.7    0.928    73.0
##  8 Azerbaijan    Common…    5.16  0.0342    5.23    5.10    9.69   0.819    65.5
##  9 Bahrain       Middle…    6.23  0.0819    6.39    6.07   10.7    0.876    68.5
## 10 Bangladesh    South …    4.83  0.0401    4.91    4.75    8.29   0.687    64.5
## # … with 133 more rows, 12 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2020_test1 <- X2020_alpha%>% #Rename
    select(1,2,3,14,15,16,17,18,19,21)%>% #Select the columns you want
    rename(Country = `Country name`,
          Score = `Ladder score`,
          GDP = `Explained by: Log GDP per capita`,
          "Social_support" = `Explained by: Social support`,
          "Life_expectancy"=`Explained by: Healthy life expectancy`,
          Freedom = `Explained by: Freedom to make life choices`,
          Generosity = `Explained by: Generosity`,
          Corruption = `Explained by: Perceptions of corruption`,
          "Regional_indicator"=`Regional indicator`) #Rename the columns

  X2020_test1 #Check it
```

```
## # A tibble: 143 × 10
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1 Afghanistan South …  2.57 0.301   0.356   0.266  0       0.135  0.00123 2020
##  2 Albania     Centra…  4.88 0.907   0.830   0.846  0.462   0.171  0.0254  2020
##  3 Algeria     Middle…  5.01 0.944   1.14    0.745  0.0839  0.119  0.129   2020
##  4 Argentina   Latin …  5.97 1.03    1.37    0.850  0.521   0.0701 0.0604  2020
##  5 Armenia     Common…  4.68 0.808   1.03    0.776  0.378   0.107  0.105   2020
##  6 Australia   North …  7.22 1.31    1.48    1.02   0.622   0.325  0.336   2020
##  7 Austria     Wester…  7.29 1.32    1.44    1.00   0.603   0.256  0.281   2020
##  8 Azerbaijan  Common…  5.16 0.990   1.18    0.731  0.468   0.0401 0.247   2020
##  9 Bahrain     Middle…  6.23 1.30    1.32    0.839  0.610   0.287  0.127   2020
## 10 Bangladesh  South …  4.83 0.556   0.869   0.695  0.604   0.177  0.177   2020
## # … with 133 more rows, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows
```

Rename and select columns for 2021

Using select() the variables of interest were taken from the 2021 Happiness data set to remain consistent across all of the Happiness data sets and the columns were for convenience and clarity.

```
X2021_alpha #Look at the data set for reference
```

```
## # A tibble: 143 × 21
##    Country nam…¹ Regio…² Ladde…³ Stand…⁴ upper…⁵ lower…⁶ Logge…⁷ Socia…⁸ Healt…⁹
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Afghanistan   South …    2.52   0.038    2.60    2.45    7.70   0.463    52.5
##  2 Albania       Centra…    5.12   0.059    5.23    5.00    9.52   0.697    69.0
##  3 Algeria       Middle…    4.89   0.053    4.99    4.78    9.34   0.802    66.0
##  4 Argentina     Latin …    5.93   0.056    6.04    5.82    9.96   0.898    69
##  5 Armenia       Common…    5.28   0.058    5.40    5.17    9.49   0.799    67.1
##  6 Australia     North …    7.18   0.041    7.26    7.10   10.8    0.94     73.9
##  7 Austria       Wester…    7.27   0.036    7.34    7.20   10.9    0.934    73.3
##  8 Azerbaijan    Common…    5.17   0.04     5.25    5.09    9.57   0.836    65.7
##  9 Bahrain       Middle…    6.65   0.068    6.78    6.51   10.7    0.862    69.5
## 10 Bangladesh    South …    5.03   0.046    5.12    4.93    8.45   0.693    64.8
## # … with 133 more rows, 12 more variables:
## #   `Freedom to make life choices` <dbl>, Generosity <dbl>,
## #   `Perceptions of corruption` <dbl>, `Ladder score in Dystopia` <dbl>,
## #   `Explained by: Log GDP per capita` <dbl>,
## #   `Explained by: Social support` <dbl>,
## #   `Explained by: Healthy life expectancy` <dbl>,
## #   `Explained by: Freedom to make life choices` <dbl>, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
X2021_test1 <- X2021_alpha%>% #Rename
  select(1,2,3,14,15,16,17,18,19,21)%>% #Select the columns you want
  rename(Country = `Country name`,
         Score = `Ladder score`,
         GDP = `Explained by: Log GDP per capita`,
         "Social_support" = `Explained by: Social support`,
         "Life_expectancy"=`Explained by: Healthy life expectancy`,
         Freedom = `Explained by: Freedom to make life choices`,
         Generosity = `Explained by: Generosity`,
         Corruption = `Explained by: Perceptions of corruption`,
         "Regional_indicator"=`Regional indicator`) #Rename the columns

X2021_test1 #Check it
```

```
## # A tibble: 143 × 10
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1 Afghanistan South …  2.52 0.37    0       0.126   0       0.122   0.01  2021
##  2 Albania     Centra…  5.12 1.01    0.529   0.646   0.491   0.168   0.024 2021
##  3 Algeria     Middle…  4.89 0.946   0.765   0.552   0.119   0.144   0.12  2021
##  4 Argentina   Latin …  5.93 1.16    0.98    0.646   0.544   0.069   0.067 2021
##  5 Armenia     Common…  5.28 0.996   0.758   0.585   0.54    0.079   0.198 2021
##  6 Australia   North …  7.18 1.45    1.08    0.801   0.647   0.291   0.317 2021
##  7 Austria     Wester…  7.27 1.49    1.06    0.782   0.64    0.215   0.292 2021
##  8 Azerbaijan  Common…  5.17 1.02    0.841   0.541   0.526   0.043   0.276 2021
##  9 Bahrain     Middle…  6.65 1.41    0.899   0.662   0.661   0.246   0.139 2021
## 10 Bangladesh  South …  5.03 0.635   0.52    0.514   0.603   0.161   0.164 2021
## # … with 133 more rows, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows
```

3. Joining/Merging

General Information

Total observations in each dataset: Covid Daily: 184,260 observations Covid Sum: 226 observations Happiness 2018: 156 observations Happiness 2020: 153 observations Happiness 2021: 149 observations

Unique IDs in each dataset (5): X2018: Overall rank, Country or region, Score, GDP per capita, Social support, Health life expectancy, Freedom to make life choices, Generosity, Perceptions of corruption

X2020: Country name, Regional indicator, Ladder score, Standard error of ladder score, upperwhisker, lowerwhisker, Logged GDP per capita, Social support, Healthy life expectancy, Freedom to make life choices, Generosity, Perceptions of corruption, Ladder score in Dystopia, Explained by: Log GDP per capita, Explained by: Social support, Explained by: Healthy life expectancy, Explained by: Freedom to make life choices, Explained by: Generosity, Explained by: Perceptions of corruption, Dystopia + residual

X2021: Country name, Regional indicator, Ladder score, Standard error of ladder score, upperwhisker, lowerwhisker, Logged GDP per capita, Social support, Healthy life expectancy, Freedom to make life choices, Generosity, Perceptions of corruption, Ladder score in Dystopia, Explained by: Log GDP per capita, Explained by: Social support, Explained by: Healthy life expectancy, Explained by: Freedom to make life choices, Explained by: Generosity, Explained by: Perceptions of corruption, Dystopia + residual

covid_daily: date, country, cumulative_total_cases, daily_new_cases, active_cases, cumulative_total_deaths, daily_new_deaths

covid_sum: country, continent, total_confirmed, total_deaths, total_recovered, active_cases, serious_or_critical, total_cases_per_1m_population, total_tests, total_tests_per_1m_population, population

IDs that appear in one dataset but not the other: Overall rank (2018), date (covid_daily), cumulative_total_cases, daily_new_cases, active_cases, daily_new_deaths, total_confirmed, total_recovered, serious_or_critical, total_cases_per_1m_population, total_tests, total_tests_per_1m_population, population

IDs in common (in all datasets): Country

IDs that may have been left out: Dystopia + residual, Overall rank, Logged GDP per capita, Social support, Healthy life expectancy, Freedom to make life choices, Generosity, Perceptions of corruption, Ladder score in Dystopia, upperwhisker, lowerwhisker, Standard error of ladder score

117 observations in total were removed when joining the data sets, some issues with excluding these data sets could be that we are narrowing the scope of our analysis and are examining fewer countries.

Joining/Merging Begins

Combine the data sets into one

rbind() was chosen to combine all three data sets in order to avoid the x y function that occurs when non key variables are repeated when data sets are joined using a join function. This rbind() allows us to keep all of the columns together, it is more clear and easier to read. I found this code at the following link: https://www.tutorialkart.com/r-tutorial/r-combine-data-frames-with-same-column-names/# (https://www.tutorialkart.com/r-tutorial/r-combine-data-frames-with-same-column-names/#):~:text=R%20%E2%80%93%20Combine%20Data%20Frames%20with%20Same%20Column%20Names&text=To%20combine%20two%20data%20frames,two%20d

```
#Combine 2018 and 2020
X2018_2020 <- rbind(X2018_test1, X2020_test1)

X2018_2020 #Check it
```

```
## # A tibble: 286 × 10
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2 Albania     Centra…  4.59 0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3 Algeria     Middle…  5.30 0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4 Argentina   Latin …  6.39 1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5 Armenia     Common…  4.32 0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6 Australia   North …  7.27 1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7 Austria     Wester…  7.14 1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8 Azerbaijan  Common…  5.20 1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9 Bahrain     Middle…  6.10 1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10 Bangladesh  South …  4.5  0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 276 more rows, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows
```

```
#Combine the previous combination to 2021
X2018_2020_2021 <- rbind(X2018_2020, X2021_test1)

X2018_2020_2021 #Check it
```

```
## # A tibble: 429 × 10
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2 Albania     Centra…  4.59 0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3 Algeria     Middle…  5.30 0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4 Argentina   Latin …  6.39 1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5 Armenia     Common…  4.32 0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6 Australia   North …  7.27 1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7 Austria     Wester…  7.14 1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8 Azerbaijan  Common…  5.20 1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9 Bahrain     Middle…  6.10 1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10 Bangladesh  South …  4.5  0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 419 more rows, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows
```

Cleaning the Daily Covid Data

Cleaning the Daily Covid data

```
X2018_2020_2021%>% #Check what is in the Happiness Report that is not contained in Covid daily
    anti_join(covid_daily, by = c("Country"="country"))
```

```
## # A tibble: 21 × 10
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 Ivory Coast Sub-Sa…  4.67 0.541   0.872   0.08    0.467  0.146  0.103   2018
##  2 Kosovo      Centra…  5.66 0.855   1.23    0.578   0.448  0.274  0.023   2018
##  3 Palestinia… Middle…  4.74 0.642   1.22    0.602   0.266  0.086  0.076   2018
##  4 Turkmenist… Common…  5.64 1.02    1.53    0.517   0.417  0.199  0.037   2018
##  5 United Kin… Wester…  7.19 1.24    1.43    0.888   0.464  0.262  0.082   2018
##  6 United Sta… North …  6.89 1.40    1.47    0.819   0.547  0.291  0.133   2018
##  7 Vietnam     Southe…  5.10 0.715   1.36    0.702   0.618  0.177  0.079   2018
##  8 Ivory Coast Sub-Sa…  5.23 0.537   0.800   0.155   0.397  0.170  0.0934… 2020
##  9 Kosovo      Centra…  6.33 0.840   1.18    0.673   0.557  0.325  0.0085… 2020
## 10 Palestinia… Middle…  4.55 0.588   1.19    0.614   0.299  0.0918 0.0719… 2020
## # … with 11 more rows, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows
```

```
covid_daily_split <- covid_daily%>% #Rename
  separate(date, into = c("year","month","day"))%>% #Separate the date variable
  rename(Country = country)%>% #Rename
  mutate(Country=recode(Country, 'USA'='United States', 'UK'='United Kingdom', 'Viet Nam'='Vietnam',
            'State Of Palestine'='Palestinian Territories','Cote D Ivoire'='Ivory Coast'))
            #This code renames countries so they match

X2018_2020_2021 <- X2018_2020_2021%>% #Update
  filter(!Country=="Kosovo",
         !Country=="Turkmenistan")#Remove these values

X2018_2020_2021%>% #Check that it worked
    anti_join(covid_daily_split, by = "Country")
```

```
## # A tibble: 0 × 10
## # … with 10 variables: Country <chr>, Regional_indicator <chr>, Score <dbl>,
## #   GDP <dbl>, Social_support <dbl>, Life_expectancy <dbl>, Freedom <dbl>,
## #   Generosity <dbl>, Corruption <chr>, Year <chr>
## # ℹ Use `colnames()` to see all variable names
```

```
covid_daily_split #Check it
```

```
## # A tibble: 184,260 × 9
##    year  month day   Country     cumulative_to…¹ daily…² activ…³ cumul…⁴ daily…⁵
##    <chr> <chr> <chr> <chr>                 <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 2020  2     15    Afghanistan               0      NA       0       0      NA
##  2 2020  2     16    Afghanistan               0      NA       0       0      NA
##  3 2020  2     17    Afghanistan               0      NA       0       0      NA
##  4 2020  2     18    Afghanistan               0      NA       0       0      NA
##  5 2020  2     19    Afghanistan               0      NA       0       0      NA
##  6 2020  2     20    Afghanistan               0      NA       0       0      NA
##  7 2020  2     21    Afghanistan               0      NA       0       0      NA
##  8 2020  2     22    Afghanistan               0      NA       0       0      NA
##  9 2020  2     23    Afghanistan               0      NA       0       0      NA
## 10 2020  2     24    Afghanistan               1      NA       1       0      NA
## # … with 184,250 more rows, and abbreviated variable names
## #   ¹cumulative_total_cases, ²daily_new_cases, ³active_cases,
## #   ⁴cumulative_total_deaths, ⁵daily_new_deaths
## # ℹ Use `print(n = ...)` to see more rows
```

4. Wrangling

Make some new data sets to analyze the data in different ways.

Numeric Summary Statistics

Monthly covid data for each year

The functions group_by(), filter, and summarize are used were used to find the total number of Covid cases per country for each month across all three years separating them into three separate data sets.

The monthly covid data shows that overall the new cases per month (cases) are greater in 2021 than in 2020 across a majority of the countries in the data set.

```
#Per month for the year 2018
covid_monthly_view_2018 <- covid_daily_split%>% #Rename
  group_by(Country,month,year)%>% #Group
  filter(year=="2018")%>% #Filter year 2018
  summarize(new_cases_per_month = sum(daily_new_cases, na.rm = T),
            new_deaths_per_month = sum(daily_new_deaths, na.rm=T)) #Find monthly data

covid_monthly_view_2018 #Check it
```

```
## # A tibble: 0 × 5
## # Groups:   Country, month [0]
## # … with 5 variables: Country <chr>, month <chr>, year <chr>,
## #   new_cases_per_month <dbl>, new_deaths_per_month <dbl>
## # ℹ Use `colnames()` to see all variable names
```

```
#Per month for the year 2020
covid_monthly_view_2020 <- covid_daily_split%>% #Rename
  group_by(Country,month,year)%>% #Group
  filter(year=="2020")%>% #Filter year 2020
  summarize(new_cases_per_month = sum(daily_new_cases, na.rm = T),
            new_deaths_per_month = sum(daily_new_deaths, na.rm=T)) #Find monthly data

covid_monthly_view_2020 #Check it
```

```
## # A tibble: 2,476 × 5
## # Groups:   Country, month [2,476]
##    Country     month year  new_cases_per_month new_deaths_per_month
##    <chr>       <chr> <chr>               <dbl>                <dbl>
##  1 Afghanistan 10    2020                 2157                   78
##  2 Afghanistan 11    2020                 5073                  238
##  3 Afghanistan 12    2020                 6015                  427
##  4 Afghanistan 2     2020                    0                    0
##  5 Afghanistan 3     2020                  173                    4
##  6 Afghanistan 4     2020                 1997                   60
##  7 Afghanistan 5     2020                13034                  193
##  8 Afghanistan 6     2020                16312                  489
##  9 Afghanistan 7     2020                 5158                  526
## 10 Afghanistan 8     2020                 1490                  130
## # … with 2,466 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

```
#Per month for the year 2021
covid_monthly_view_2021 <- covid_daily_split%>% #Rename
  group_by(Country,month,year)%>% #Group
  filter(year=="2021")%>% #Filter year 2021
  summarize(new_cases_per_month = sum(daily_new_cases, na.rm = T),
            new_deaths_per_month = sum(daily_new_deaths, na.rm=T)) #Find monthly data

covid_monthly_view_2021 #Check it
```

```
## # A tibble: 2,693 × 5
## # Groups:   Country, month [2,693]
##    Country     month year  new_cases_per_month new_deaths_per_month
##    <chr>       <chr> <chr>               <dbl>                <dbl>
##  1 Afghanistan 1     2021                 2546                  203
##  2 Afghanistan 10    2021                 1059                   74
##  3 Afghanistan 11    2021                 1053                   28
##  4 Afghanistan 12    2021                  795                   48
##  5 Afghanistan 2     2021                  674                   40
##  6 Afghanistan 3     2021                  784                   45
##  7 Afghanistan 4     2021                 3425                  142
##  8 Afghanistan 5     2021                13039                  343
##  9 Afghanistan 6     2021                47235                 1988
## 10 Afghanistan 7     2021                27285                 1775
## # … with 2,683 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

Yearly Covid Data

The functions group_by(), filter, and summarize are used were used to find the total number of Covid cases per country for each year across all three years in the same data set. The yearly covid data shows that there are significantly more covid cases (cases) and deaths (deceased individuals) in the year 2021 compared to 2020 for the majority of the countries in the data set.

```
covid_yearly_view <- covid_daily_split%>% #Rename
  filter(year %in% c(2020,2021))%>% #Filter year
  group_by(Country,year)%>% #Group
  summarize(new_cases_per_year = sum(daily_new_cases, na.rm = T),
            new_deaths_per_year = sum(daily_new_deaths, na.rm = T)) #Find yearly data

covid_yearly_view #Check it
```

```
## # A tibble: 450 × 4
## # Groups:   Country [226]
##    Country     year  new_cases_per_year new_deaths_per_year
##    <chr>       <chr>              <dbl>               <dbl>
##  1 Afghanistan 2020               52512                2201
##  2 Afghanistan 2021              105585                5155
##  3 Albania     2020               58314                1180
##  4 Albania     2021              151908                2036
##  5 Algeria     2020               99609                2756
##  6 Algeria     2021              118822                3520
##  7 Andorra     2020                8048                  84
##  8 Andorra     2021               15691                  56
##  9 Angola      2020               17552                 403
## 10 Angola      2021               64040                1365
## # … with 440 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

Sort by region

The functions group_by() and summarize were used to find the average value for all seven numeric variables for each region for each year. Europe, Middle East/Northern Africa, and the Americas/ANZ show higher Happiness scores and much higher GDP. West Asia and Sub-saharan Africa show the lowest happiness scores and GDP. Europe, East Asia, and North America/ANZ show the highest life expectancy, and Sub-saharan Africa showed the lowest. The highest sense of freedom was shown by Southeast Asia and the lowest by the Middle East and Northern Africa. The highest generosity was shown by North America/ANZ and the least by Central/Eastern Europe. The highest social support was recorded in North America/ANZ and the lowest by South Asia. The highest sense of government corruption was felt in North America/ANZ, and the lowest in Central/Eastern Europe.

```
X2018_2020_2021%>%
  distinct(Regional_indicator) #Look at the different regions
```

```
## # A tibble: 10 × 1
##    Regional_indicator
##    <chr>
##  1 South Asia
##  2 Central and Eastern Europe
##  3 Middle East and North Africa
##  4 Latin America and Caribbean
##  5 Commonwealth of Independent States
##  6 North America and ANZ
##  7 Western Europe
##  8 Sub-Saharan Africa
##  9 Southeast Asia
## 10 East Asia
```

```
X2018_2020_2021_regional_avg <- X2018_2020_2021%>% #Rename
  na_if("N/A")%>% #Fix the N/As
  group_by(Regional_indicator)%>% #Group by
  summarize(average_score = mean(Score, na.rm = T), #Find the average score
            average_GDP = mean(GDP, na.rm=T), #Find the average GDP
            average_life_expectancy = mean(Life_expectancy,na.rm=T), #Find the average life expectancy
            average_freedom = mean(Freedom,na.rm=T), #Find the average freedom
            average_generosity = mean(Generosity,na.rm=T), #Find the average generosity
            average_social_support = mean(Social_support,na.rm=T), #Find the average social support
            average_corruption = mean(as.numeric(Corruption),na.rm=T)) #Find the average corruption

X2018_2020_2021_regional_avg #Try it out
```

```
## # A tibble: 10 × 8
##    Regional_indicator    avera…¹ avera…² avera…³ avera…⁴ avera…⁵ avera…⁶ avera…⁷
##    <chr>                   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Central and Eastern …    5.81   1.14    0.739   0.449   0.117   1.21   0.0533
##  2 Commonwealth of Inde…    5.36   0.874   0.621   0.454   0.150   1.14   0.124
##  3 East Asia                5.78   1.19    0.803   0.450   0.136   1.18   0.108
##  4 Latin America and Ca…    5.94   0.901   0.677   0.521   0.144   1.15   0.0877
##  5 Middle East and Nort…    5.26   1.04    0.637   0.392   0.141   1.01   0.115
##  6 North America and ANZ    7.17   1.37    0.866   0.623   0.303   1.35   0.306
##  7 South Asia               4.40   0.612   0.490   0.441   0.221   0.739  0.0921
##  8 Southeast Asia           5.37   0.919   0.608   0.617   0.302   1.08   0.142
##  9 Sub-Saharan Africa       4.40   0.476   0.296   0.419   0.185   0.797  0.102
## 10 Western Europe           6.93   1.37    0.887   0.566   0.212   1.31   0.250
## # … with abbreviated variable names ¹average_score, ²average_GDP,
## #   ³average_life_expectancy, ⁴average_freedom, ⁵average_generosity,
## #   ⁶average_social_support, ⁷average_corruption
```

Average Happiness by Country

The functions group_by() and summarize() were used to find the average happiness score by country from all three years. This data shows that the top three happiest countries on average are Finland, Denmark and Switzerland, and the three unhappiest countries on average are Afghanistan, Rwanda, and Zimbabwe.

```
X2018_2020_2021_avg_happiness_by_country <- X2018_2020_2021%>% #Rename
  group_by(Country)%>% #Group
  summarize(mean_Score=mean(Score,na.rm=T)) #Find average score

X2018_2020_2021_avg_happiness_by_country%>%
  arrange(desc(mean_Score))#Arrange by happiest first
```

```
## # A tibble: 141 × 2
##    Country     mean_Score
##    <chr>            <dbl>
##  1 Finland           7.76
##  2 Denmark           7.61
##  3 Switzerland       7.54
##  4 Iceland           7.52
##  5 Norway            7.49
##  6 Netherlands       7.45
##  7 Sweden            7.34
##  8 New Zealand       7.30
##  9 Austria           7.23
## 10 Australia         7.23
## # … with 131 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

```
X2018_2020_2021_avg_happiness_by_country%>% #Try it
  arrange(mean_Score)#Arrange by unhappiest first
```

```
## # A tibble: 141 × 2
##    Country     mean_Score
##    <chr>            <dbl>
##  1 Afghanistan       2.91
##  2 Rwanda            3.38
##  3 Zimbabwe          3.38
##  4 Tanzania          3.47
##  5 Burundi           3.49
##  6 Botswana          3.51
##  7 Yemen             3.51
##  8 Malawi            3.58
##  9 Haiti             3.64
## 10 Lesotho           3.66
## # … with 131 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

Pivoting

This allows us to view the Rank and the Overall Score for each country in order to compare the way the score affects the rank. It seems that the higher the score the higher the rank.

```
X2018%>%
  pivot_longer(cols=c('Overall rank','Score'),
                   names_to='Rank',
                   values_to='Scores')%>%
  select(1,8,9,2,3,4,5,6,7)
```

```
## # A tibble: 312 × 9
##    Country or reg…¹ Rank  Scores GDP p…² Socia…³ Healt…⁴ Freed…⁵ Gener…⁶ Perce…⁷
##    <chr>            <chr>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
##  1 Finland          Over…   1       1.30    1.59   0.874   0.681   0.202 0.393
##  2 Finland          Score   7.63    1.30    1.59   0.874   0.681   0.202 0.393
##  3 Norway           Over…   2       1.46    1.58   0.861   0.686   0.286 0.340
##  4 Norway           Score   7.59    1.46    1.58   0.861   0.686   0.286 0.340
##  5 Denmark          Over…   3       1.35    1.59   0.868   0.683   0.284 0.408
##  6 Denmark          Score   7.56    1.35    1.59   0.868   0.683   0.284 0.408
##  7 Iceland          Over…   4       1.34    1.64   0.914   0.677   0.353 0.138
##  8 Iceland          Score   7.50    1.34    1.64   0.914   0.677   0.353 0.138
##  9 Switzerland      Over…   5       1.42    1.55   0.927   0.66    0.256 0.357
## 10 Switzerland      Score   7.49    1.42    1.55   0.927   0.66    0.256 0.357
## # … with 302 more rows, and abbreviated variable names ¹`Country or region`,
## #   ²`GDP per capita`, ³`Social support`, ⁴`Healthy life expectancy`,
## #   ⁵`Freedom to make life choices`, ⁶Generosity, ⁷`Perceptions of corruption`
## # ℹ Use `print(n = ...)` to see more rows
```

Categorical Summary Statistics

Number of countries in each region

This allows us to see the number of countries in each region in order to guage how many different peoples and cultures each region contains. It seems that Sub-Saharan Africa contains the most countries with 33 countries and North America and ANZ contain the least with 4 countries.

```
X2018_2020_2021%>%
  group_by(Regional_indicator)%>%
  summarize(distinct_countries= n_distinct(Country))
```

```
## # A tibble: 10 × 2
##    Regional_indicator                distinct_countries
##    <chr>                                          <int>
##  1 Central and Eastern Europe                        16
##  2 Commonwealth of Independent States                11
##  3 East Asia                                          5
##  4 Latin America and Caribbean                       20
##  5 Middle East and North Africa                      17
##  6 North America and ANZ                              4
##  7 South Asia                                         6
##  8 Southeast Asia                                     9
##  9 Sub-Saharan Africa                                33
## 10 Western Europe                                    20
```

Total days contained in the Covid daily data set

This allows us to understand just how many days are being examined by the covid daily data set to get a better view of just how long the pandemic went on. There were 844 total days of covid data recorded in this data set, that is almost 2 and a half years of Covid.

```
covid_daily%>%
  summarize(total_days= n_distinct(date))
```

```
## # A tibble: 1 × 1
##   total_days
##        <int>
## 1        844
```

```
844/365
```

```
## [1] 2.312329
```

Here is all of our data sets / summary statistics

```
#Monthly covid data
covid_monthly_view_2018
```

```
## # A tibble: 0 × 5
## # Groups:   Country, month [0]
## # … with 5 variables: Country <chr>, month <chr>, year <chr>,
## #   new_cases_per_month <dbl>, new_deaths_per_month <dbl>
## # ℹ Use `colnames()` to see all variable names
```

```
covid_monthly_view_2020
```

```
## # A tibble: 2,476 × 5
## # Groups:   Country, month [2,476]
##    Country     month year  new_cases_per_month new_deaths_per_month
##    <chr>       <chr> <chr>               <dbl>                <dbl>
##  1 Afghanistan 10    2020                 2157                   78
##  2 Afghanistan 11    2020                 5073                  238
##  3 Afghanistan 12    2020                 6015                  427
##  4 Afghanistan 2     2020                    0                    0
##  5 Afghanistan 3     2020                  173                    4
##  6 Afghanistan 4     2020                 1997                   60
##  7 Afghanistan 5     2020                13034                  193
##  8 Afghanistan 6     2020                16312                  489
##  9 Afghanistan 7     2020                 5158                  526
## 10 Afghanistan 8     2020                 1490                  130
## # … with 2,466 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

```
covid_monthly_view_2021
```

```
## # A tibble: 2,693 × 5
## # Groups:   Country, month [2,693]
##    Country     month year  new_cases_per_month new_deaths_per_month
##    <chr>       <chr> <chr>               <dbl>                <dbl>
##  1 Afghanistan 1     2021                 2546                  203
##  2 Afghanistan 10    2021                 1059                   74
##  3 Afghanistan 11    2021                 1053                   28
##  4 Afghanistan 12    2021                  795                   48
##  5 Afghanistan 2     2021                  674                   40
##  6 Afghanistan 3     2021                  784                   45
##  7 Afghanistan 4     2021                 3425                  142
##  8 Afghanistan 5     2021                13039                  343
##  9 Afghanistan 6     2021                47235                 1988
## 10 Afghanistan 7     2021                27285                 1775
## # … with 2,683 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

```
#Yearly covid data
covid_yearly_view
```

```
## # A tibble: 450 × 4
## # Groups:   Country [226]
##    Country     year  new_cases_per_year new_deaths_per_year
##    <chr>       <chr>              <dbl>               <dbl>
##  1 Afghanistan 2020               52512                2201
##  2 Afghanistan 2021              105585                5155
##  3 Albania     2020               58314                1180
##  4 Albania     2021              151908                2036
##  5 Algeria     2020               99609                2756
##  6 Algeria     2021              118822                3520
##  7 Andorra     2020                8048                  84
##  8 Andorra     2021               15691                  56
##  9 Angola      2020               17552                 403
## 10 Angola      2021               64040                1365
## # … with 440 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

```
#Happiness regional data
X2018_2020_2021_regional_avg
```

```
## # A tibble: 10 × 8
##    Regional_indicator    avera…¹ avera…² avera…³ avera…⁴ avera…⁵ avera…⁶ avera…⁷
##    <chr>                   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Central and Eastern …    5.81   1.14    0.739   0.449   0.117   1.21   0.0533
##  2 Commonwealth of Inde…    5.36   0.874   0.621   0.454   0.150   1.14   0.124
##  3 East Asia                5.78   1.19    0.803   0.450   0.136   1.18   0.108
##  4 Latin America and Ca…    5.94   0.901   0.677   0.521   0.144   1.15   0.0877
##  5 Middle East and Nort…    5.26   1.04    0.637   0.392   0.141   1.01   0.115
##  6 North America and ANZ    7.17   1.37    0.866   0.623   0.303   1.35   0.306
##  7 South Asia               4.40   0.612   0.490   0.441   0.221   0.739  0.0921
##  8 Southeast Asia           5.37   0.919   0.608   0.617   0.302   1.08   0.142
##  9 Sub-Saharan Africa       4.40   0.476   0.296   0.419   0.185   0.797  0.102
## 10 Western Europe           6.93   1.37    0.887   0.566   0.212   1.31   0.250
## # … with abbreviated variable names ¹average_score, ²average_GDP,
## #   ³average_life_expectancy, ⁴average_freedom, ⁵average_generosity,
## #   ⁶average_social_support, ⁷average_corruption
```

```
#Covid daily data
covid_daily_split
```

```
## # A tibble: 184,260 × 9
##    year  month day   Country     cumulative_to…¹ daily…² activ…³ cumul…⁴ daily…⁵
##    <chr> <chr> <chr> <chr>                 <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 2020  2     15    Afghanistan               0      NA       0       0      NA
##  2 2020  2     16    Afghanistan               0      NA       0       0      NA
##  3 2020  2     17    Afghanistan               0      NA       0       0      NA
##  4 2020  2     18    Afghanistan               0      NA       0       0      NA
##  5 2020  2     19    Afghanistan               0      NA       0       0      NA
##  6 2020  2     20    Afghanistan               0      NA       0       0      NA
##  7 2020  2     21    Afghanistan               0      NA       0       0      NA
##  8 2020  2     22    Afghanistan               0      NA       0       0      NA
##  9 2020  2     23    Afghanistan               0      NA       0       0      NA
## 10 2020  2     24    Afghanistan               1      NA       1       0      NA
## # … with 184,250 more rows, and abbreviated variable names
## #   ¹cumulative_total_cases, ²daily_new_cases, ³active_cases,
## #   ⁴cumulative_total_deaths, ⁵daily_new_deaths
## # ℹ Use `print(n = ...)` to see more rows
```

```
#Covid summary data
covid_sum_clean1
```

```
## # 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
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
#All Happiness Data
X2018_2020_2021
```

```
## # A tibble: 423 × 10
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2 Albania     Centra…  4.59 0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3 Algeria     Middle…  5.30 0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4 Argentina   Latin …  6.39 1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5 Armenia     Common…  4.32 0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6 Australia   North …  7.27 1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7 Austria     Wester…  7.14 1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8 Azerbaijan  Common…  5.20 1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9 Bahrain     Middle…  6.10 1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10 Bangladesh  South …  4.5  0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 413 more rows, and abbreviated variable names ¹Regional_indicator,
## #   ²Social_support, ³Life_expectancy, ⁴Generosity, ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows
```

```
#Average Happiness by country across all three years
X2018_2020_2021_avg_happiness_by_country
```

```
## # A tibble: 141 × 2
##    Country     mean_Score
##    <chr>            <dbl>
##  1 Afghanistan       2.91
##  2 Albania           4.86
##  3 Algeria           5.06
##  4 Argentina         6.10
##  5 Armenia           4.76
##  6 Australia         7.23
##  7 Austria           7.23
##  8 Azerbaijan        5.18
##  9 Bahrain           6.33
## 10 Bangladesh        4.79
## # … with 131 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

Master data set

The function left_join() was used to join all of the Happiness data sets to the yearly covid summary.

```
X2018_2020_2021_by_yearly_covid  <- X2018_2020_2021%>% #Rename
  left_join(covid_yearly_view, by = c("Year"="year","Country")) #Join

X2018_2020_2021_by_yearly_covid #Try it out
```

```
## # A tibble: 423 × 12
##    Country     Regio…¹ Score   GDP Socia…² Life_…³ Freedom Gener…⁴ Corru…⁵ Year
##    <chr>       <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 Afghanistan South …  3.63 0.332   0.537   0.255   0.085   0.191 0.036   2018
##  2 Albania     Centra…  4.59 0.916   0.817   0.79    0.419   0.149 0.032   2018
##  3 Algeria     Middle…  5.30 0.979   1.15    0.687   0.077   0.055 0.135   2018
##  4 Argentina   Latin …  6.39 1.07    1.47    0.744   0.57    0.062 0.054   2018
##  5 Armenia     Common…  4.32 0.816   0.99    0.666   0.26    0.077 0.028   2018
##  6 Australia   North …  7.27 1.34    1.57    0.91    0.647   0.361 0.302   2018
##  7 Austria     Wester…  7.14 1.34    1.50    0.891   0.617   0.242 0.224   2018
##  8 Azerbaijan  Common…  5.20 1.02    1.16    0.603   0.43    0.031 0.176   2018
##  9 Bahrain     Middle…  6.10 1.34    1.37    0.698   0.594   0.243 0.123   2018
## 10 Bangladesh  South …  4.5  0.532   0.85    0.579   0.58    0.153 0.144   2018
## # … with 413 more rows, 2 more variables: new_cases_per_year <dbl>,
## #   new_deaths_per_year <dbl>, and abbreviated variable names
## #   ¹Regional_indicator, ²Social_support, ³Life_expectancy, ⁴Generosity,
## #   ⁵Corruption
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Second Master Data Set

The function left_join was used to join the first master data set that contained all three years of happiness data and the covid yearly data to the covid summary data.

This is the main dataset we will be pulling information from. It is a combination of 4 different datasets: happiness data from 2018, 2020, and 2021, as well as covid_yearly_view. When we combined the X2018_2020_2021 dataset with the covid_yearly_view dataset, we lost roughly 85 countries due to the fact that they did not exist in the X2018_2020_2021. Then, we combined the resulting dataset with covid_sum_clean1, which summarizes covid statistics for each country. There were no lost observations when doing this, and the final dataset called happiness_sumcovid_yearlycovid has 423 rows, all representing an individual country-year combo.

```
happiness_sumcovid_yearlycovid <- covid_sum_clean1 %>% #Rename
  right_join(X2018_2020_2021_by_yearly_covid, by =c("country"="Country")) %>% #Join
  mutate(new_cases_to_popsize = new_cases_per_year / population,
         fatality_rate = new_deaths_per_year / new_cases_per_year) #Create summary stats

happiness_sumcovid_yearlycovid #Try it out
```

```
## # A tibble: 423 × 25
##    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 Afghanistan Asia       179267    7690  162202    9375    1124    4420     190
##  3 Afghanistan Asia       179267    7690  162202    9375    1124    4420     190
##  4 Albania     Europe     275574    3497  271826     251       2   95954    1218
##  5 Albania     Europe     275574    3497  271826     251       2   95954    1218
##  6 Albania     Europe     275574    3497  271826     251       2   95954    1218
##  7 Algeria     Africa     265816    6875  178371   80570       6    5865     152
##  8 Algeria     Africa     265816    6875  178371   80570       6    5865     152
##  9 Algeria     Africa     265816    6875  178371   80570       6    5865     152
## 10 Argentina   South Am… 9101319  128729 8895999   76591     372  197992    2800
## # … with 413 more rows, 16 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 <chr>, new_cases_per_year <dbl>, new_deaths_per_year <dbl>,
## #   new_cases_to_popsize <dbl>, fatality_rate <dbl>, and abbreviated variable
## #   names ¹total_confirmed, ²total_deaths, ³total_recovered, ⁴active_cases, …
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

5. Visualizing

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', 'Democratic Republic of the Congo'='Congo','Palestine'='Palest
inian Territories','UK'='United Kingdom', '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>   <chr>
##  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
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
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
## # ℹ Use `print(n = ...)` to see more rows
```

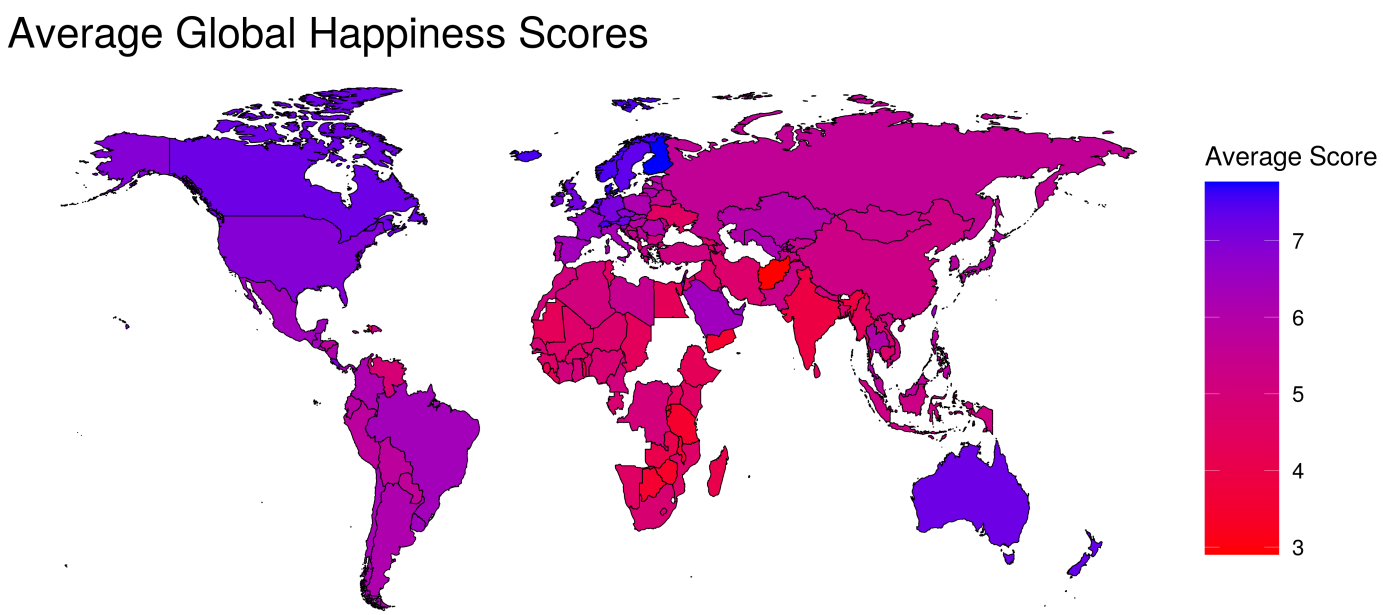
Average Global Happiness Scores

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
## # ℹ Use `print(n = ...)` to see 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=40), #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/
```

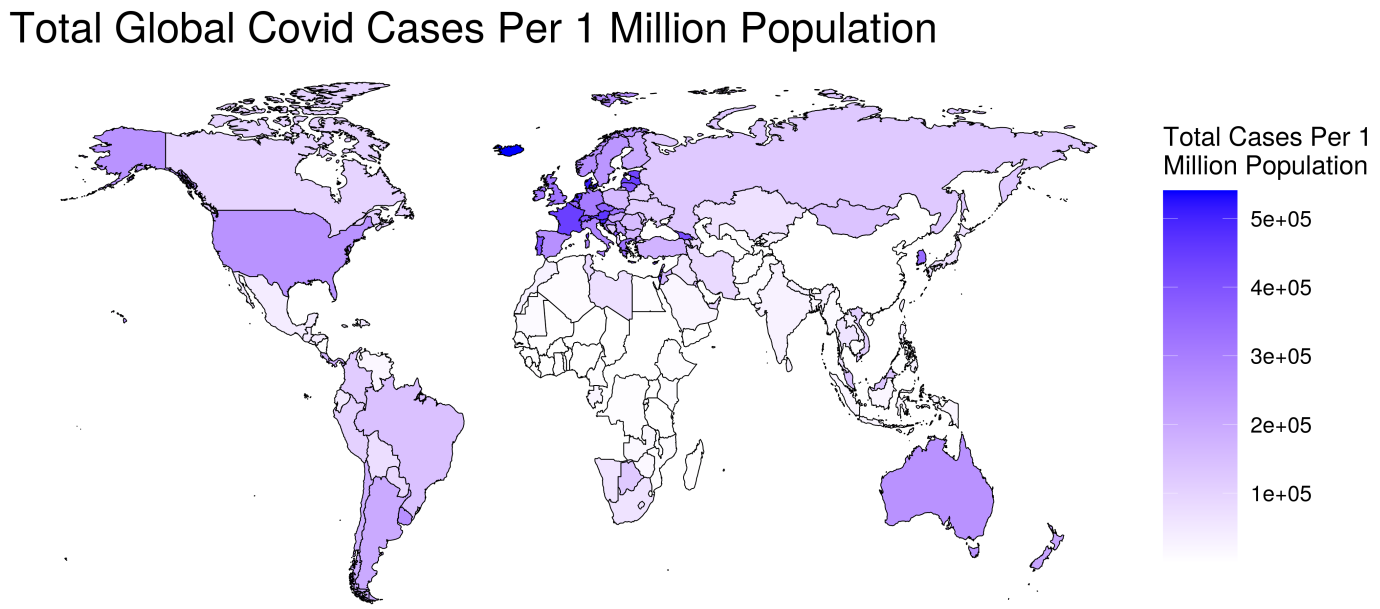
Total Global Covid Cases Per 1 Million Population

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
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
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 aesthetics
  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=40), #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
```



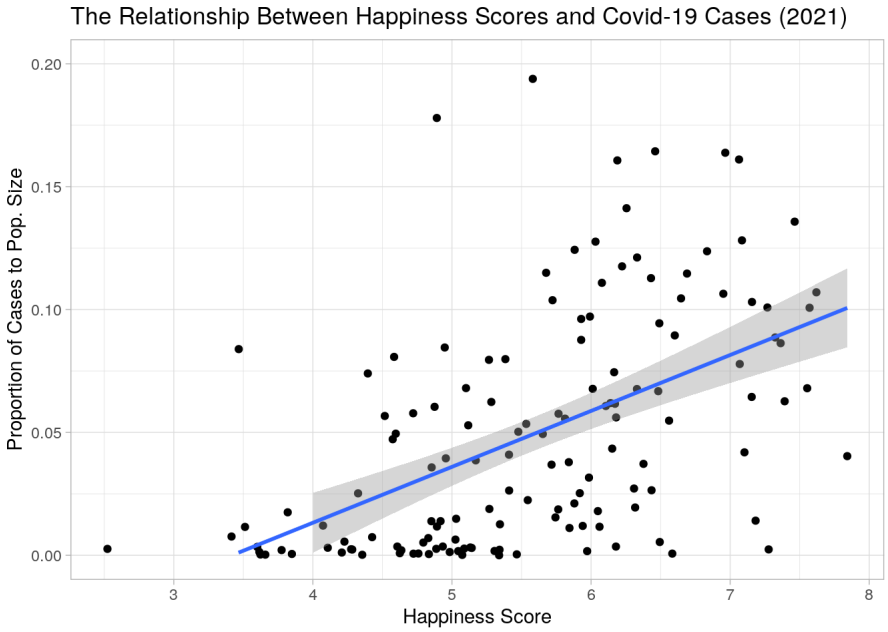

```
#This is the source I used: https://www.statology.org/remove-axis-labels-ggplot2/
```

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.

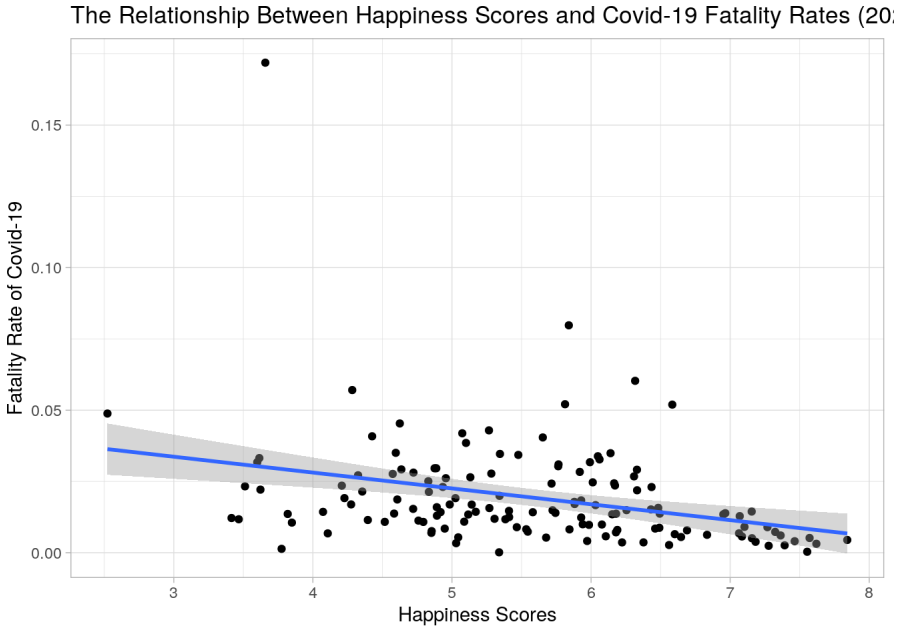
```
# Score vs cases 2021
happiness_sumcovid_yearlycovid %>%
  filter(Year == 2021) %>% #filter only the year 2021
  ggplot(aes(x = Score, y = new_cases_to_popsize)) + #Set aesthetics
  geom_point() + #Create scatterplot
  geom_smooth(method = 'lm') + #Create a smooth trend line
  ylim(0,0.2) + #Adjust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



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 and Covid-19 Fatality Rates (2021)',
       x = 'Happiness Scores',
       y = 'Fatality Rate of Covid-19') + #Add labels
  theme_light() #Change theme
```

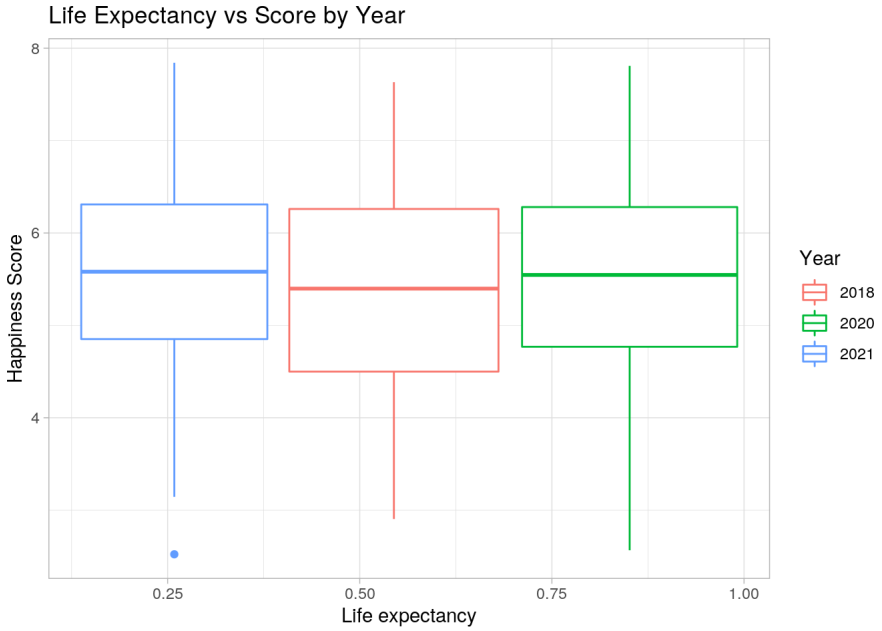


Three-Variable Graphs:

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.

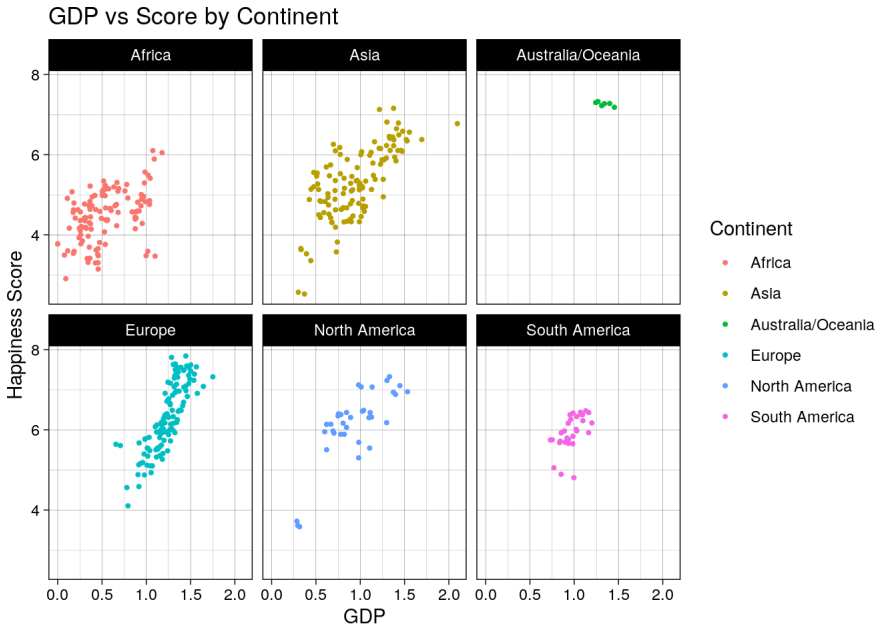
```
happiness_sumcovid_yearlycovid %>%
  ggplot()+
  labs(title = 'Life Expectancy vs Score by Year', x = 'Life expectancy', y = 'Happiness Score') +
  geom_boxplot(aes(x = Life_expectancy, y =Score , color = Year))+
  scale_x_continuous(breaks = c(0.25, 0.5, 0.75,1))+
  theme_light()
```



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
```

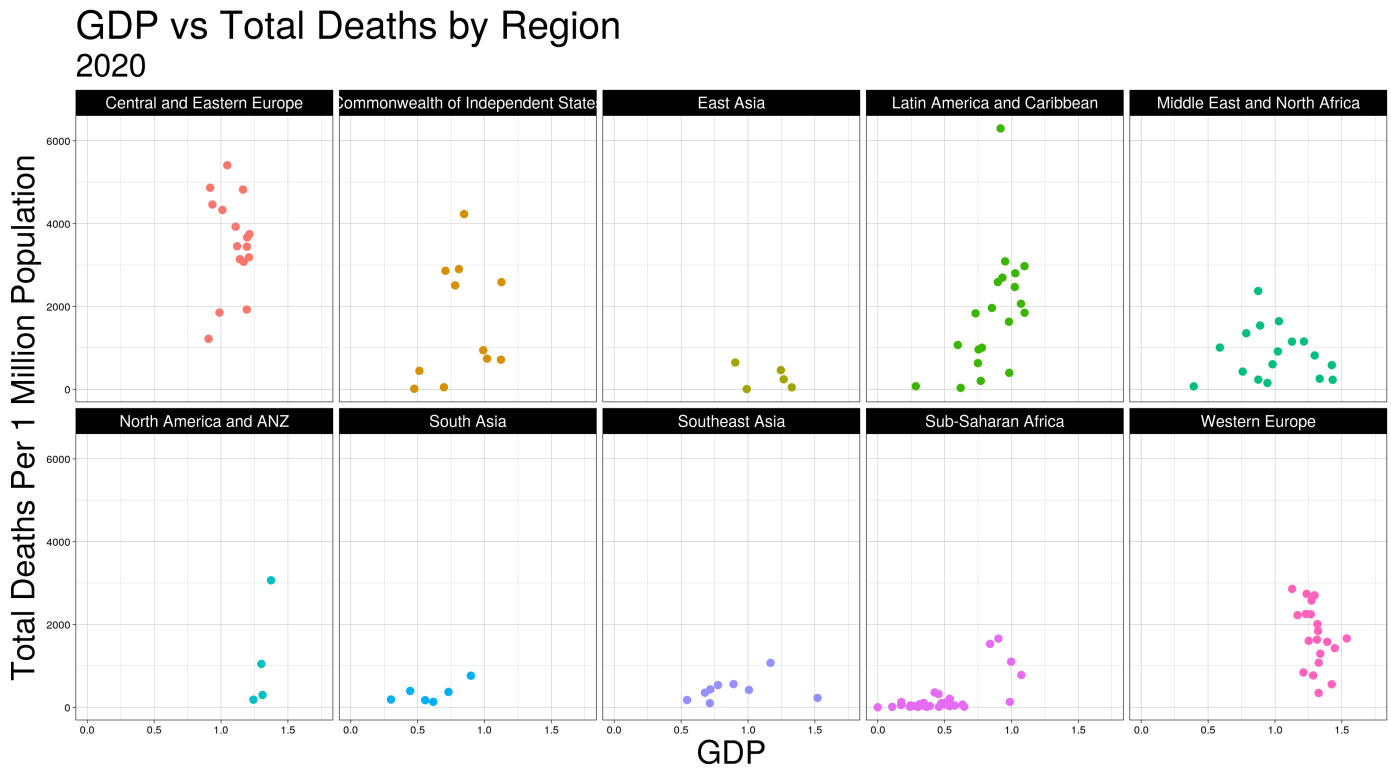


Bonus Animated Graphs

GDP vs Total Deaths by Region for 2020 and 2021

This graph shows scatterplots of the gross domestic product (GDP) versus Total deaths per 1 Million of the Population by Region for the Years 2020 and 2021. The plot overall shows a lower GDP in 2020 across all regions and higher covid deaths per 1 million population in 2021. There does not seem to be a definite relationship between GDP and total deaths per 1 million of the population in any of the regions.

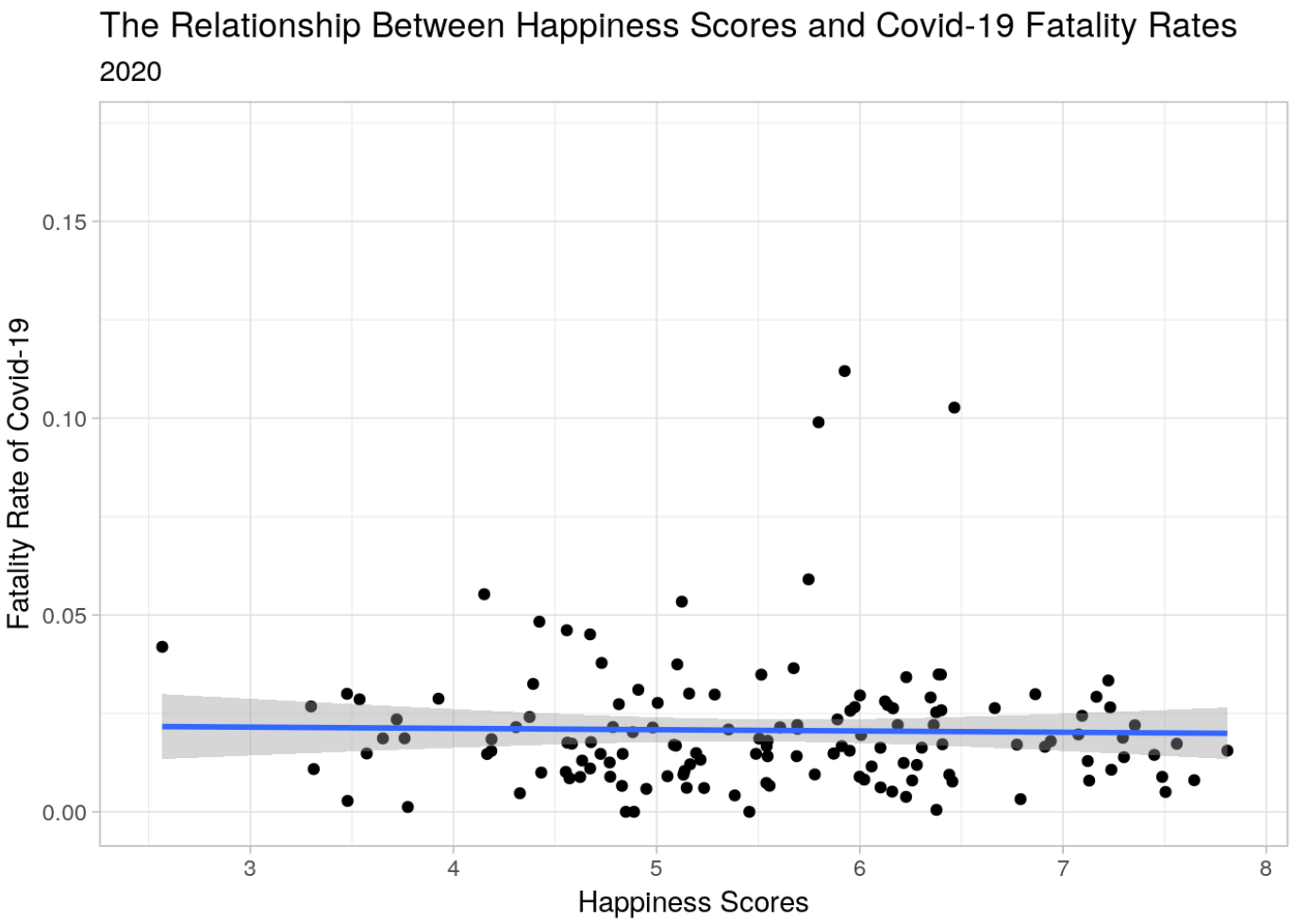
```
library(gganimate) #Call package

happiness_sumcovid_yearlycovid %>%
  filter(!Year=="2018")%>% #Take out 2018
  ggplot(aes(x = GDP, y = total_deaths_per_1m_population, color = Regional_indicator))+  #Set aesthetics
  labs(title = 'GDP vs Total Deaths by Region', x = 'GDP', y = 'Total Deaths Per 1 Million Population',
       color = 'Region') + #Label
  geom_point(stat = "identity", size=3) + #Create scatterplot
  facet_wrap(~ Regional_indicator, nrow = 2) + #Facet wrap
  theme_linedraw()+ #Change theme
  theme(legend.position = 'none', #Remove legend
        strip.text.x = element_text(size = 15), #Change font size
        title = element_text(size=30), #Change font size
        axis.text = element_text(size=10))+ #Change font size
  transition_states(Year, transition_length = 2, state_length = 1) + #Animate
  enter_fade() + #Smooth animation
  exit_shrink() + #Smooth animation
  labs(subtitle = '{closest_state}') #Label
```



The Relationship Between Happiness Scores and Covid-19 Fatality Rates Across 2020 and 2021

Again, 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. There is a visible change in the correlation between Happiness and Covid fatality between 2020 and 2021. For 2020 there was no correlation between happiness and covid mortality, with the trend line exactly horizontal. However, for 2021, 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. This is all more apparent in the year 2021 because by this year covid has fully penetrated the population and its effect is clear. In the year 2020 there may not have been much of a correlation because the pandemic was just beginning.

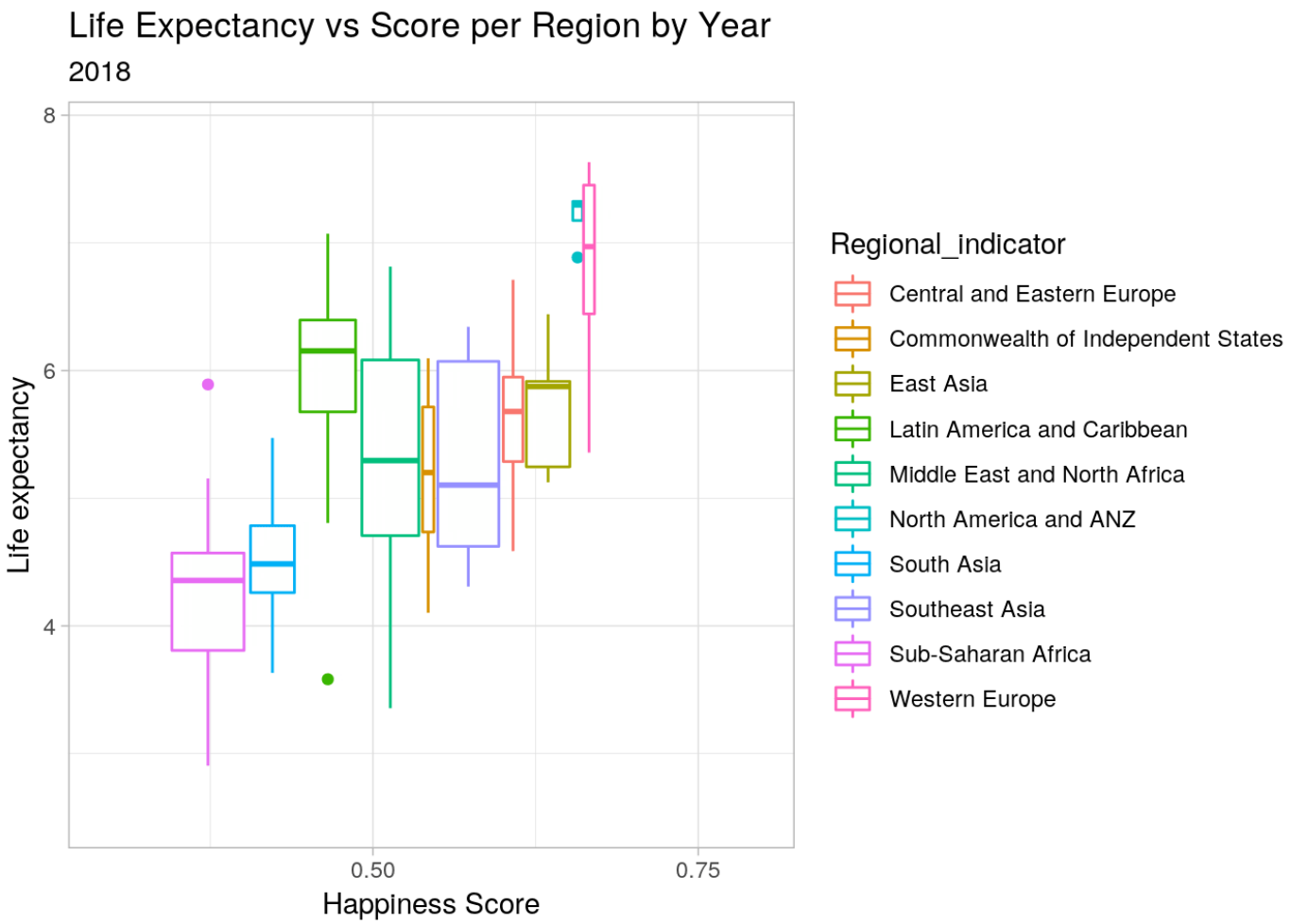
```
happiness_sumcovid_yearlycovid %>%
  filter(!Year=="2018", #Take out 2018
         fatality_rate < .2) %>% #Remove outlier
  ggplot(aes(x = Score, y = fatality_rate)) + #Set aesthetics
  geom_point() + #Create scatterplot
  geom_smooth(method = 'lm') + #Create a smooth trend line
  scale_x_continuous(breaks = seq(0,10,1)) + #Adjust x limit
  labs(title='The Relationship Between Happiness Scores and Covid-19 Fatality Rates',
       x = 'Happiness Scores',
       y = 'Fatality Rate of Covid-19') + #Add labels
  theme_light()+ #Change theme
  transition_states(Year,transition_length = 2,state_length = 2 ) + #Animate
  enter_fade() + #Smooth animation
  exit_shrink() + #Smooth animation
  labs(subtitle = '{closest_state}') #Label
```



Life Expectancy vs Score per Region by Year

This graph compares life expectancy by happiness score per region for the years 2018, 2020, and 2021. This graph gives us a unique insight on life expectancy and happiness pre pandemic, early pandemic, and late pandemic. We can see a dramatic increase in happiness for 2020 and a dramatic decrease in happiness for 2021, these may be influenced by the pandemic. The increased happiness for 2020 could be explained by everyone staying indoors, many not working and spending lots of time with family. Then once covid has made its way around the globe in 2021 there is a dramatic decrease in happiness possibly due to the effects of the pandemic, many lost jobs, many passed away, and many fell ill. The shift in life expectancy seems to vary extremely from region to region, some regions had an increased life expectancy after the pandemic and others saw a decrease.

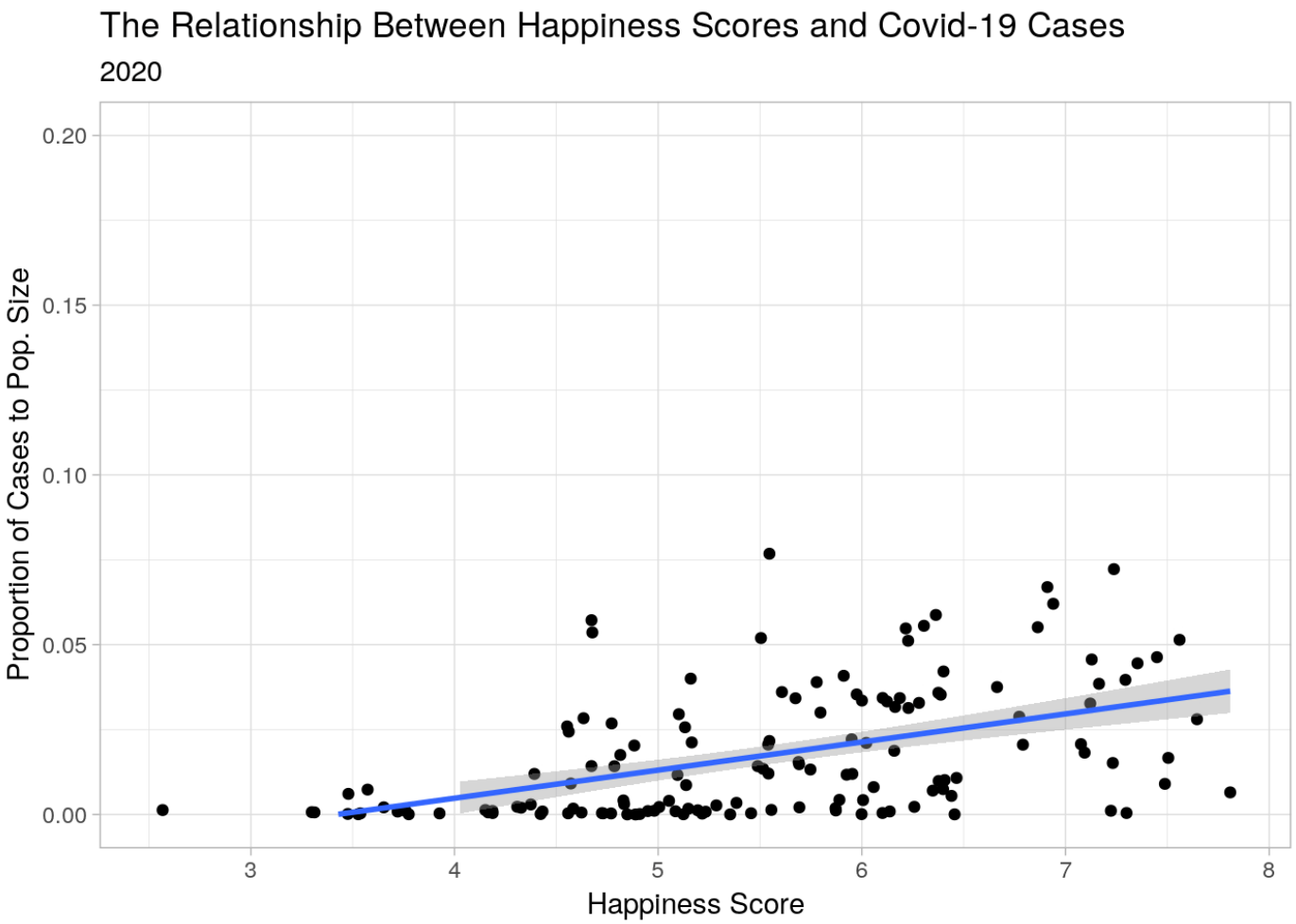
```
happiness_sumcovid_yearlycovid %>%
  ggplot()+
  labs(title = 'Life Expectancy vs Score per Region by Year', y = 'Life expectancy', x = 'Happiness Score') + #Labe;
  geom_boxplot(aes(x = Life_expectancy, y =Score , color = Regional_indicator))+ #Set aesthetics
  scale_x_continuous(breaks = c(0.25, 0.5, 0.75,1))+ #X axis adjustments
  theme_light()+ #Change theme
  transition_states(Year,transition_length = 2,state_length = 2 ) + #Animate
  enter_fade() + #Smooth animation
  exit_shrink() + #Smooth animation
  labs(subtitle = '{closest_state}') #Label
```



The Relationship Between Happiness Scores and Covid-19 Cases

This graph examines the relationship between covid-19 cases proportionate to population and happiness scores for the years 2020 and 2021. For the year 2020 there is a very very slight positive correlation between happiness and covid-19 cases, and in 2021 there is a dramatic increase in the positive correlation between happiness and covid-19 cases. This could show that the countries with higher covid cases had more freedom to travel and interact and thus produced higher case numbers.

```
happiness_sumcovid_yearlycovid %>%
  filter(!Year=="2018")%>%
  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 and Covid-19 Cases',
       x = 'Happiness Score',
       y = 'Proportion of Cases to Pop. Size') + #Add labels
  theme_light() + #Change theme
  transition_states(
    Year,
    transition_length = 2,
    state_length = 2
  ) +
  enter_fade() +
  exit_shrink() +
  labs(subtitle = '{closest_state}')
```


The Relationship Between Happiness Scores and Covid-19 Cases
2020
Proportion of Cases to Pop. Size
0.20
0.15
0.10
0.05
0.00
3
4
5
6
7
8
Happiness Score