World Happiness Report 2021 Analysis

STAT-597: Data Wrangling and Husbandry

Harshini Bonam (sdb202)

4/26/2021

Introduction

World Happiness Report (ref: https://worldhappiness.report/faq/)

World Happiness Report 2021 use data that come from the Gallup World Poll surveys from 2005 to 2020 which is a publication of the United Nations Sustainable Development Solutions Network.

They are based on answers to the main life evaluation question asked in the poll. This is called the Cantril ladder: it asks respondents to think of a ladder, with the best possible life for them being a 10, and the worst possible life being a 0.

They are then asked to rate their own current lives on that 0 to 10 scale.

They are based entirely on the survey scores, using the Gallup weights to make the estimates representative.

Data Wrangling

Load the data as tibbles.

```
# Read the data for 2021 and the previous years from 2005 to 2020
df_2021_raw = as_tibble(read.csv("./data/world-happiness-report-2021.csv"))
df_prev_raw = as_tibble(read.csv("./data/world-happiness-report.csv"))

# Let's take a peek at the data
head(df_prev_raw)
```

```
## # A tibble: 6 x 11
##   Country.name  year Life.Ladder Log.GDP.per.capita Social.support
##   <chr>        <int>       <dbl>              <dbl>          <dbl>
## 1 Afghanistan   2008        3.72               7.37          0.451
## 2 Afghanistan   2009        4.40               7.54          0.552
## 3 Afghanistan   2010        4.76               7.65          0.539
## 4 Afghanistan   2011        3.83               7.62          0.521
## 5 Afghanistan   2012        3.78               7.70          0.521
## 6 Afghanistan   2013        3.57               7.72          0.484
## # ... with 6 more variables: Healthy.life.expectancy.at.birth <dbl>,
## #   Freedom.to.make.life.choices <dbl>, Generosity <dbl>,
## #   Perceptions.of.corruption <dbl>, Positive.affect <dbl>,
## #   Negative.affect <dbl>
```

```r
head(df_2021_raw)
```

```
## # A tibble: 6 x 20
##   Country.name Regional.indicat~ Ladder.score Standard.error.of.la~ upperwhisker
##   <chr>        <chr>                    <dbl>                 <dbl>        <dbl>
## 1 Finland      Western Europe            7.84                 0.032         7.90
## 2 Denmark      Western Europe            7.62                 0.035         7.69
## 3 Switzerland  Western Europe            7.57                 0.036         7.64
## 4 Iceland      Western Europe            7.55                 0.059         7.67
## 5 Netherlands  Western Europe            7.46                 0.027         7.52
## 6 Norway       Western Europe            7.39                 0.035         7.46
## # ... with 15 more variables: 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>, Explained.by..Social.support <dbl>,
## #   Explained.by..Healthy.life.expectancy <dbl>,
## #   Explained.by..Freedom.to.make.life.choices <dbl>,
## #   Explained.by..Generosity <dbl>,
## #   Explained.by..Perceptions.of.corruption <dbl>, Dystopia...residual <dbl>
```

Data Exploration

```r
# Let's see the common column names between the two datasets
#Column names in 2021 dataset
colnames(df_2021_raw)
```

```
##  [1] "Country.name"
##  [2] "Regional.indicator"
##  [3] "Ladder.score"
##  [4] "Standard.error.of.ladder.score"
##  [5] "upperwhisker"
##  [6] "lowerwhisker"
##  [7] "Logged.GDP.per.capita"
##  [8] "Social.support"
##  [9] "Healthy.life.expectancy"
## [10] "Freedom.to.make.life.choices"
## [11] "Generosity"
## [12] "Perceptions.of.corruption"
## [13] "Ladder.score.in.Dystopia"
## [14] "Explained.by..Log.GDP.per.capita"
## [15] "Explained.by..Social.support"
## [16] "Explained.by..Healthy.life.expectancy"
## [17] "Explained.by..Freedom.to.make.life.choices"
## [18] "Explained.by..Generosity"
## [19] "Explained.by..Perceptions.of.corruption"
## [20] "Dystopia...residual"
```

```r
#Column names in 2005-2020 dataset
colnames(df_prev_raw)
```

```
##  [1] "Country.name"                     "year"
##  [3] "Life.Ladder"                      "Log.GDP.per.capita"
##  [5] "Social.support"                   "Healthy.life.expectancy.at.birth"
##  [7] "Freedom.to.make.life.choices"     "Generosity"
##  [9] "Perceptions.of.corruption"        "Positive.affect"
## [11] "Negative.affect"
```

```
common_colnames = intersect(colnames(df_2021_raw), colnames(df_prev_raw))
common_colnames
```

```
## [1] "Country.name"                 "Social.support"
## [3] "Freedom.to.make.life.choices" "Generosity"
## [5] "Perceptions.of.corruption"
```

```
# Let's take a peek at the data
#The summary of 2021 dataset is
summary(df_2021_raw)
```

```
##  Country.name       Regional.indicator  Ladder.score
##  Length:149         Length:149         Min.   :2.523
##  Class :character   Class :character   1st Qu.:4.852
##  Mode  :character   Mode  :character   Median :5.534
##                                        Mean   :5.533
##                                        3rd Qu.:6.255
##                                        Max.   :7.842
##  Standard.error.of.ladder.score  upperwhisker    lowerwhisker
##  Min.   :0.02600                Min.   :2.596   Min.   :2.449
##  1st Qu.:0.04300                1st Qu.:4.991   1st Qu.:4.706
##  Median :0.05400                Median :5.625   Median :5.413
##  Mean   :0.05875                Mean   :5.648   Mean   :5.418
##  3rd Qu.:0.07000                3rd Qu.:6.344   3rd Qu.:6.128
##  Max.   :0.17300                Max.   :7.904   Max.   :7.780
##  Logged.GDP.per.capita Social.support   Healthy.life.expectancy
##  Min.   : 6.635        Min.   :0.4630   Min.   :48.48
##  1st Qu.: 8.541        1st Qu.:0.7500   1st Qu.:59.80
##  Median : 9.569        Median :0.8320   Median :66.60
##  Mean   : 9.432        Mean   :0.8147   Mean   :64.99
##  3rd Qu.:10.421        3rd Qu.:0.9050   3rd Qu.:69.60
##  Max.   :11.647        Max.   :0.9830   Max.   :76.95
##  Freedom.to.make.life.choices   Generosity       Perceptions.of.corruption
##  Min.   :0.3820               Min.   :-0.28800   Min.   :0.0820
##  1st Qu.:0.7180               1st Qu.:-0.12600   1st Qu.:0.6670
##  Median :0.8040               Median :-0.03600   Median :0.7810
##  Mean   :0.7916               Mean   :-0.01513   Mean   :0.7274
##  3rd Qu.:0.8770               3rd Qu.: 0.07900   3rd Qu.:0.8450
##  Max.   :0.9700               Max.   : 0.54200   Max.   :0.9390
##  Ladder.score.in.Dystopia Explained.by..Log.GDP.per.capita
##  Min.   :2.43             Min.   :0.0000
##  1st Qu.:2.43             1st Qu.:0.6660
##  Median :2.43             Median :1.0250
##  Mean   :2.43             Mean   :0.9772
##  3rd Qu.:2.43             3rd Qu.:1.3230
##  Max.   :2.43             Max.   :1.7510
```

```
##  Explained.by..Social.support Explained.by..Healthy.life.expectancy
##  Min.   :0.0000               Min.   :0.0000
##  1st Qu.:0.6470               1st Qu.:0.3570
##  Median :0.8320               Median :0.5710
##  Mean   :0.7933               Mean   :0.5202
##  3rd Qu.:0.9960               3rd Qu.:0.6650
##  Max.   :1.1720               Max.   :0.8970
##  Explained.by..Freedom.to.make.life.choices Explained.by..Generosity
##  Min.   :0.0000                             Min.   :0.000
##  1st Qu.:0.4090                             1st Qu.:0.105
##  Median :0.5140                             Median :0.164
##  Mean   :0.4987                             Mean   :0.178
##  3rd Qu.:0.6030                             3rd Qu.:0.239
##  Max.   :0.7160                             Max.   :0.541
##  Explained.by..Perceptions.of.corruption Dystopia...residual
##  Min.   :0.0000                          Min.   :0.648
##  1st Qu.:0.0600                          1st Qu.:2.138
##  Median :0.1010                          Median :2.509
##  Mean   :0.1351                          Mean   :2.430
##  3rd Qu.:0.1740                          3rd Qu.:2.794
##  Max.   :0.5470                          Max.   :3.482
```

```
# There are no missing values in the 2021 dataset

#The summary of previous years(2005-2020) dataset is
summary(df_prev_raw)
```

```
##  Country.name            year       Life.Ladder    Log.GDP.per.capita
##  Length:1949        Min.   :2005   Min.   :2.375   Min.   : 6.635
##  Class :character   1st Qu.:2010   1st Qu.:4.640   1st Qu.: 8.464
##  Mode  :character   Median :2013   Median :5.386   Median : 9.460
##                     Mean   :2013   Mean   :5.467   Mean   : 9.368
##                     3rd Qu.:2017   3rd Qu.:6.283   3rd Qu.:10.353
##                     Max.   :2020   Max.   :8.019   Max.   :11.648
##                                                    NA's   :36
##  Social.support   Healthy.life.expectancy.at.birth Freedom.to.make.life.choices
##  Min.   :0.2900   Min.   :32.30                    Min.   :0.2580
##  1st Qu.:0.7498   1st Qu.:58.69                    1st Qu.:0.6470
##  Median :0.8355   Median :65.20                    Median :0.7630
##  Mean   :0.8126   Mean   :63.36                    Mean   :0.7426
##  3rd Qu.:0.9050   3rd Qu.:68.59                    3rd Qu.:0.8560
##  Max.   :0.9870   Max.   :77.10                    Max.   :0.9850
##  NA's   :13       NA's   :55                       NA's   :32
##    Generosity      Perceptions.of.corruption Positive.affect  Negative.affect
##  Min.   :-0.3350   Min.   :0.0350            Min.   :0.3220   Min.   :0.0830
##  1st Qu.:-0.1130   1st Qu.:0.6900            1st Qu.:0.6255   1st Qu.:0.2060
##  Median :-0.0255   Median :0.8020            Median :0.7220   Median :0.2580
##  Mean   : 0.0001   Mean   :0.7471            Mean   :0.7100   Mean   :0.2685
##  3rd Qu.: 0.0910   3rd Qu.:0.8720            3rd Qu.:0.7990   3rd Qu.:0.3200
##  Max.   : 0.6980   Max.   :0.9830            Max.   :0.9440   Max.   :0.7050
##  NA's   :89        NA's   :110               NA's   :22       NA's   :16
```

```r
# There are many NA's in the previous year dataset.
# These columns are:
colMeans(is.na(df_prev_raw))
```

```
##                     Country.name                             year
##                      0.000000000                      0.000000000
##                      Life.Ladder                Log.GDP.per.capita
##                      0.000000000                      0.018471011
##                   Social.support Healthy.life.expectancy.at.birth
##                      0.006670087                      0.028219600
##     Freedom.to.make.life.choices                       Generosity
##                      0.016418676                      0.045664443
##        Perceptions.of.corruption                  Positive.affect
##                      0.056439200                      0.011287840
##                  Negative.affect
##                      0.008209338
```

```r
# Let's also check common countries in the two datasets
diff_countries = setdiff(df_prev_raw$Country.name, df_2021_raw$Country.name)
diff_countries
```

```
##  [1] "Angola"                   "Belize"
##  [3] "Bhutan"                   "Central African Republic"
##  [5] "Congo (Kinshasa)"         "Cuba"
##  [7] "Djibouti"                 "Guyana"
##  [9] "Oman"                     "Qatar"
## [11] "Somalia"                  "Somaliland region"
## [13] "South Sudan"              "Sudan"
## [15] "Suriname"                 "Syria"
## [17] "Trinidad and Tobago"
```

Data Cleansing

```r
# Let's fill these values with mean value of each country in the given range
# of years.

df_prev = df_prev_raw %>%
    select(Country.name, year, Log.GDP.per.capita, Social.support,
           Healthy.life.expectancy.at.birth, Freedom.to.make.life.choices,
           Generosity, Perceptions.of.corruption, Life.Ladder) %>%
    group_by(Country.name) %>%
    mutate(
        Log.GDP.per.capita = impute_mean(Log.GDP.per.capita),
        Social.support = impute_mean(Social.support),
        Healthy.life.expectancy.at.birth = impute_mean(Healthy.life.expectancy.at.birth),
        Freedom.to.make.life.choices = impute_mean(Freedom.to.make.life.choices),
        Generosity = impute_mean(Generosity),
        Perceptions.of.corruption = impute_mean(Perceptions.of.corruption)
    ) %>%
    rename(
```

```
      Ladder.score = Life.Ladder,
      Logged.GDP.per.capita = Log.GDP.per.capita,
      Healthy.life.expectancy = Healthy.life.expectancy.at.birth
    )

# Now let's see how much we improved on filling missing values.
colMeans(is.na(df_prev))
```

```
##                 Country.name                         year
##                0.0000000000                0.0000000000
##        Logged.GDP.per.capita               Social.support
##                0.0097485890                0.0005130836
##      Healthy.life.expectancy Freedom.to.make.life.choices
##                0.0184710108                0.0000000000
##                  Generosity    Perceptions.of.corruption
##                0.0097485890                0.0143663417
##                Ladder.score
##                0.0000000000
```

```
#reduced by
colMeans(is.na(df_prev_raw %>% select(Country.name, year, Log.GDP.per.capita, Social.support,
          Healthy.life.expectancy.at.birth, Freedom.to.make.life.choices,
          Generosity, Perceptions.of.corruption, Life.Ladder))) - colMeans(is.na(df_prev))
```

```
##                      Country.name                              year
##                       0.000000000                       0.000000000
##                Log.GDP.per.capita                    Social.support
##                       0.008722422                       0.006157004
## Healthy.life.expectancy.at.birth     Freedom.to.make.life.choices
##                       0.009748589                       0.016418676
##                        Generosity        Perceptions.of.corruption
##                       0.035915854                       0.042072858
##                       Life.Ladder
##                       0.000000000
```

```
# Improved common column names
common_colnames = intersect(colnames(df_2021_raw), colnames(df_prev))
common_colnames
```

```
## [1] "Country.name"                 "Ladder.score"
## [3] "Logged.GDP.per.capita"        "Social.support"
## [5] "Healthy.life.expectancy"      "Freedom.to.make.life.choices"
## [7] "Generosity"                   "Perceptions.of.corruption"
```

```
df_2021 = df_2021_raw %>%
  select (Country.name, Ladder.score, Logged.GDP.per.capita, Social.support,
          Healthy.life.expectancy, Freedom.to.make.life.choices, Generosity,
          Perceptions.of.corruption, Regional.indicator) %>%
  mutate(year = 2021)

df_total = df_2021 %>%
```

```
  select(-Regional.indicator) %>%
  bind_rows(df_prev)

df_total_ladder_wider  = df_total %>%
  select(Country.name, year, Ladder.score) %>%
  pivot_wider(names_from = year, values_from = Ladder.score)
```

Data Analysis

We cleansed the data for some basic data analysis like to find out the following: > Top 10 happiest countries in 2021 > Least 10 happiest countries in 2021 > Region wise happiness concentration in 2021

```
# dimensions
dimensions = c('Ladder.score', 'Logged.GDP.per.capita', 'Social.support',
               'Healthy.life.expectancy', 'Freedom.to.make.life.choices',
               'Generosity', 'Perceptions.of.corruption')


# Transform the dataset to longer structure, like
# country, dimension, score
df_2021_long = df_2021 %>%
    select(country = Country.name, all_of(dimensions)) %>%
    mutate(absence_of_corruption = 1 - Perceptions.of.corruption) %>%
    pivot_longer(
      cols = c(all_of(dimensions),'absence_of_corruption'),
      names_to = 'dimension', values_to = 'score') %>%
    filter(dimension != "Perceptions.of.corruption")

head(df_2021_long, n = 5)
```

```
## # A tibble: 5 x 3
##   country dimension                    score
##   <chr>   <chr>                        <dbl>
## 1 Finland Ladder.score                 7.84
## 2 Finland Logged.GDP.per.capita       10.8
## 3 Finland Social.support               0.954
## 4 Finland Healthy.life.expectancy     72
## 5 Finland Freedom.to.make.life.choices  0.949
```

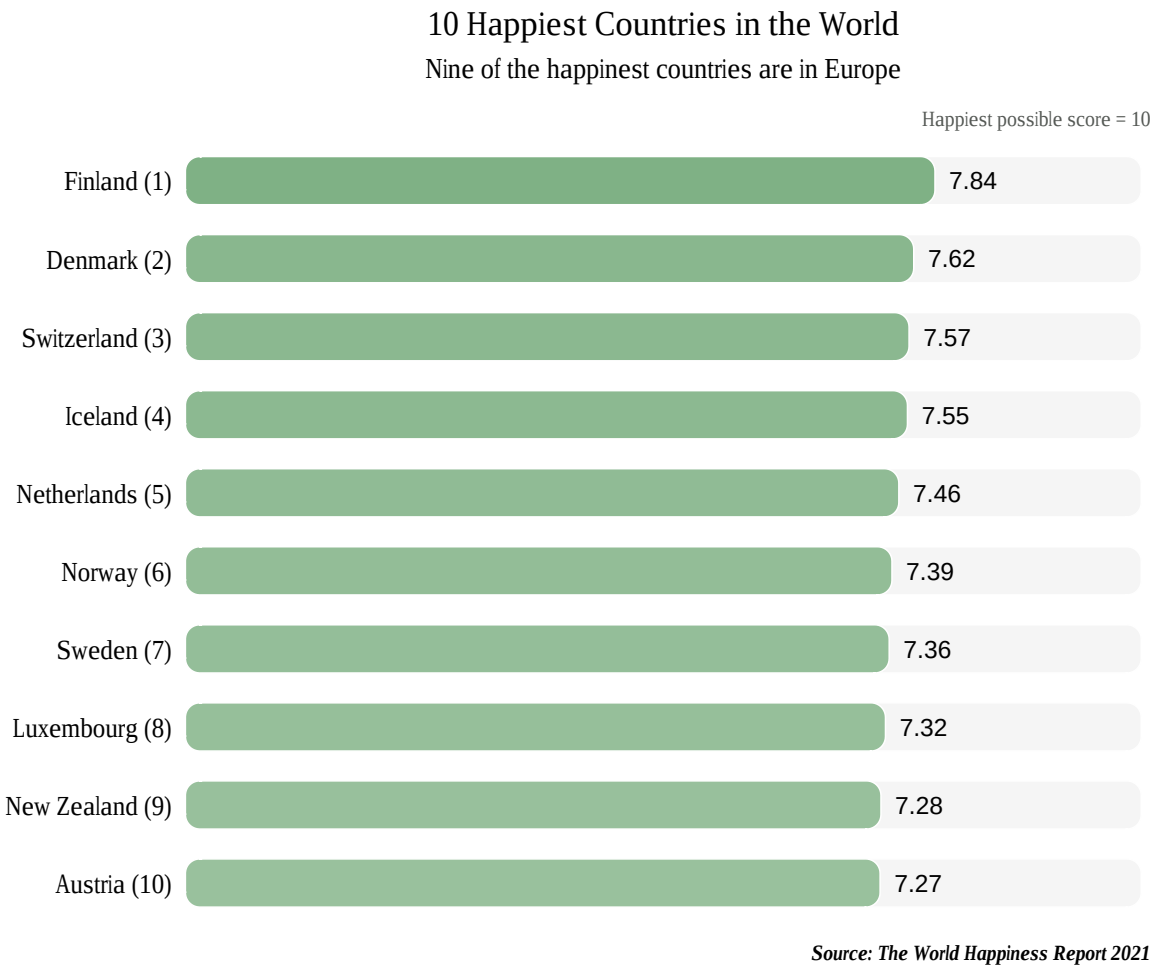
```
# Compute the percentage of the dimensional score for each country
df_2021_tranformed = df_2021_long %>%
    group_by(dimension) %>%
    mutate(min_value = min(score),
             max_value = max(score)) %>%
    mutate(score_pct = (score-min_value)/(max_value-min_value)) %>%
    ungroup()

head(df_2021_tranformed, n = 5)
```

```
## # A tibble: 5 x 6
##   country dimension                    score min_value max_value score_pct
```

```
##   <chr>   <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
## 1 Finland Ladder.score                    7.84      2.52      7.84      1
## 2 Finland Logged.GDP.per.capita          10.8       6.64     11.6       0.826
## 3 Finland Social.support                  0.954     0.463     0.983     0.944
## 4 Finland Healthy.life.expectancy        72        48.5      77.0       0.826
## 5 Finland Freedom.to.make.life.choices   0.949     0.382     0.97      0.964
```

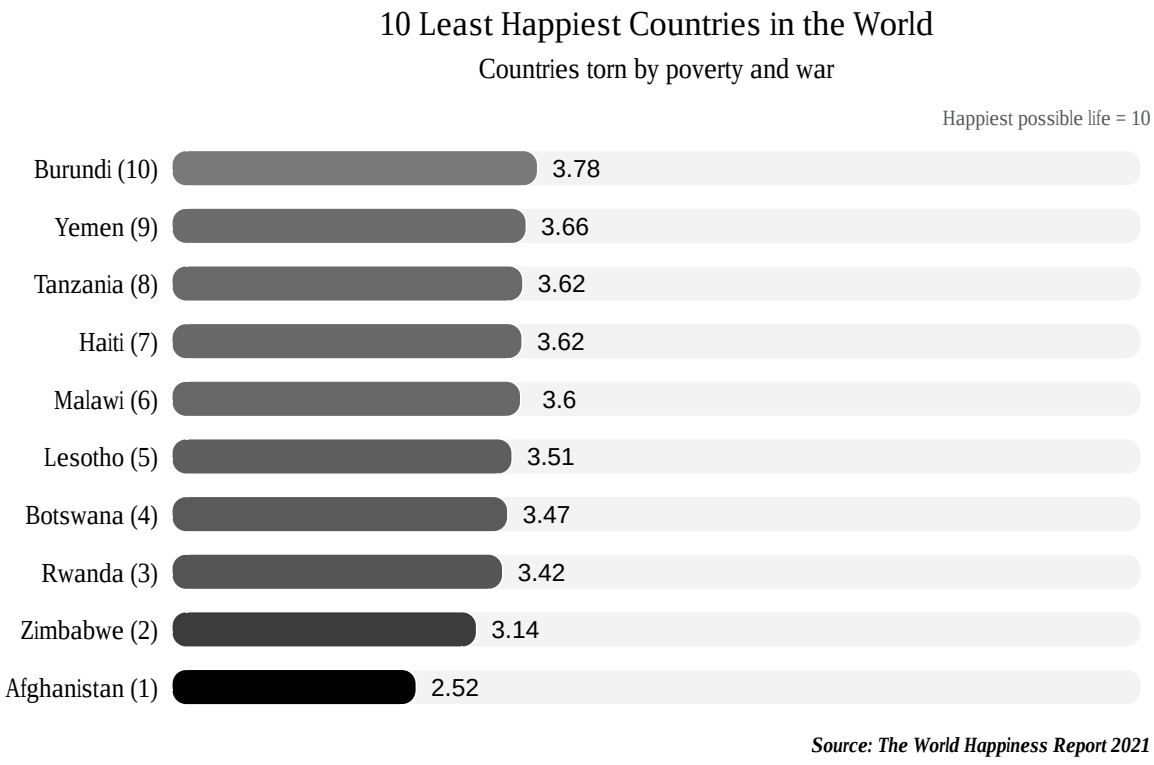
Top 10 happiest countries

```
# Let's retrieve top 10 happiest countries based on Ladder.score
df_2021_top10 = df_2021_tranformed %>%
    filter(dimension == "Ladder.score") %>%
    slice_max(score, n = 10) %>%
    mutate(cat = 'top_10',
           rank = rank(-score),
           country_label = paste0(country, ' (', rank, ')'))
```



Least 10 happiest countries

```
# Let's retrieve least 10 happiest countries based on Ladder.score
df_2021_bottom10 = df_2021_tranformed %>%
    filter(dimension == "Ladder.score") %>%
    mutate(rank = rank(score),
           country_label = paste0(country, ' (', rank, ')')) %>%
    slice_min(score, n = 10) %>%
    mutate(cat = 'bottom_10')
```



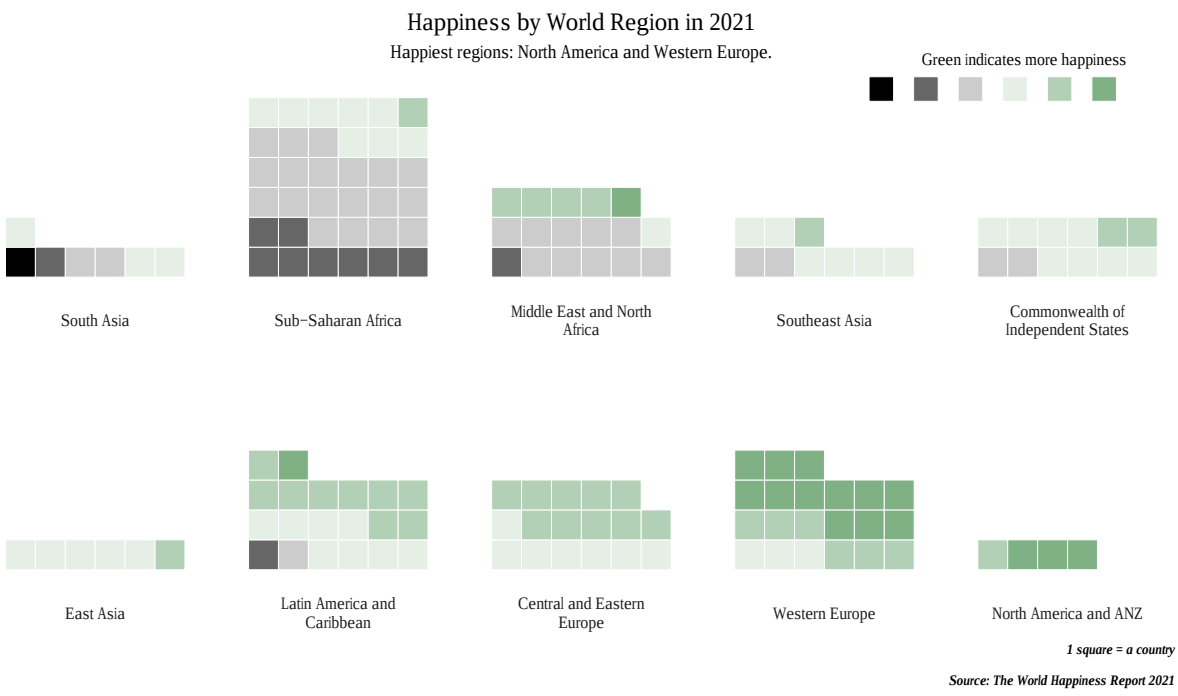
World happiness by regions

```
# map country to regions
country_region_dict = df_2021 %>%
  select(
    country = Country.name,
    region = Regional.indicator) %>%
  unique()

head(country_region_dict)
```

```
## # A tibble: 6 x 2
##   country     region
```

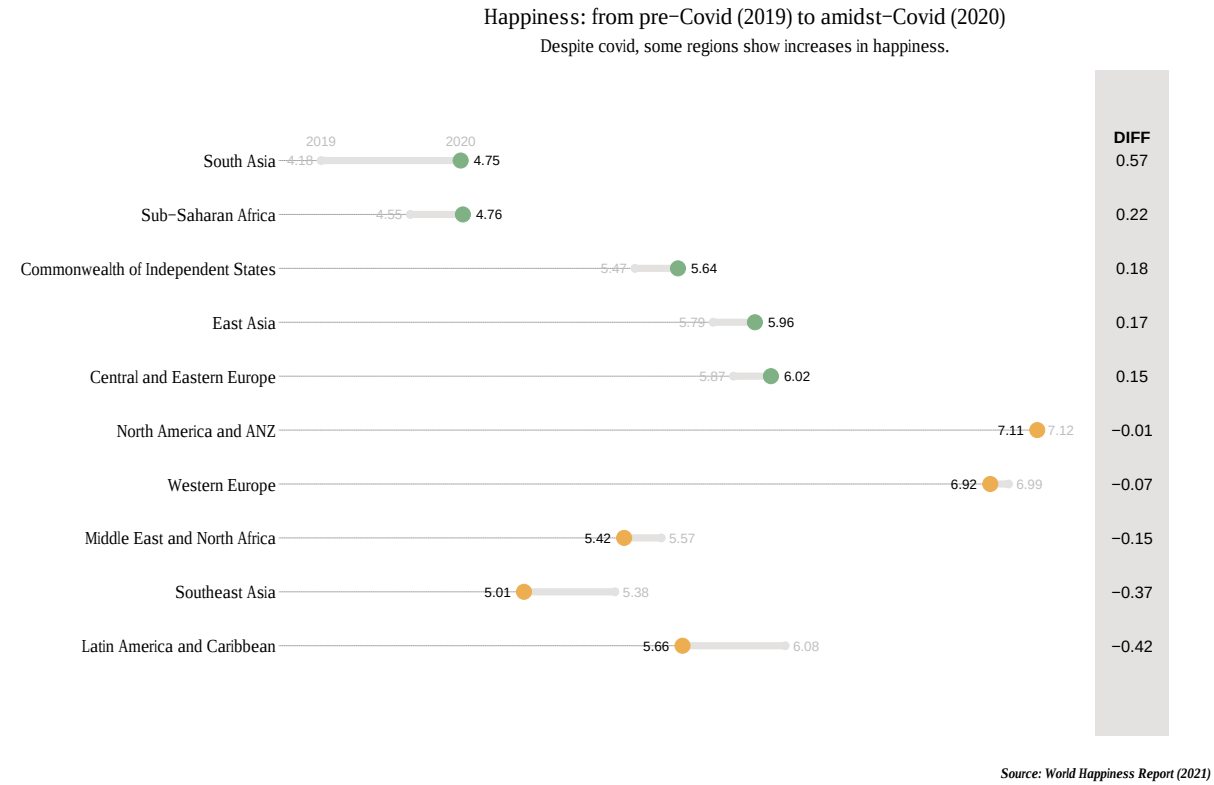
```
##   <chr>       <chr>
## 1 Finland     Western Europe
## 2 Denmark     Western Europe
## 3 Switzerland Western Europe
## 4 Iceland     Western Europe
## 5 Netherlands Western Europe
## 6 Norway      Western Europe
```

```r
# Using the transformed data for 2021 which has the format
# country, dimension, score, min, max, percentage and region
# Let's plot only the Ladder score for each country and group them by regions
# We saw that the min score is above 2 and max score is less than 8
# Hence let's add a new column that tells us the bucket where score falls into...
df_region_happiness = df_2021_tranformed %>%
    filter(dimension == 'Ladder.score') %>%
    left_join(country_region_dict, by = 'country') %>%
    mutate(score_bin = cut(score, seq(2,8, 1), right = FALSE)) %>%
    group_by(region) %>%
    mutate(region_avg = mean(score)) %>%
    ungroup() %>%
    mutate(region = reorder(region, region_avg)) %>%
    count(region, score_bin) %>%
    arrange(score_bin, n)

score_levels = levels(df_region_happiness$score_bin)
```



Happiness under Covid-19 during 2019 and 2020

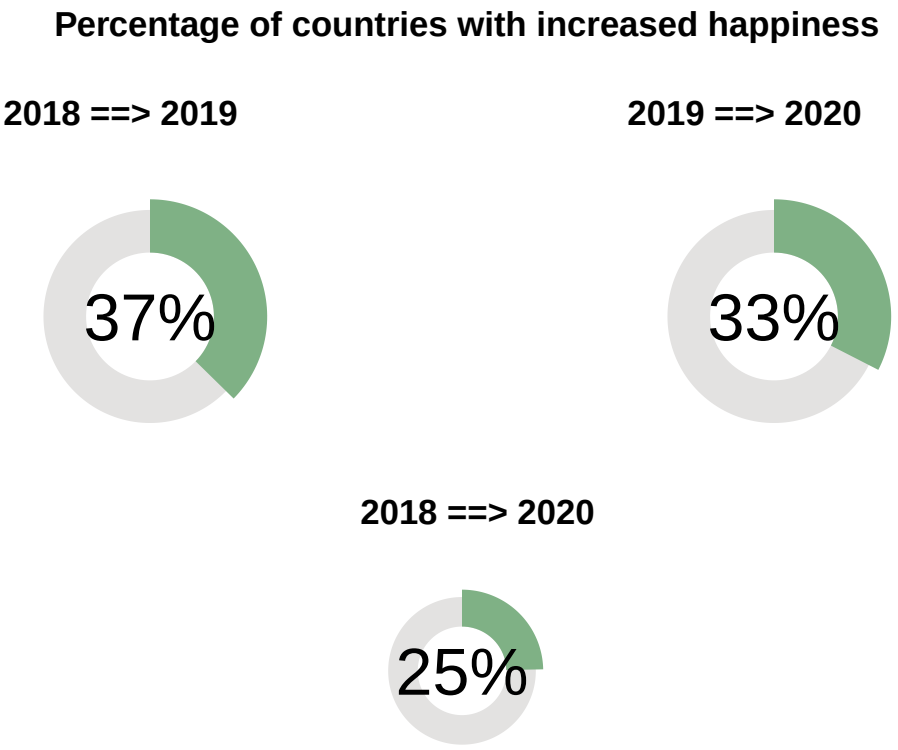
```
df_2019_2020 = df_prev %>%
    filter(year >= 2019) %>%
    left_join(
      country_region_dict,
      by = c('Country.name' = 'country')) %>%
    select(
      country = Country.name,
      region,
      year,
      ladder = Ladder.score)  %>%
    pivot_wider(
      names_from = 'year',
      names_prefix = 'year',
      values_from = 'ladder') %>%
    filter(
      !is.na(year2019) & !is.na(year2020)) %>%
    group_by(region) %>%
    summarize(happiness_2019 = mean(year2019, na.rm = TRUE),
              happiness_2020 = mean(year2020, na.rm = TRUE)) %>%
    mutate(diff = happiness_2020 - happiness_2019) %>%
    arrange(diff) %>%
    mutate(region = factor(region, levels = region))
```



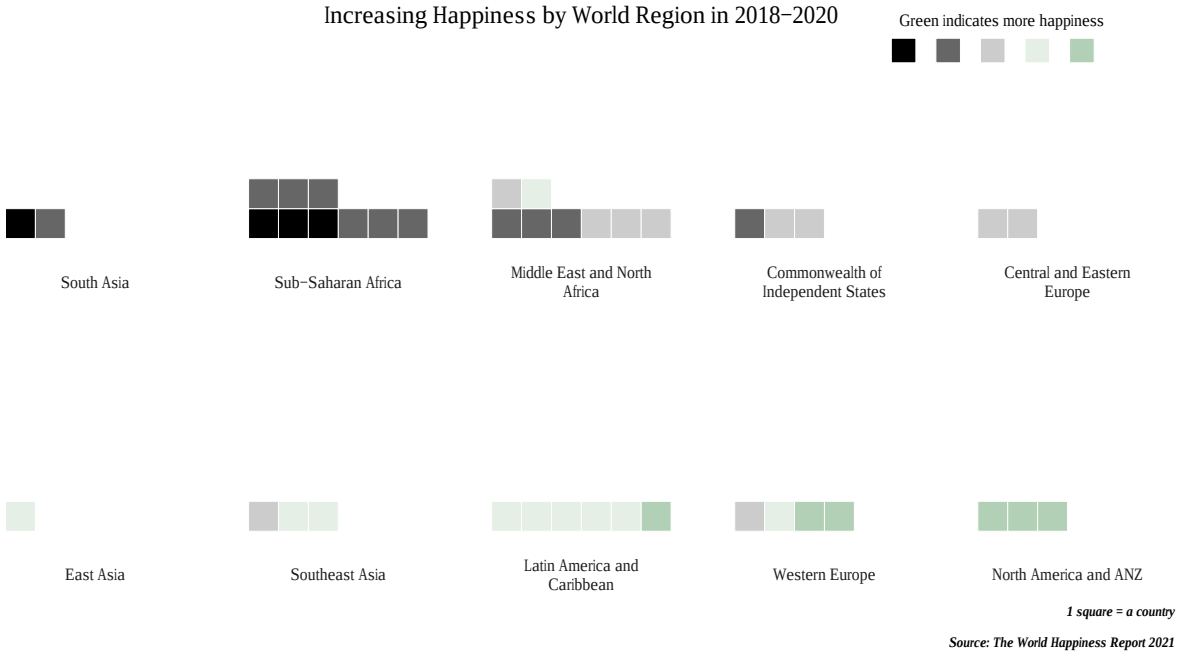
Countries with increased happiness

Let's see how many countries increased happiness from 2019->2020. For this analysis, we have to fill the missing values for the years 2018-2020. In this project, mean of the score of each country is replaced into the missing values.

```
df_countries_increased_happiness = df_total_ladder_wider %>%
  rowwise() %>%
  mutate(
    `2018` = mean(c_across(where(is.numeric)), na.rm = TRUE),
    `2019` = mean(c_across(where(is.numeric)), na.rm = TRUE),
    `2020` = mean(c_across(where(is.numeric)), na.rm = TRUE)) %>%
  pivot_longer(!Country.name, names_to = "year", values_to = "Ladder.score") %>%
  filter(year >= 2018 & year < 2021) %>%
    left_join(
      country_region_dict,
      by = c('Country.name' = 'country')) %>%
    select(
      country = Country.name,
      year,
      ladder = Ladder.score)  %>%
    pivot_wider(
      names_from = 'year',
      names_prefix = 'year',
      values_from = 'ladder') %>%
    mutate(
      increase_in_2019 = ifelse(year2019>year2018, 1, 0),
      increase_in_2020 = ifelse(year2020>year2019, 1, 0))

df_increase_in_2019 = df_countries_increased_happiness %>%
  summarize(pct = mean(increase_in_2019, na.rm = TRUE))

df_increase_in_2020 = df_countries_increased_happiness %>%
  summarize(pct = mean(increase_in_2020, na.rm = TRUE))

df_increase_in_2019_2020 = df_countries_increased_happiness %>%
  mutate(increase = ifelse(increase_in_2019&increase_in_2020, 1, 0))%>%
  summarize(pct = mean(increase, na.rm = TRUE))
```



Source: World Happiness Report (2021)

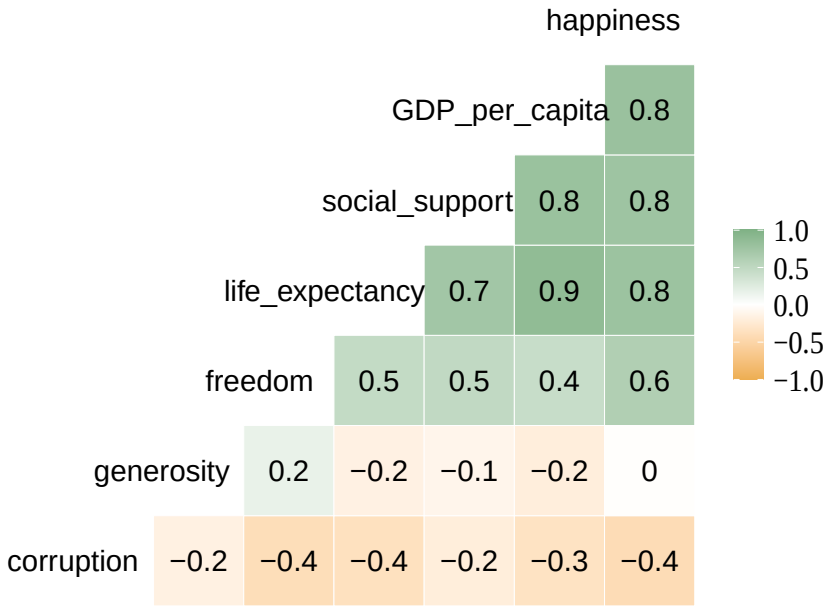


Correlation Matrix

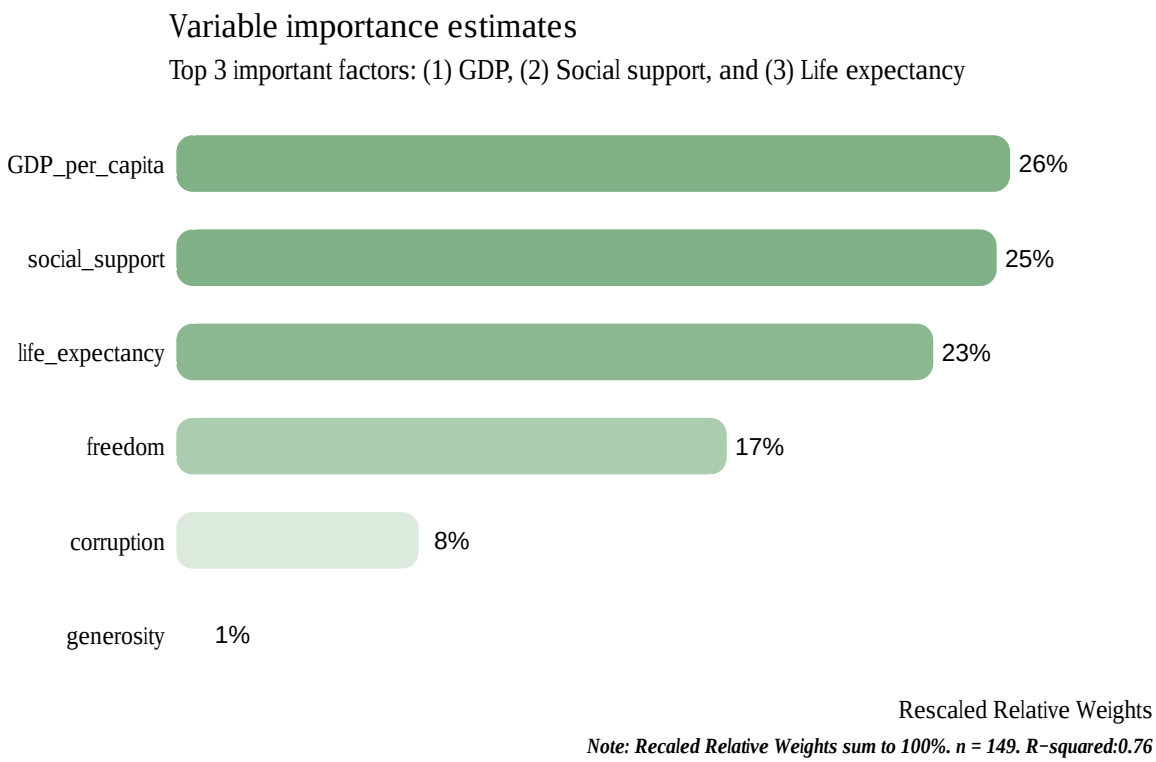
Let's see which factors most strongly correlate with happiness.

Correlation Matrix

Happiness most strongly correlates with:
(1) wealth (GDP),
(2) social support,
(3) health, and
(4) freedom



Key driver analysis



Insights

The North America and Western Europe regions have the most happy regions in the world. Despite Covid-19, about one third of the countries in the world see an increase in happiness from 2019 to 2020. Three top drivers of happiness: (1) Wealth (2) Social support (3) Health

Appendix

ref: https://happiness-report.s3.amazonaws.com/2021/Appendix1WHR2021C2.pdf

Happiness (ladder)

> Please imagine a ladder, with steps numbered from 0 at the bottom to 10 at the top. The top of the ladder represents the best possible life for you and the bottom of the ladder represents the worst possible life for you. On which step of the ladder would you say you personally feel you stand at this time?"

Six major factors to explain happiness

- GDP per capita: > The statistics of GDP per capita (variable name gdp) in purchasing power parity (PPP) at constant 2017 international dollar prices are from the October 14, 2020 update of the World Development Indicators (WDI)

- Healthy Life Expectancy (HLE) > Healthy life expectancies at birth are based on the data extracted from the World Health Organization's (WHO) Global Health Observatory data repository
- Social support > National average of the binary responses (either 0 or 1) to the GWP question "If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?
- Freedom to make life choices > National average of responses to the GWP question "Are you satisfied or dissatisfied with your freedom to choose whatyou do with your life?
- Generosity > The residual of regressing national average of response to the GWP question "Have you donated money to a charity in the past month?" on GDP per capita.
- Corruption Perception > The measure is the national average of the survey responses to two questions in the GWP: "Is corruption widespread throughout the government or not" and "Is corruption widespread within businesses or not?" The overall perception is just the average of the two 0-or-1 responses. In case the perception of government corruption is missing, we use the perception of business corruption as the overall perception.