

Are Danes having fun in Denmark?

Semester project for Statistics course.

Author: Adam Wolkowyci

```
library(plyr)
# install.packages("GGally")
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.6      v dplyr  1.0.7
## v tidyr   1.1.4      v stringr 1.4.0
## v readr   2.1.0      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()  masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()

library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

library(ggthemes)
options(repr.plot.width=32, repr.plot.height=20)

read_csv("./archive/world-happiness-report-2021.csv") -> data_2021_raw

## Rows: 149 Columns: 20

## -- Column specification -----
## Delimiter: ","
## chr  (2): Country name, Regional indicator
## dbl (18): Ladder score, Standard error of ladder score, upperwhisker, lowerw...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
read_csv("./archive/world-happiness-report.csv") -> data_all_raw

## Rows: 1949 Columns: 11

## -- Column specification -----
## Delimiter: ","
## chr  (1): Country name
## dbl (10): year, Life Ladder, Log GDP per capita, Social support, Healthy lif...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(data_2021_raw)
```

```
## # A tibble: 6 x 20
##   `Country name` `Regional indica~ `Ladder score` `Standard error ~ 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>, ...
```

```
head(data_all_raw)
```

```
## # A tibble: 6 x 11
##   `Country name` year `Life Ladder` `Log GDP per capita` `Social support`
##   <chr>          <dbl>          <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>
```

Combine two dataframes

```
# Changing the column names of data_all_raw and naming the dataframe data_all
data_all_raw %>% select(country = 'Country name', score = 'Life Ladder',
                        economy = 'Log GDP per capita',
                        social_support = 'Social support',
                        life_expectancy = 'Healthy life expectancy at birth',
                        freedom = 'Freedom to make life choices',
                        generosity = 'Generosity',
                        corruption = 'Perceptions of corruption', year) -> data_all
```

```
# Changing the column names of data_2021_raw and naming the dataframe data_all
data_2021_raw %>% select(country = 'Country name', score = 'Ladder score',
                        economy = 'Logged GDP per capita',
                        social_support = 'Social support',
                        life_expectancy = 'Healthy life expectancy',
                        freedom = 'Freedom to make life choices',
```

```

    generosity = 'Generosity',
    corruption = 'Perceptions of corruption',
    region = 'Regional indicator') -> data_2021

data_2021 %>% select(country, region) -> continent

full_join(data_all, continent, by = "country") -> data_all

bind_rows(data_all, data_2021) -> data

data[, "year"][is.na(data[, "year"])] <- 2021

head(data)

## # A tibble: 6 x 10
##   country      score economy social_support life_expectancy freedom generosity
##   <chr>      <dbl>   <dbl>         <dbl>         <dbl>   <dbl>      <dbl>
## 1 Afghanistan 3.72    7.37          0.451          50.8    0.718    0.168
## 2 Afghanistan 4.40    7.54          0.552          51.2    0.679    0.19
## 3 Afghanistan 4.76    7.65          0.539          51.6    0.6      0.121
## 4 Afghanistan 3.83    7.62          0.521          51.9    0.496    0.162
## 5 Afghanistan 3.78    7.70          0.521          52.2    0.531    0.236
## 6 Afghanistan 3.57    7.72          0.484          52.6    0.578    0.061
## # ... with 3 more variables: corruption <dbl>, year <dbl>, region <chr>

```

Denmark

```

dk <- data %>% filter(country == "Denmark")

ggplot(dk, aes(x = year, y = score, label = score)) +
  geom_point(size = 8, color = "black") +
  theme_fivethirtyeight() +
  geom_segment(aes(x = year, xend = year, y = 0, yend = score)) +
  scale_x_continuous(breaks = seq(2005, 2021, 1)) +
  scale_y_continuous(breaks = seq(0, 7, 1)) + geom_text(color = 'white', size = 2) +
  labs(title = "Are Danes having fun in Denmark?",
       subtitle = "Happiness scores of Danes across the years",
       x = "Year", y = "Happiness score") +
  theme(plot.title = element_text(size = 15, face = "bold"),
        plot.subtitle = element_text(size = 10),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8)) +
  coord_flip()

```

Are Danes having fun in Denmark?

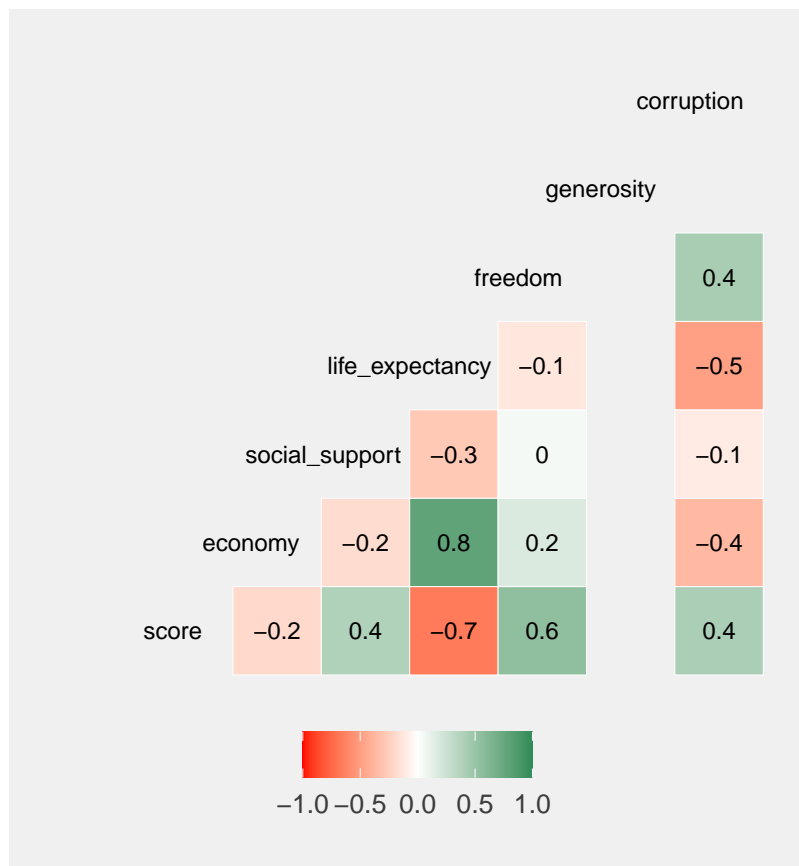
Happiness scores of Danes across the years



Linear relationship between score and other variables

```
dk_corr <- dk %>% select(-c(country, year, region))
```

```
ggcorr(dk_corr,  
  method = c("everything", "pearson"),  
  size = 3, hjust = 0.77,  
  low = "#ff0000", mid = "white", high = "#2e8b57",  
  label = TRUE, label_size = 3,  
  layout.exp = 1) + theme_fivethirtyeight()
```



Note: live expectancy and economy have a strong linear correlation.

Denmark vs Europe

```
happiest_countries_2021 <- data_2021 %>%
  select(country, score) %>%
  group_by(score) %>%
  arrange(desc(score)) %>%
  head(10)
data.frame(happiest_countries_2021)
```

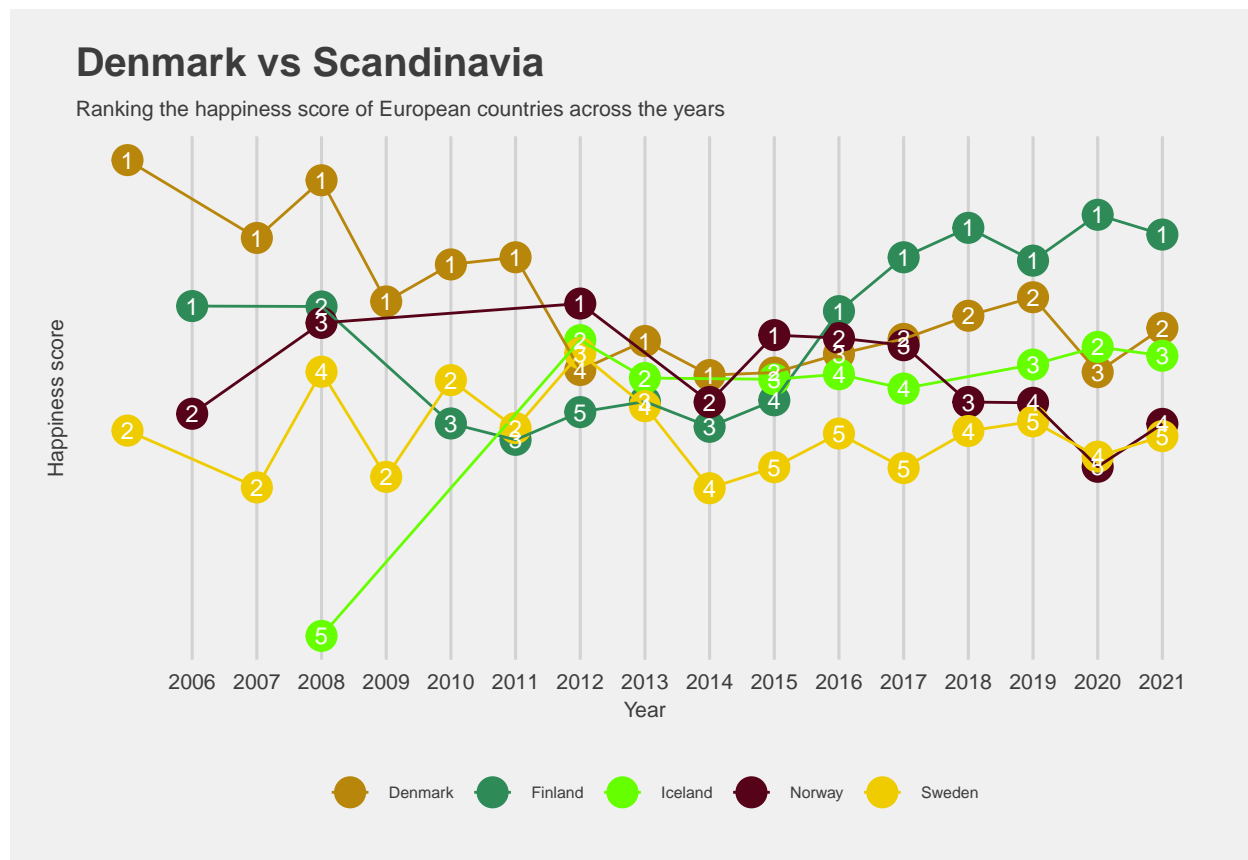
```
##      country score
## 1    Finland 7.842
## 2    Denmark 7.620
## 3 Switzerland 7.571
## 4    Iceland 7.554
## 5 Netherlands 7.464
## 6    Norway 7.392
## 7    Sweden 7.363
## 8  Luxembourg 7.324
## 9 New Zealand 7.277
## 10   Austria 7.268
```

Denmark vs Scandinavia

```
scandinavia <- data %>% filter(country == 'Denmark' | country == 'Norway' |
                             country == 'Sweden' | country == 'Finland' |
                             country == 'Iceland') %>%

  group_by(year) %>%
  mutate(yrrank = row_number(-score))

ggplot(scandinavia, aes(x = year, y = score, label = yrrank)) +
  geom_point(aes(color = country), size = 5) +
  theme_fivethirtyeight() +
  geom_line(aes(color = country)) + scale_x_continuous(breaks = seq(2006, 2021, 1)) +
  scale_y_continuous(breaks = seq(0, 5, 0.5)) + geom_text(color = 'white', size = 3) +
  scale_color_manual(values = c("#b8860b", "#2e8b57", "#66ff00", "#560319", "#efcc00")) +
  labs(title = 'Denmark vs Scandinavia',
       subtitle = 'Ranking the happiness score of European countries across the years',
       x = 'Year', y = 'Happiness score') +
  theme(plot.title = element_text(size = 15, face = 'bold'),
        plot.subtitle = element_text(size = 8),
        axis.title = element_text(size = 8),
        axis.text = element_text(size = 8),
        legend.direction = 'horizontal', legend.position = 'bottom',
        legend.title = element_blank(),
        legend.text = element_text(size = 6))
```



Denmark vs World

```
data %>%
  mutate(region = case_when(country == 'Denmark' ~ 'Denmark', TRUE ~ region)) -> data

unique(data['region'])

## # A tibble: 12 x 1
##   region
##   <chr>
## 1 South Asia
## 2 Central and Eastern Europe
## 3 Middle East and North Africa
## 4 <NA>
## 5 Latin America and Caribbean
## 6 Commonwealth of Independent States
## 7 North America and ANZ
## 8 Western Europe
## 9 Sub-Saharan Africa
## 10 Southeast Asia
## 11 East Asia
## 12 Denmark

colSums(is.na(data))

##           country           score           economy  social_support life_expectancy
##           0             0             36             13             55
##    freedom generosity corruption           year           region
##           32             89             110            0             63

data %>% filter(is.na(region)) -> na_region

data %>%
  mutate(
    region = case_when(country == "Angola" | country == "Central African Republic" |
      country == "Congo (Kinshasa)" | country == "Djibouti" |
      country == "Somalia" | country == "Somaliland region" |
      country == "South Sudan" | country == "Sudan"
      ~ "Sub-Saharan Africa", TRUE ~ region)) %>%
  mutate(
    region = case_when(country == "Belize" | country == "Cuba" | country == "Guyana" |
      country == "Suriname" | country == "Trinidad and Tobago"
      ~ "Latin America and Caribbean", TRUE ~ region)) %>%
  mutate(
    region = case_when(country == "Oman" | country == "Qatar" | country == "Syria"
      ~ "Middle East and North Africa", TRUE ~ region)) %>%
  mutate(
    region = case_when(country == "Bhutan" ~ "South Asia", TRUE ~ region)) -> data

data %>% filter(is.na(region)) -> na_region

revalue(data$region, c("Central and Eastern Europe" = "Europe and North Asia")) ->
  data$region
revalue(data$region, c("Commonwealth of Independent States" = "Europe and North Asia")) ->
  data$region
```

```

revalue(data$region, c("East Asia" = "West, East, Southeast Asia and North Africa")) ->
  data$region
revalue(data$region, c("Southeast Asia" = "West, East, Southeast Asia and North Africa")) ->
  data$region
revalue(data$region, c("Middle East and North Africa" = "West, East, Southeast Asia and North Africa"))
  data$region

unique(data["region"])

```

```

## # A tibble: 8 x 1
##   region
##   <chr>
## 1 South Asia
## 2 Europe and North Asia
## 3 West, East, Southeast Asia and North Africa
## 4 Sub-Saharan Africa
## 5 Latin America and Caribbean
## 6 North America and ANZ
## 7 Western Europe
## 8 Denmark

```

```

data %>%
  select(year, region, score) %>%
  group_by(year, region) %>%
  summarise(avg_score = mean(score)) %>%
  mutate_at(vars(region), factor) %>%
  group_by(year) %>%
  mutate(yrrank = row_number(-avg_score)) -> avg_region

```

```

## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
head(avg_region)

```

```

## # A tibble: 6 x 4
## # Groups:   year [1]
##   year region          avg_score yrrank
##   <dbl> <fct>          <dbl>   <int>
## 1  2005 Denmark            8.02     1
## 2  2005 Europe and North Asia  5.57     6
## 3  2005 Latin America and Caribbean  6.80     4
## 4  2005 North America and ANZ    7.38     2
## 5  2005 South Asia            5.22     7
## 6  2005 West, East, Southeast Asia and North Africa  5.80     5

```

```

worldmap = map_data('world') %>% filter(region != 'Antarctica')

merged_data <- merge(x = worldmap, y = data_2021,
                     by.x = 'region', by.y = 'country', all.x = TRUE) %>%
  arrange(order)
ggplot(merged_data, aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = score)) +

theme_fivethirtyeight() +
  scale_fill_continuous(low = 'white', high = '#2e8b57') +
  labs(title = "World's Average Happiness",

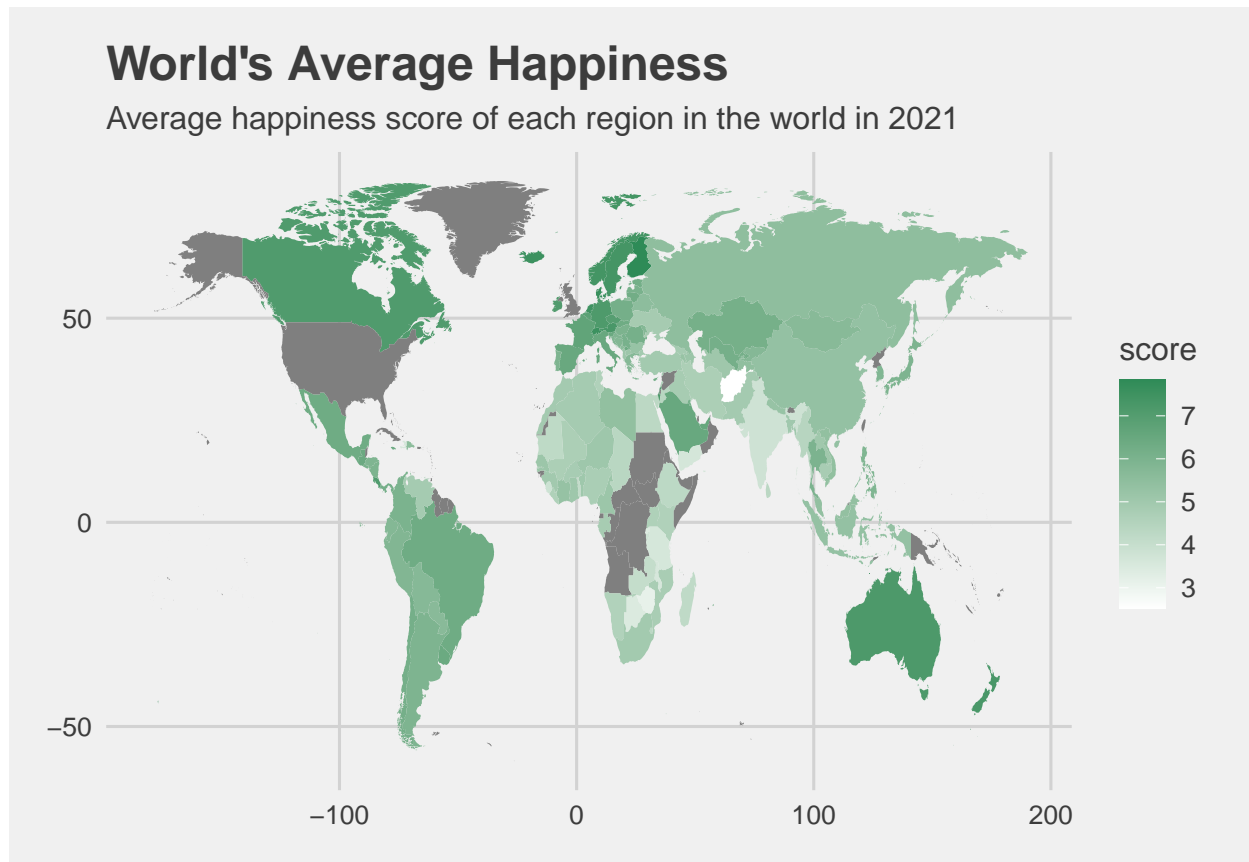
```



```

    subtitle = 'Average happiness score of each region in the world in 2021') +
  theme(legend.direction = 'vertical', legend.position = 'right',
        legend.text = element_text(size = 10))

```



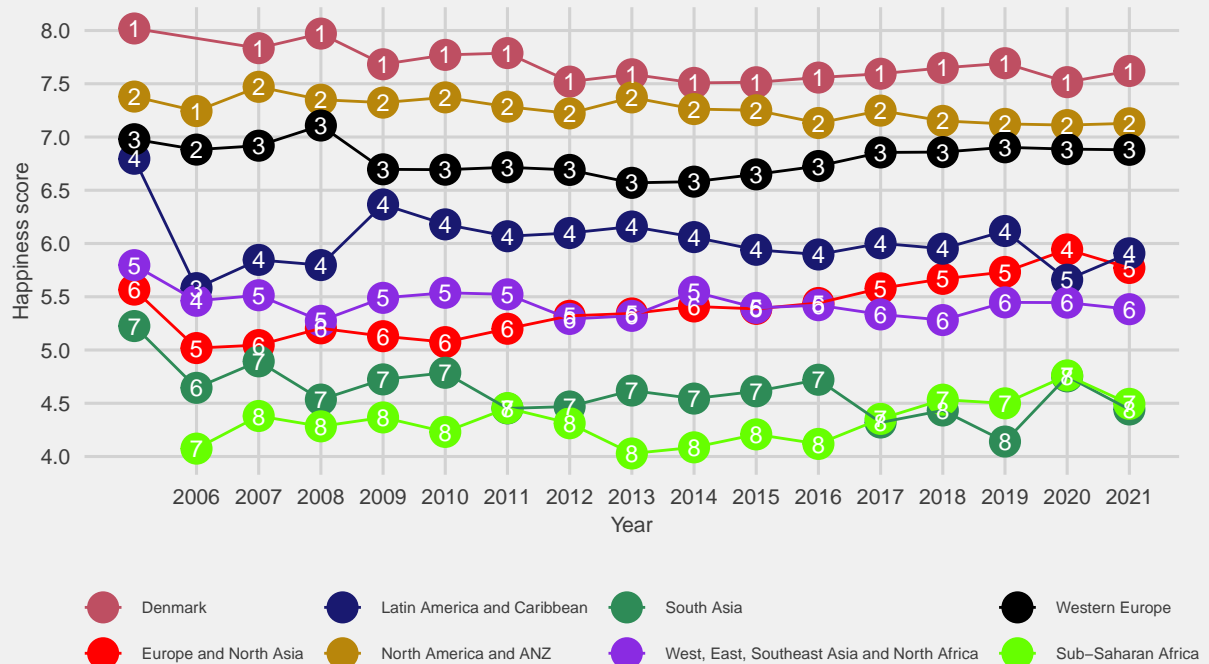
```

ggplot(avg_region, aes(x = year, y = avg_score, label = yrrank)) +
  theme_fivethirtyeight() +
  geom_point(aes(color = region), size = 5) +
  geom_line(aes(color = region)) + scale_x_continuous(breaks = seq(2006, 2021, 1)) +
  scale_y_continuous(breaks = seq(0, 8, 0.5)) + geom_text(color = 'white', size = 3) +
  scale_color_manual(values = c("#be4f62", "#ff0000", "#191970", "#b8860b",
                                "#2e8b57", "#8a2be2", "#000000", "#66ff00")) +
  labs(title = 'Denmark vs World',
        subtitle = 'Average happiness score of each region in the world across the years',
        x = 'Year', y = 'Happiness score') +
  theme(plot.title = element_text(size = 15, face = 'bold'),
        axis.title = element_text(size = 8),
        axis.text = element_text(size = 8),
        legend.direction = 'horizontal', legend.position = 'bottom',
        legend.title = element_blank(),
        legend.text = element_text(size = 6))

```

Denmark vs World

Average happiness score of each region in the world across the years



Denmark vs World in 2021

```
data %>% filter(year == 2021) -> data_2021_final
```

```
colSums(is.na(data_2021_final))
```

```
##      country      score      economy  social_support  life_expectancy
##      0            0            0            0            0
##      freedom    generosity    corruption            year            region
##      0            0            0            0            0
```

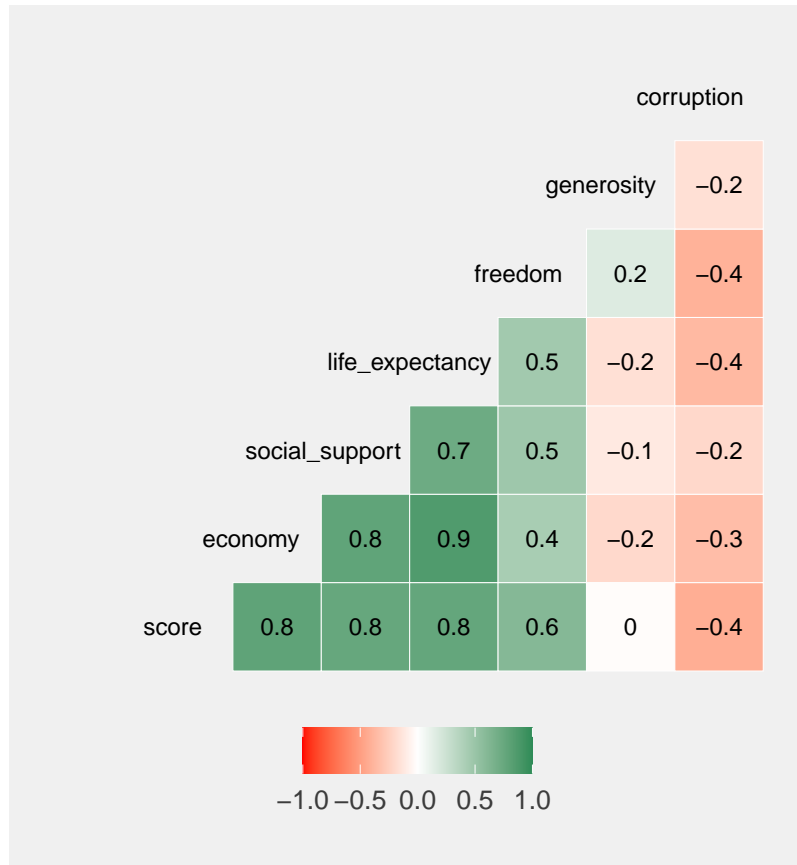
```
head(data_2021_final)
```

```
## # A tibble: 6 x 10
##   country      score economy social_support life_expectancy freedom generosity
##   <chr>      <dbl>  <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Finland    7.84    10.8      0.954        72      0.949    -0.098
## 2 Denmark    7.62    10.9      0.954        72.7    0.946     0.03
## 3 Switzerland 7.57    11.1      0.942        74.4    0.919     0.025
## 4 Iceland    7.55    10.9      0.983        73      0.955     0.16
## 5 Netherlands 7.46    10.9      0.942        72.4    0.913     0.175
## 6 Norway     7.39    11.1      0.954        73.3    0.96      0.093
## # ... with 3 more variables: corruption <dbl>, year <dbl>, region <chr>
```

```
data_2021_final %>% select(-c(country, year, region)) -> data_2021_corr
```

```
ggcorr(data_2021_corr,
```

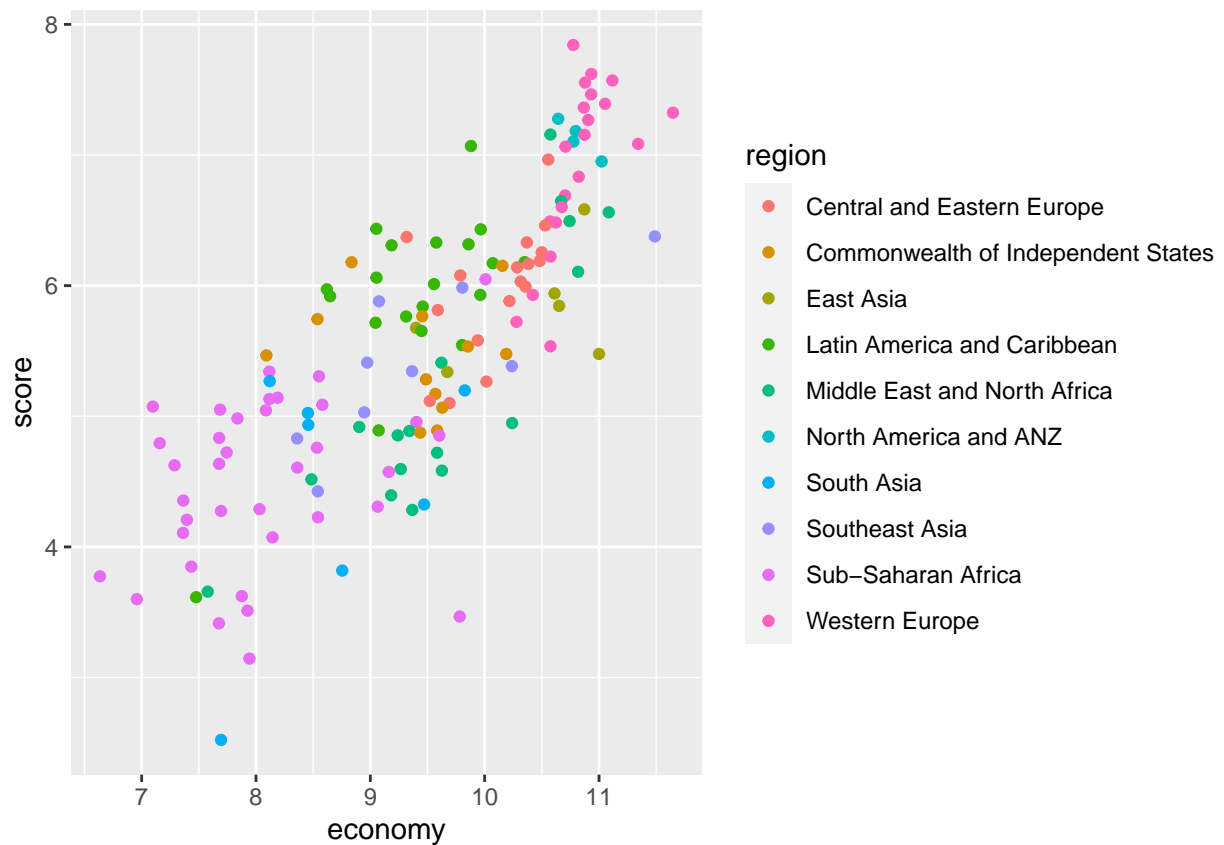
```
method = c("everything", "pearson"),
size = 3, hjust = 0.77,
low = "#ff0000", mid = "white", high = "#2e8b57",
label = TRUE, label_size = 3,
layout.exp = 1) + theme_fivethirtyeight()
```



Note: Economy, social support and life expectancy are highly correlated to score.

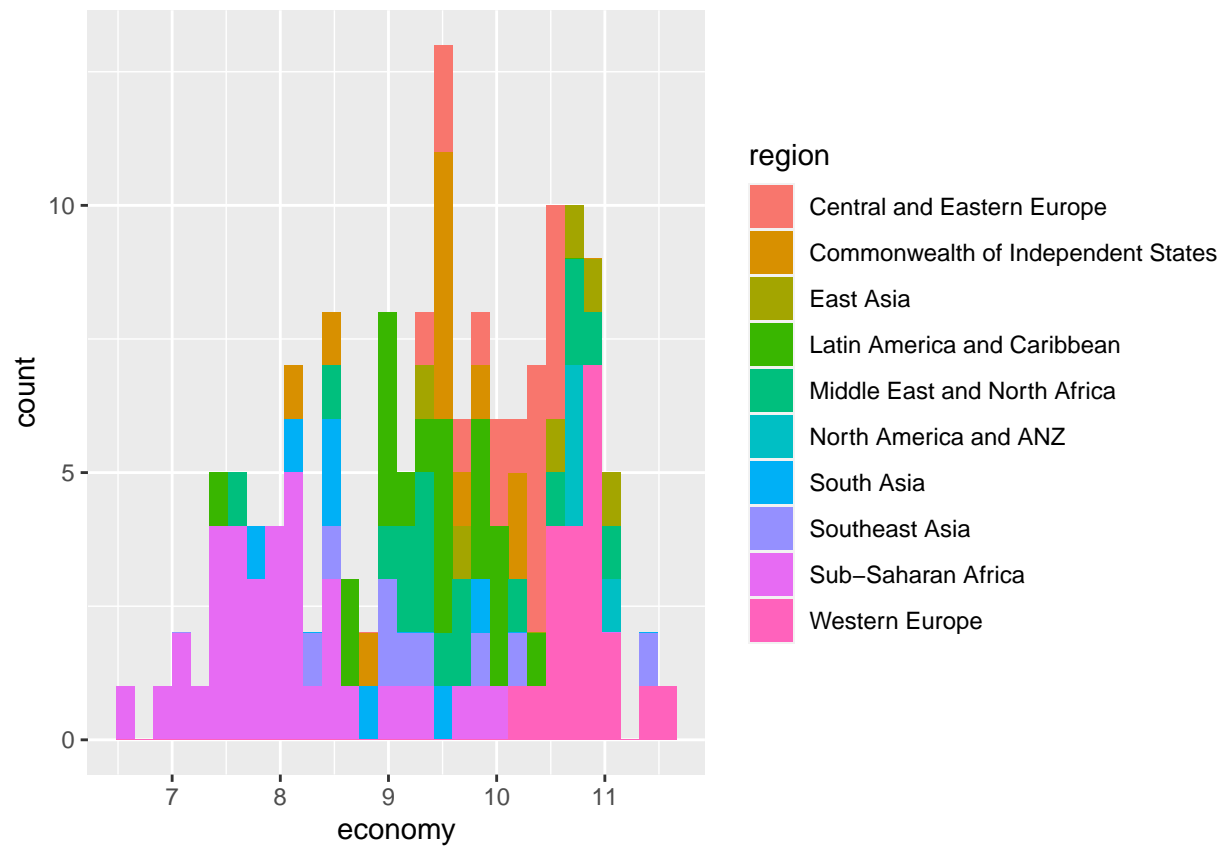
Inference

```
ggplot(data_2021) +
  geom_point(aes(x = economy, y = score, color = region))
```

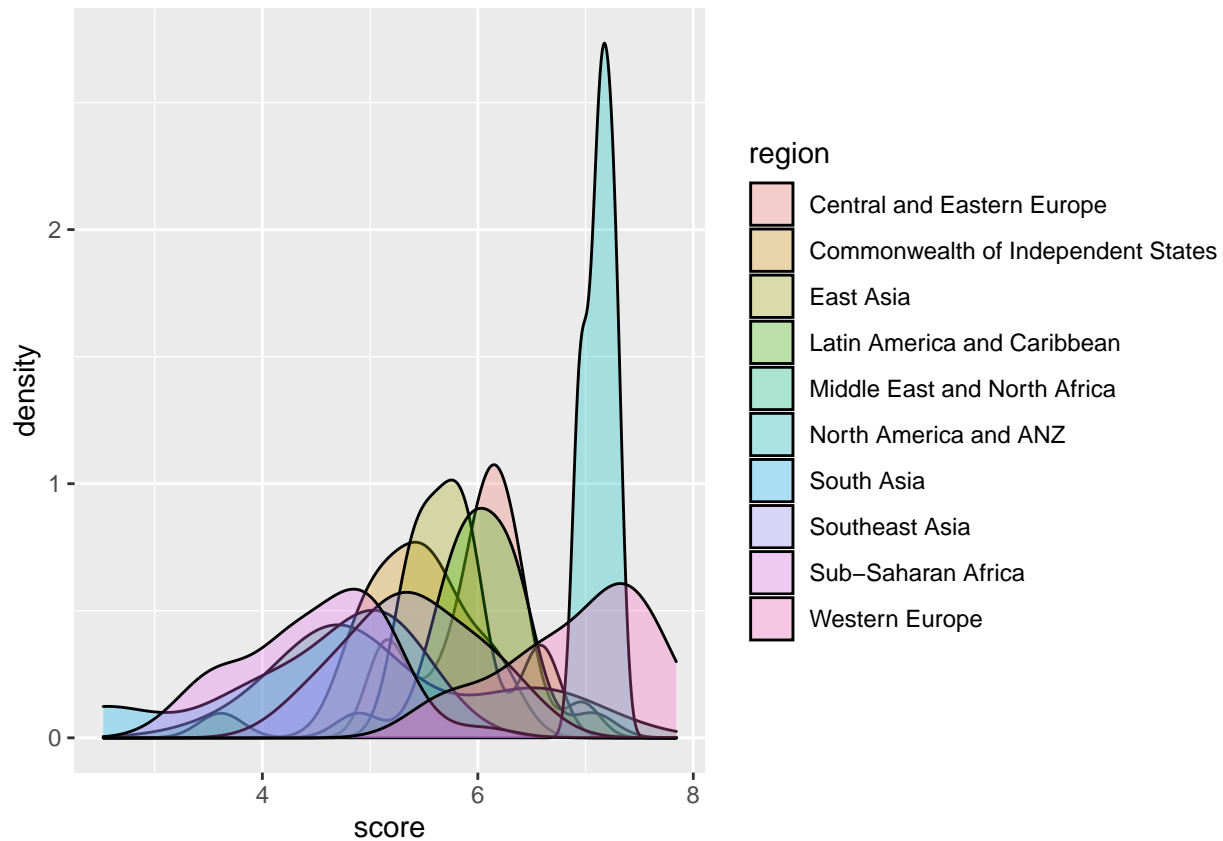


```
ggplot(data_2021) +
  geom_histogram(aes(x = economy, fill = region))
```

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



```
ggplot(data_2021) +  
  geom_density(aes(x = score, fill = region), alpha = 0.3)
```



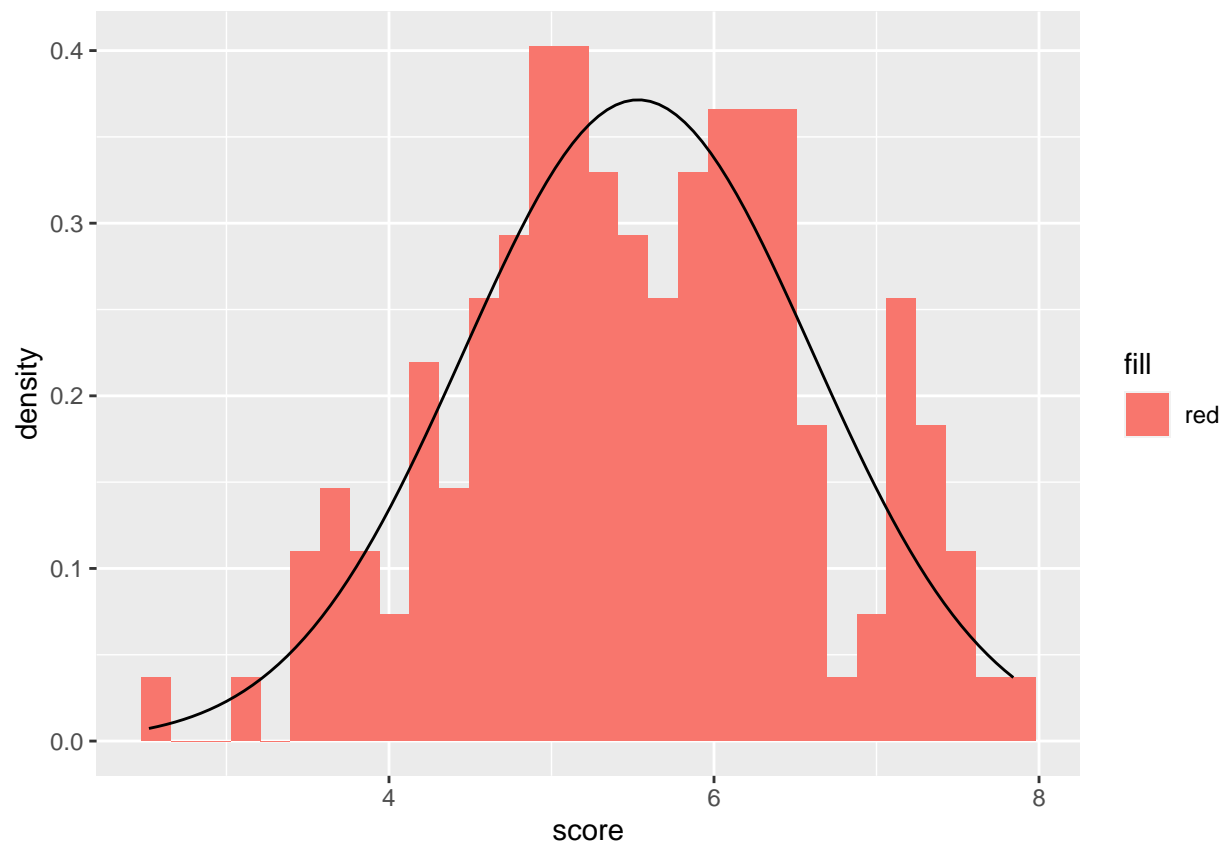
```
xframe <- seq(-10, 10, length = 100)
dnorm(xframe)
```

```
## [1] 7.694599e-23 5.684422e-22 4.031458e-21 2.744818e-20 1.794074e-19
## [6] 1.125752e-18 6.781419e-18 3.921696e-17 2.177222e-16 1.160399e-15
## [11] 5.937273e-15 2.916369e-14 1.375223e-13 6.225578e-13 2.705587e-12
## [16] 1.128805e-11 4.521180e-11 1.738442e-10 6.417178e-10 2.274068e-09
## [21] 7.736391e-09 2.526672e-08 7.921998e-08 2.384493e-07 6.890219e-07
## [26] 1.911373e-06 5.090183e-06 1.301358e-05 3.194006e-05 7.525757e-05
## [31] 1.702316e-04 3.696626e-04 7.706310e-04 1.542279e-03 2.963159e-03
## [36] 5.465405e-03 9.677547e-03 1.645068e-02 2.684588e-02 4.205786e-02
## [41] 6.325461e-02 9.132982e-02 1.265927e-01 1.684535e-01 2.151925e-01
## [46] 2.639062e-01 3.107045e-01 3.511729e-01 3.810395e-01 3.969123e-01
## [51] 3.969123e-01 3.810395e-01 3.511729e-01 3.107045e-01 2.639062e-01
## [56] 2.151925e-01 1.684535e-01 1.265927e-01 9.132982e-02 6.325461e-02
## [61] 4.205786e-02 2.684588e-02 1.645068e-02 9.677547e-03 5.465405e-03
## [66] 2.963159e-03 1.542279e-03 7.706310e-04 3.696626e-04 1.702316e-04
## [71] 7.525757e-05 3.194006e-05 1.301358e-05 5.090183e-06 1.911373e-06
## [76] 6.890219e-07 2.384493e-07 7.921998e-08 2.526672e-08 7.736391e-09
## [81] 2.274068e-09 6.417178e-10 1.738442e-10 4.521180e-11 1.128805e-11
## [86] 2.705587e-12 6.225578e-13 1.375223e-13 2.916369e-14 5.937273e-15
## [91] 1.160399e-15 2.177222e-16 3.921696e-17 6.781419e-18 1.125752e-18
## [96] 1.794074e-19 2.744818e-20 4.031458e-21 5.684422e-22 7.694599e-23
```

```
sd_world <- sd(data_2021$score)
mean_world <- mean(data_2021$score)
ggplot(data_2021) +
  geom_histogram(aes(x = score, y = ..density.., fill = 'red')) +
```

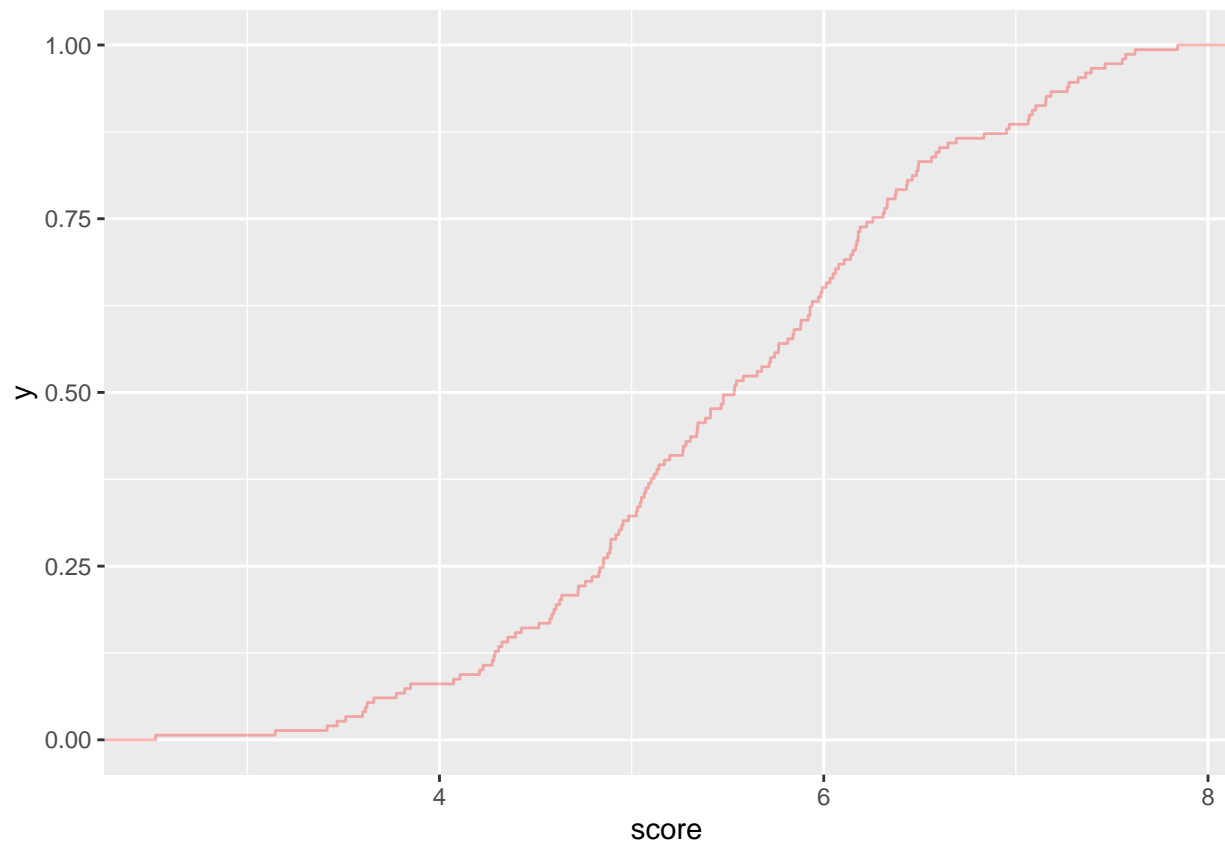
```
stat_function(fun = dnorm, args = list(mean = mean_world, sd = sd_world))
```

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

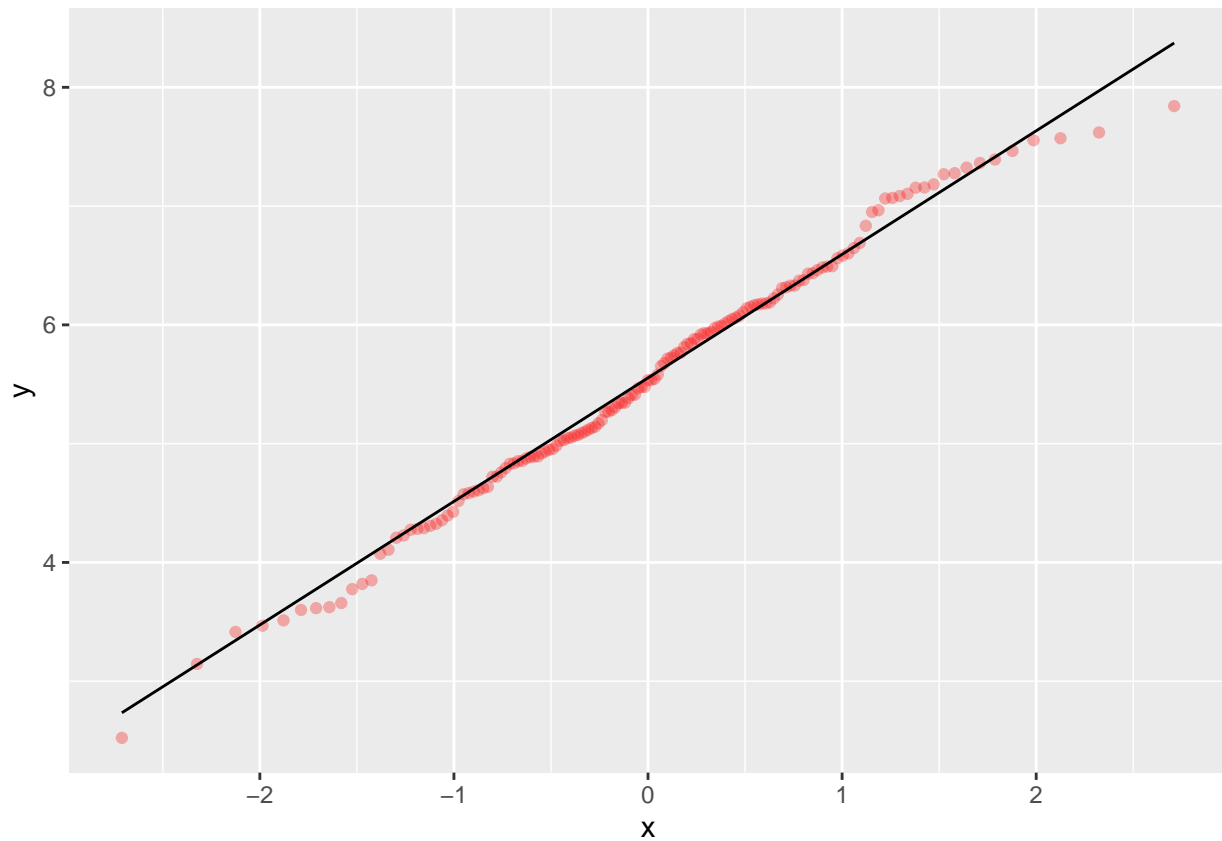


```
ggplot(data_2021) +
  stat_ecdf(aes(x = score), color = 'red', alpha = 0.3) +
  geom_line(stat = 'function', fun = pnorm, args = list(mean = mean, sd = sd))
```

```
## Warning: Computation failed in `stat_function()`:
## Argument nieliczbowy przekazany do funkcji matematycznej
```



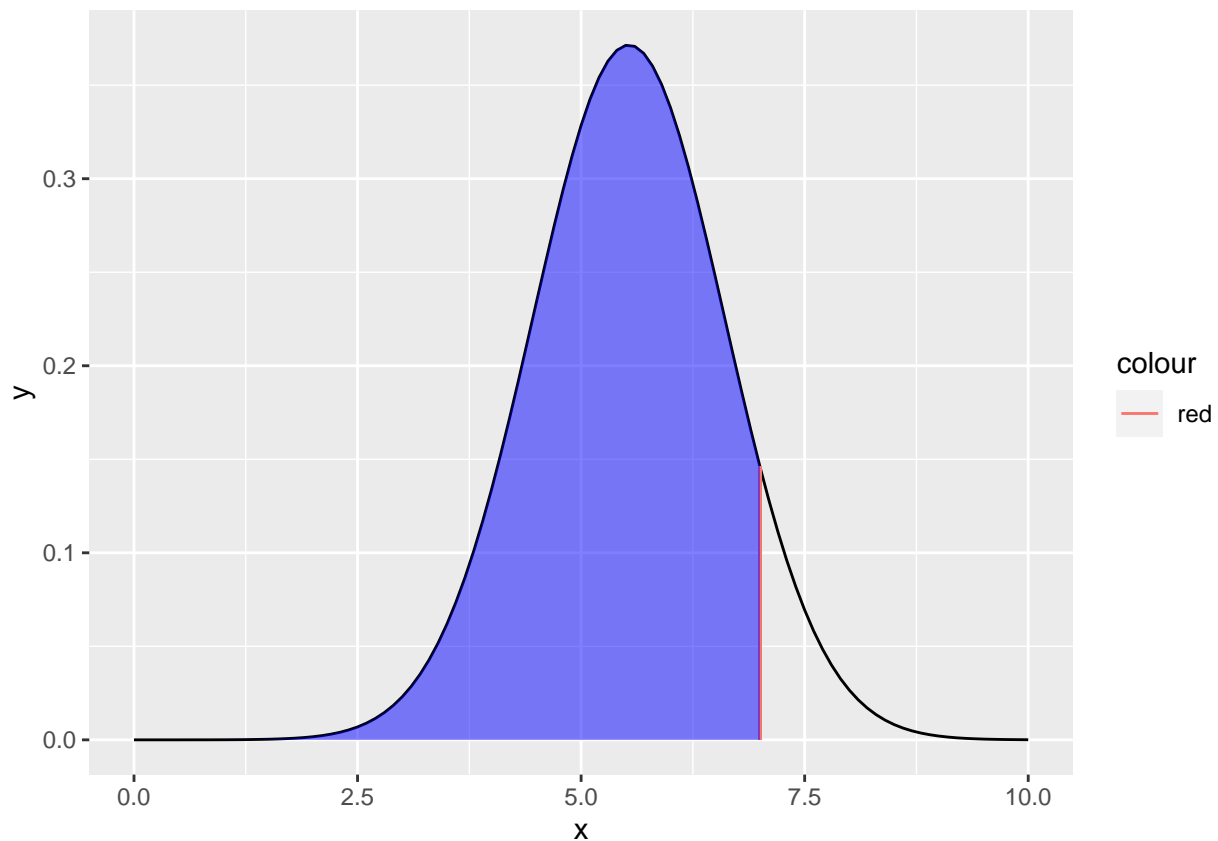
```
ggplot(data_2021) +  
  stat_qq(aes(sample = score), color = 'red', alpha = 0.3) +  
  stat_qq_line(aes(sample = score))
```

Let's assume that the mean happiness is less than 7.

```
x_dash <- 7
z_score <- (x_dash - mean_world) / sd_world

ggplot(data.frame(x = seq(0, 10, length = 500)), aes(x = x)) +
  stat_function(fun = dnorm, args = list(mean = mean_world, sd = sd_world)) +
  geom_segment(aes(x = x_dash, y = 0, xend = x_dash,
                  yend = dnorm(x_dash, mean = mean_world, sd = sd_world), color = 'red')) +
  geom_area(stat = 'function', fun = dnorm, args = list(mean = mean_world, sd = sd_world),
           fill = 'blue', xlim = c(0, x_dash), alpha = 0.5)
```



Regression

```
data_2021 %>%
  mutate(region = case_when(region == 'Central and Eastern Europe' |
                             region == 'Western Europe' ~ 'Europe',
                             TRUE ~ region)) -> data_2021_merged

ggplot(data_2021_merged, aes(x = life_expectancy, y = score)) +
  geom_point(aes(color = region), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = region, fill = region), method = 'lm', fullrange = TRUE) +
  facet_wrap(~region) +
  theme_fivethirtyeight() +
  theme(legend.direction = 'horizontal', legend.position = 'bottom',
        legend.title = element_blank(),
        legend.text = element_text(size = 6))

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

