Happy Planet Index clustering script

HarvardX Data Science Capstone Own Project

Codrin Kruijne

2019-06-03

```
# PLEASE NOTE
# THIS SCRIPT HAS ONLY BEEN TESTED ON A 6-CORE (12-THREAD), 64GB MEMORY MACHINE
# AND WILL PROBABLY NOT RUN WITH LOWER COMPUTER SPECIFICATIONS.
parallel::detectCores()
```

```
## [1] 12
```

```
memory.limit()
```

```
## [1] 65471
```

```
# Script settings
knitr::opts_chunk$set(
    message = FALSE,
    warning = FALSE,
    cache = TRUE,
    tidy.opts = list(width.cutoff = 100),
    tidy = TRUE
)
script_start <- Sys.time()

# Load required packages
library(tidyverse)
library(gridExtra)
library(readxl)
library(tidyimpute)
library(cluster)
```

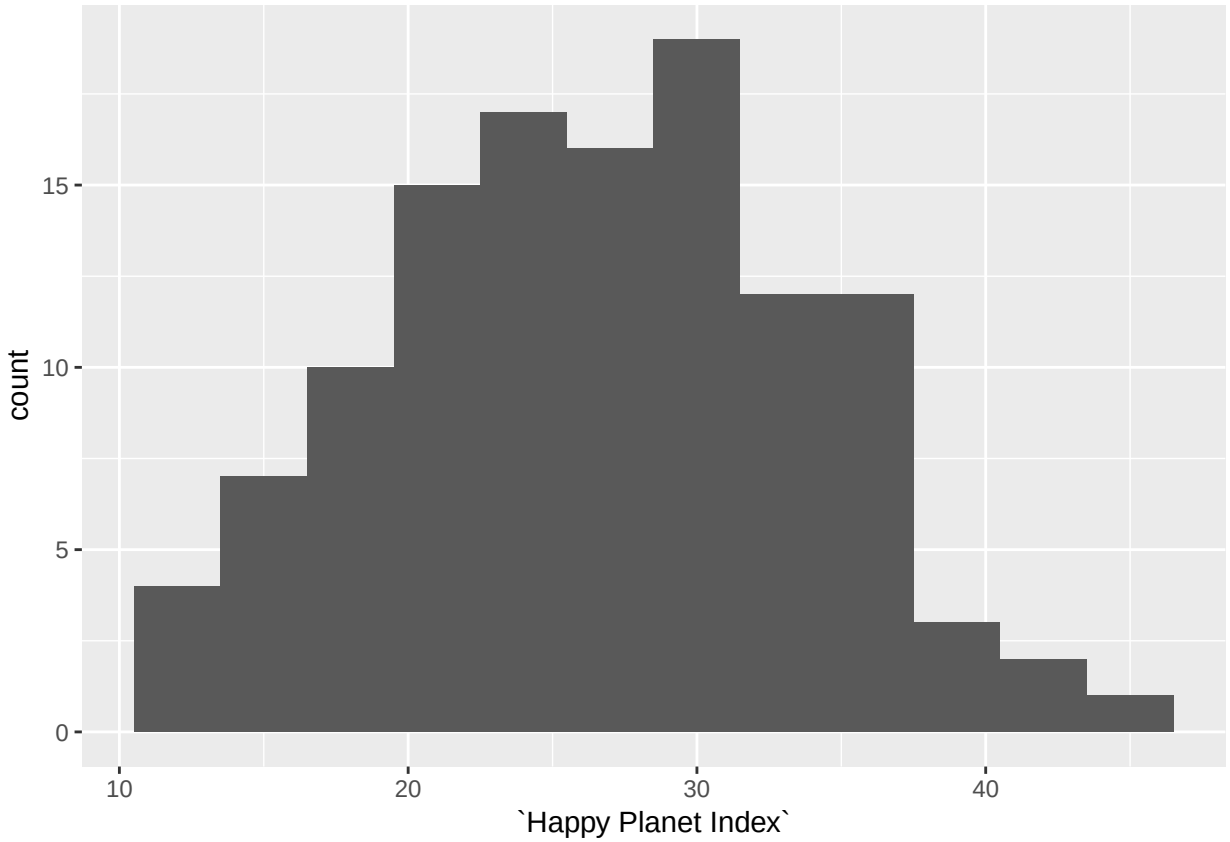
Loading data

```
# Happy Planet Index data from http://happyplanetindex.org/s/hpi-data-2016.xlsx retrieved on 2019-06-01

HPI <- read_excel("Data/hpi-data-2016.xlsx",
                  sheet = "Complete HPI data",
                  range = "C6:N146",
                  trim_ws = TRUE)
HPI[1:2] <- lapply(HPI[1:2], as.factor)

# Social Progress Initiative data from https://www.socialprogress.org/download retrieved in 2019-06-01
SPI <- read_excel("Data/Social Progress Index 2018-Results.xlsx",
```

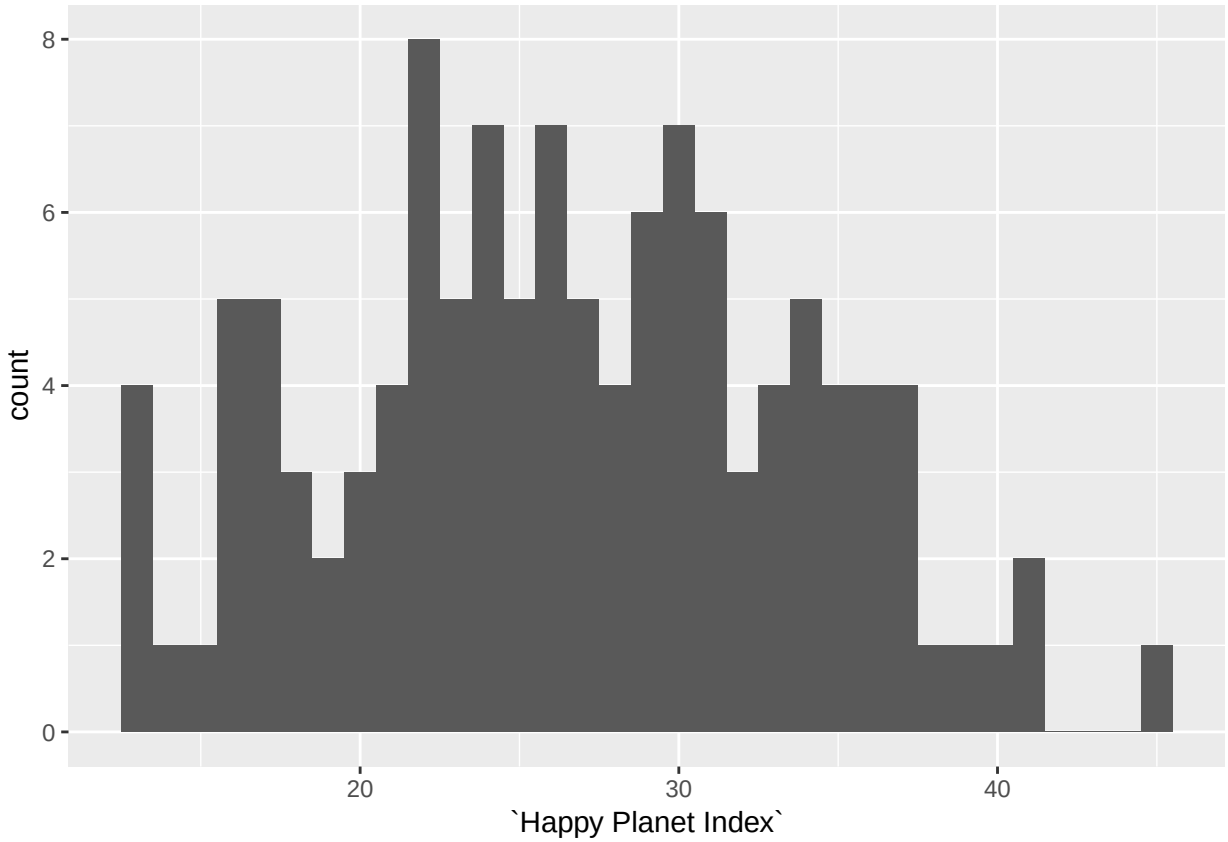
```
                  sheet = "2016",
                  range = "A1:BQ147", # data for countries that are in the index
                  trim_ws = TRUE)
SPI[3:76] <- lapply(SPI[3:69], as.numeric)
SPI[1:2] <- lapply(SPI[1:2], as.factor)
SPI <- SPI %>% select(-Code)
```

```
# Explore differing countries ### FIX DIFFERENT SPELLINGS
diff_countries <- HPI %>% anti_join(SPI, by = "Country") %>% select(Country)

# Join data
raw <- HPI %>% inner_join(SPI, by = "Country") %>%
                mutate(Country = as.factor(Country))

# Impute scaled data
imputed <- raw %>% impute_all(.na = mean, na.rm = TRUE)

# Scale numeric data
scaled <- imputed %>% mutate_if(is.numeric, scale) %>%
                      rename_if(is.numeric, paste, "SCALED")
```

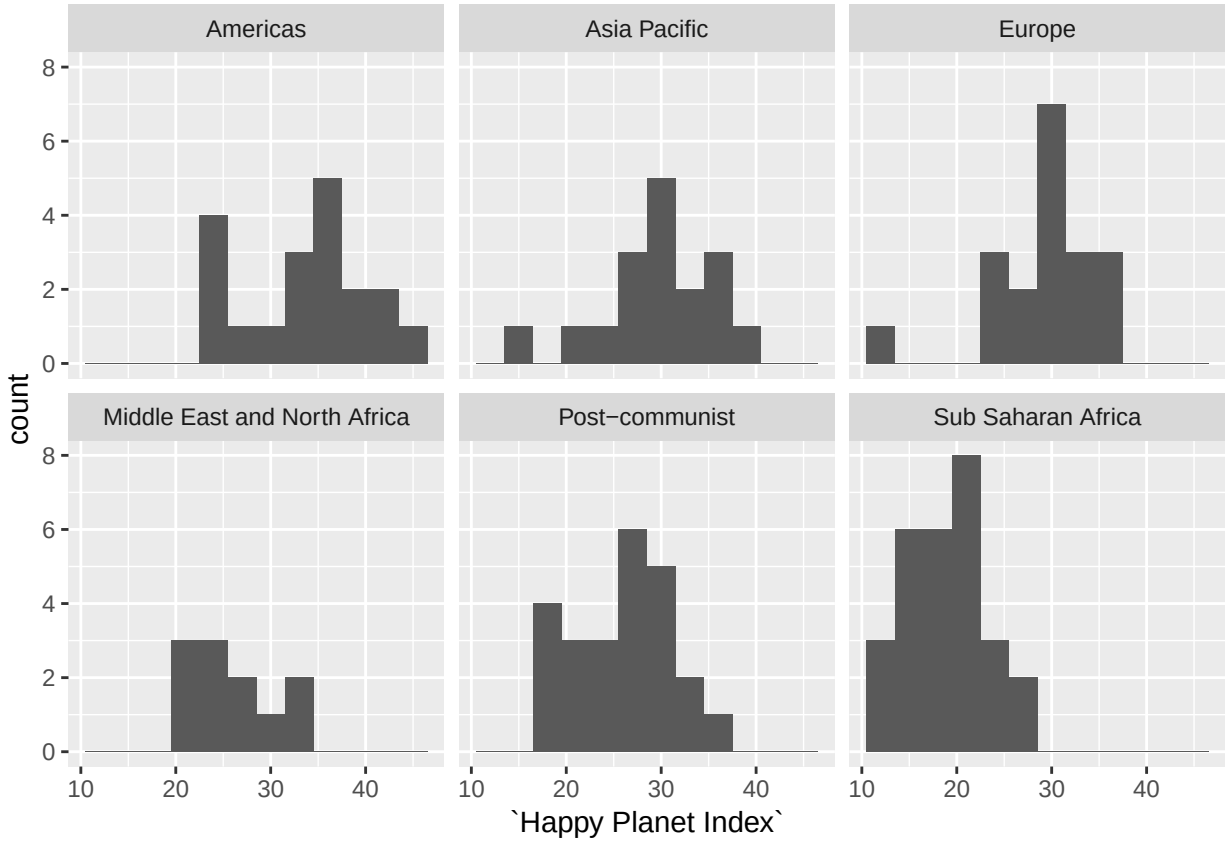
```
# Let's have a quick look at the HPI data
hpi_hist_3 <- ggplot(raw, aes(`Happy Planet Index`)) + geom_histogram(binwidth = 3)
saveRDS(hpi_hist_3, "graphs/hpi_hist_3.rds")
hpi_hist_1 <- ggplot(raw, aes(`Happy Planet Index`)) + geom_histogram(binwidth = 1)
saveRDS(hpi_hist_1, "graphs/hpi_hist_1.rds")

hpi_region_hist <- ggplot(raw, aes(`Happy Planet Index`)) + facet_wrap(~Region) +
                                        geom_histogram(binwidth = 3)
saveRDS(hpi_region_hist, "graphs/hpi_region_hist.rds")

hpi_scatter <- ggplot(raw, aes(x = `Happy Life Years`, y = 1.73 - `Footprint\r\n(gha/capita)`, size = `
  geom_point()
saveRDS(hpi_scatter, "graphs/hpi_scatter.rds")

plot(hpi_hist_3)
```



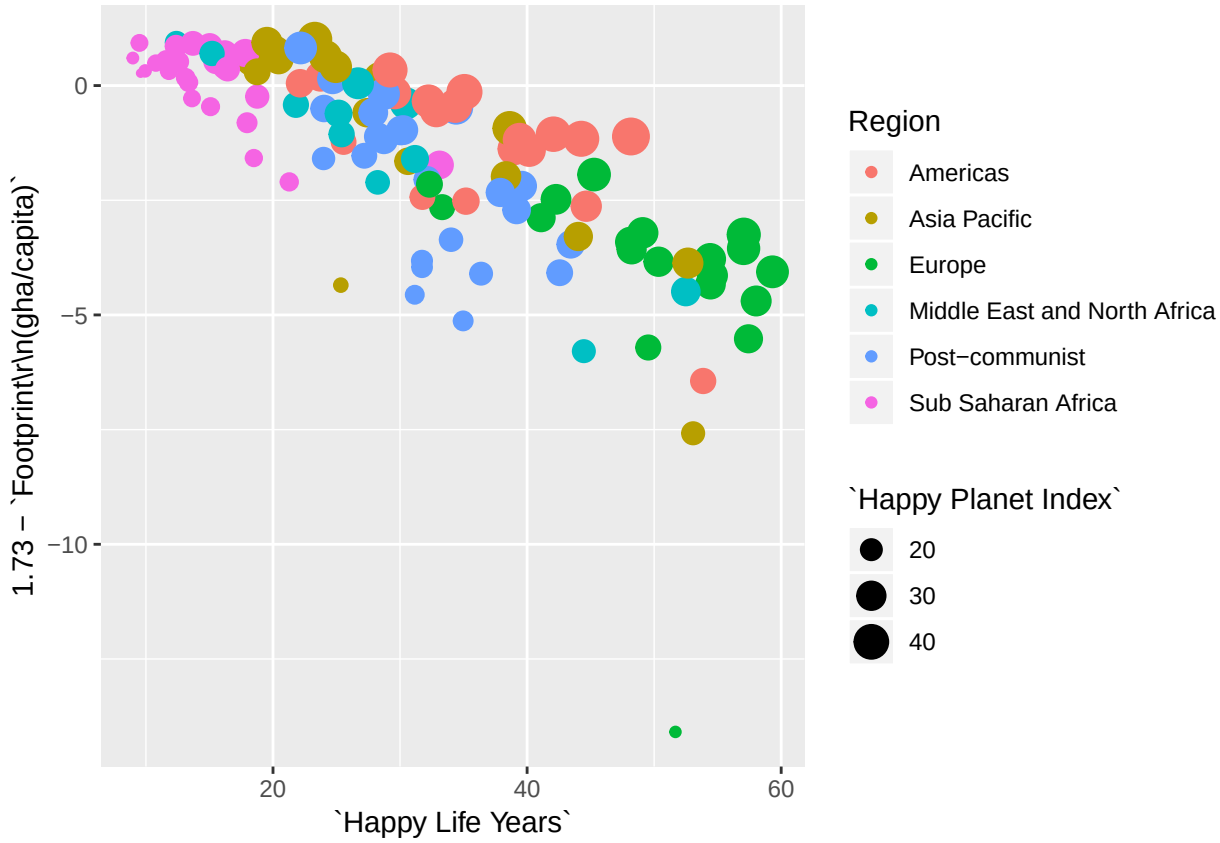

```
plot(hpi_hist_1)
```




```
plot(hpi_region_hist)
```




```
plot(hpi_scatter)
```



Cluster modeling

```
data_clusters <- kmeans(scaled[, -c(1, 2)], 6, nstart = 20)

raw_clustered <- mutate(raw, cluster = as.factor(data_clusters$cluster))
scaled_clustered <- mutate(scaled, cluster = as.factor(data_clusters$cluster))

cluster_expl <- ggplot(raw_clustered, aes(x = `Happy Life Years`, y = 1.73 - `Footprint\r\n(gha/capita)`
saveRDS(cluster_expl, "graphs/cluster_expl.rds")

plot(cluster_expl)
```

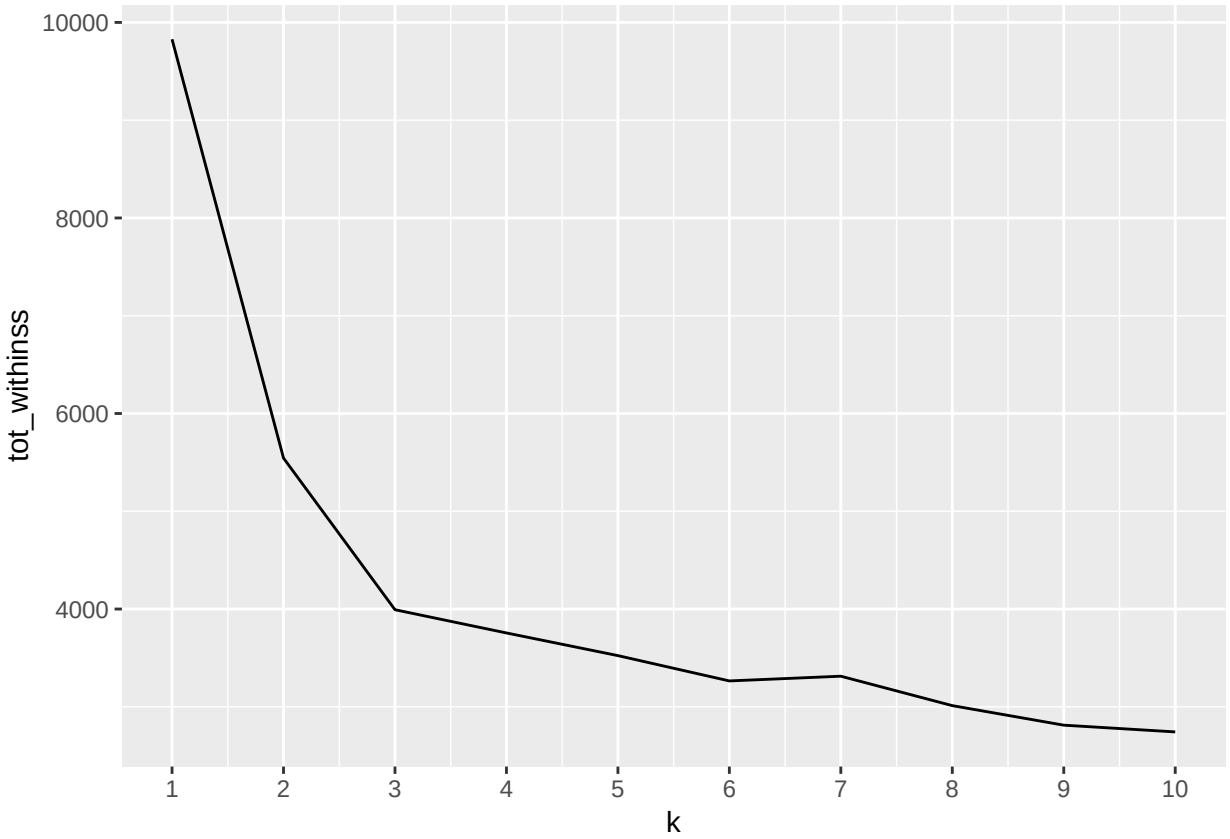

```
# Using the elbow method

tot_withinss <- map_dbl(1:10,  function(k){
  model <- kmeans(x = scaled[, -c(1, 2)], centers = k)
  model$tot.withinss
})

elbow_df <- data.frame(
  k = 1:10,
  tot_withinss = tot_withinss
)

print(elbow_df)
```

```
##     k tot_withinss
## 1   1     9828.000
## 2   2     5543.768
## 3   3     3992.840
## 4   4     3755.259
## 5   5     3523.340
## 6   6     3264.431
## 7   7     3312.782
## 8   8     3012.013
## 9   9     2811.984
## 10 10     2743.001
```

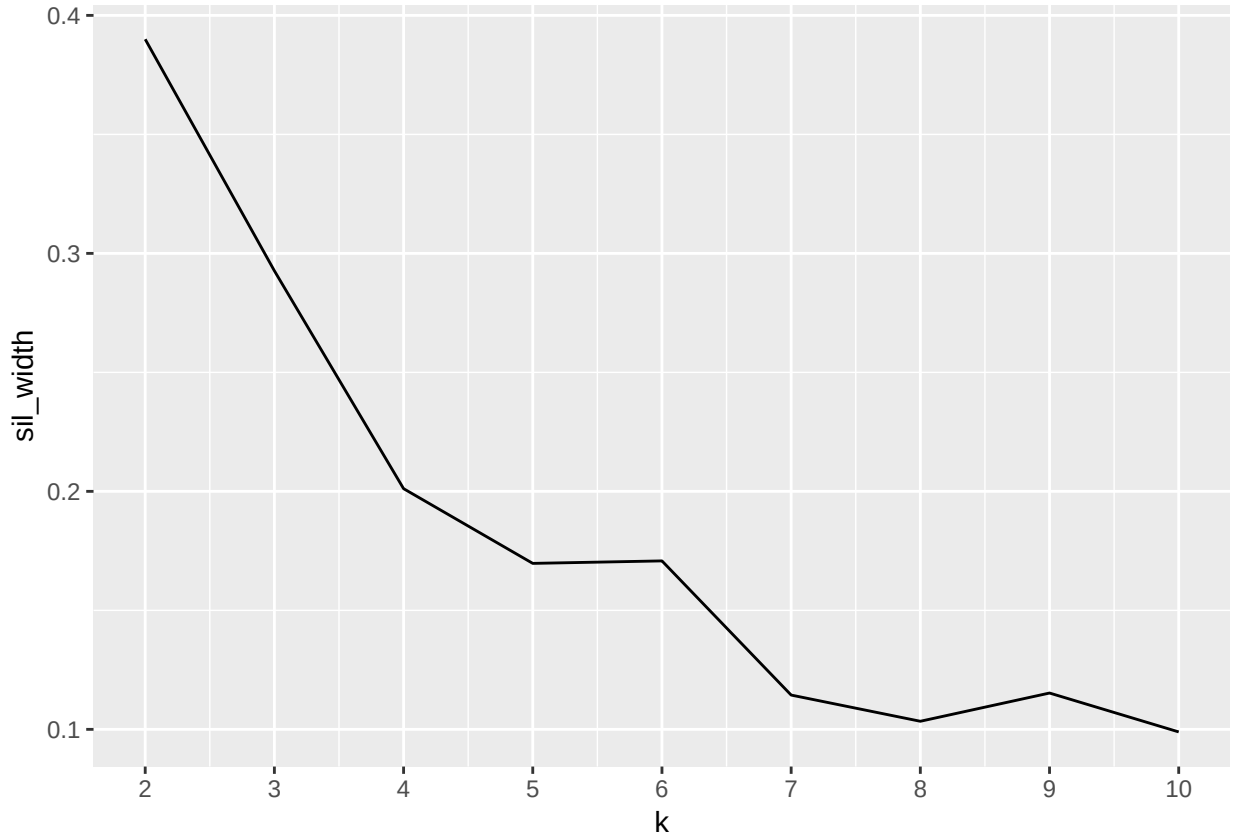
```r
elbow_plot <- ggplot(elbow_df, aes(x = k, y = tot_withinss)) +
                geom_line() +
                scale_x_continuous(breaks = 1:10)
saveRDS(elbow_plot, "graphs/elbow_plot.rds")

# Silhouette width method https://campus.datacamp.com/courses/cluster-analysis-in-r/k-means-clustering?

sil_width <- map_dbl(2:10,  function(k){
  model <- pam(x = scaled[, -c(1, 2)], k = k)
  model$silinfo$avg.width
})

sil_df <- data.frame(
  k = 2:10,
  sil_width = sil_width
)

print(sil_df)
```

```
##    k  sil_width
## 1  2 0.38997707
## 2  3 0.29268417
## 3  4 0.20113895
## 4  5 0.16974180
## 5  6 0.17077939
## 6  7 0.11441110
## 7  8 0.10339666
## 8  9 0.11525771
## 9 10 0.09892183
```

```r
sil_width_plot <- ggplot(sil_df, aes(x = k, y = sil_width)) +
                    geom_line() +
                    scale_x_continuous(breaks = 2:10)
saveRDS(sil_width_plot, "graphs/sil_width_plot.rds")

# Plot
plot(elbow_plot)
```

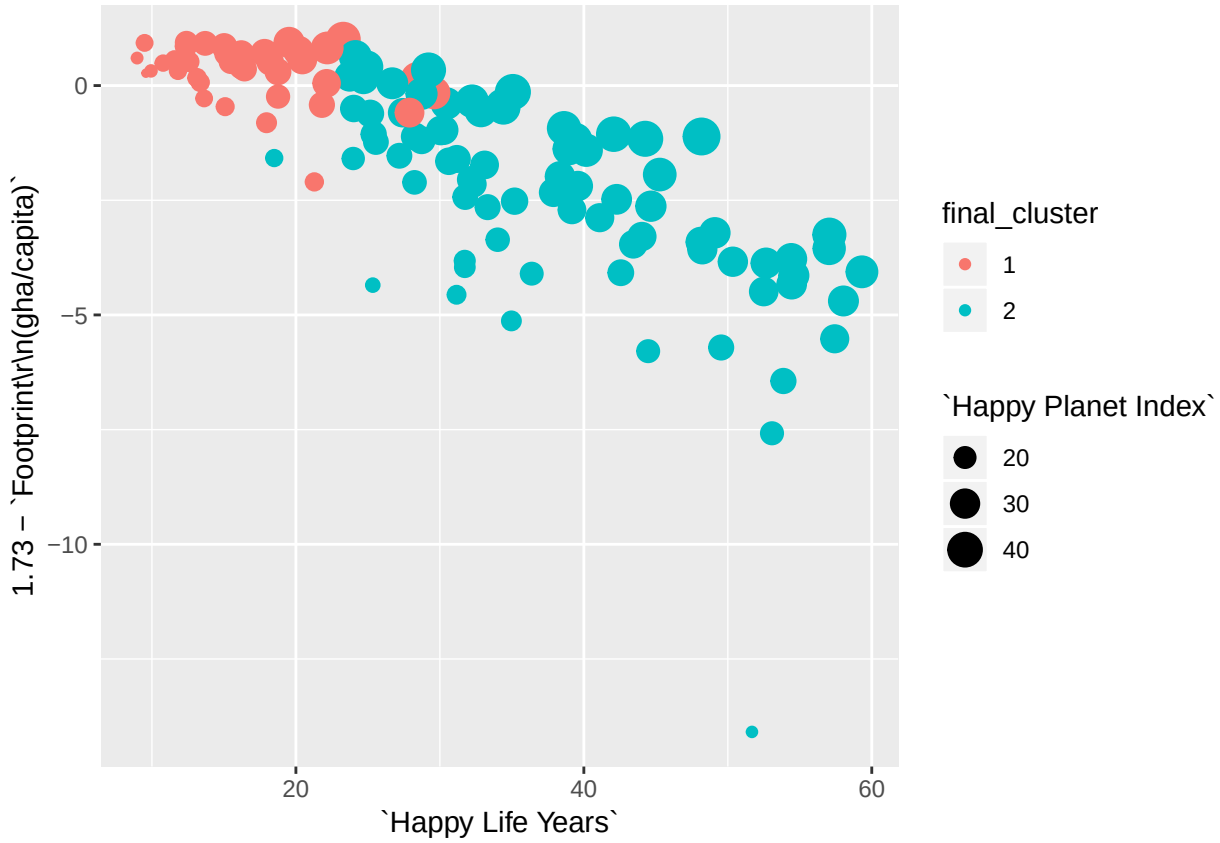



```
plot(sil_width_plot)
```



Visualising clustered data

```
# It seems 2 or 3 clusters is optimal, let's have a look

bi_clusters <- kmeans(scaled[, -c(1, 2)], centers = 2, nstart = 20)

raw_clustered <- raw_clustered %>% mutate(final_cluster = as.factor(bi_clusters$cluster))

bi_plot_hpi <- ggplot(raw_clustered, aes(x = `Happy Life Years`, y = 1.73 - `Footprint\r\n(gha/capita)`
saveRDS(bi_plot_hpi, "bi_plot_hpi.rds")

bi_plot_spi <- ggplot(raw_clustered, aes(x = `Happy Life Years`, y = 1.73 - `Footprint\r\n(gha/capita)`
saveRDS(bi_plot_spi, "bi_plot_spi.rds")

# Arrange plots
plot(bi_plot_hpi)
```

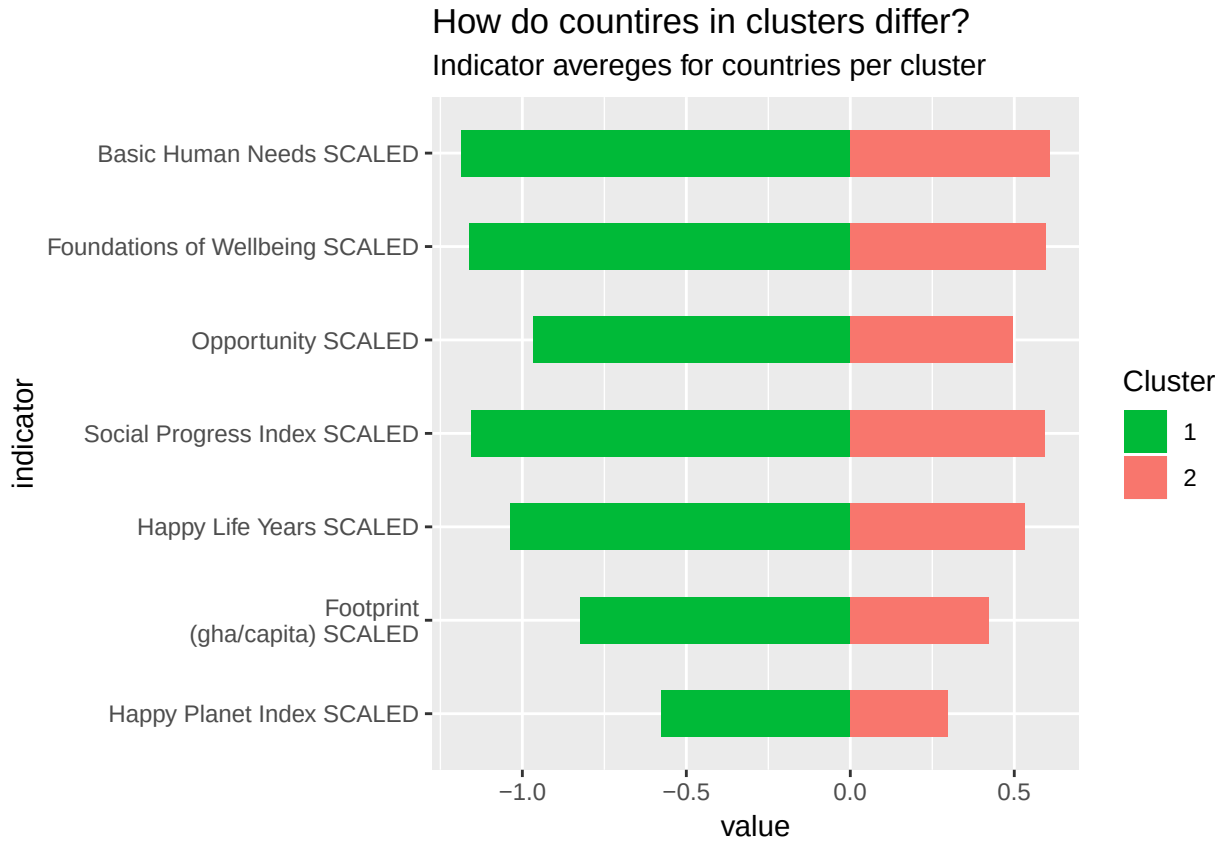



```
plot(bi_plot_spi)
```

Cluster analysis

```
# Lets combina all data for analysis and visualisation
data <- raw_clustered %>% inner_join(scaled_clustered, by = c("Country", "Region"))

# Statistics per bi_cluster to compare
fct_order <- c("Basic Human Needs SCALED",
               "Foundations of Wellbeing SCALED",
               "Opportunity SCALED",
               "Social Progress Index SCALED",
               "Happy Life Years SCALED",
               "Footprint\r\n(gha/capita) SCALED",
               "Happy Planet Index SCALED")

bi_averages <- data %>% select(`Happy Life Years SCALED`,
                               `Footprint\r\n(gha/capita) SCALED`,
                               `Happy Planet Index SCALED`,
                               `Basic Human Needs SCALED`,
                               `Foundations of Wellbeing SCALED`,
                               `Opportunity SCALED`,
                               `Social Progress Index SCALED`,
                                final_cluster) %>%
                        group_by(final_cluster) %>%
```

```
                        summarise_all(mean) %>%
                        gather(-final_cluster, key = "indicator", value = "value") %>%
                        mutate(indicator = as.factor(indicator))

bi_averages$indicator <- fct_rev(fct_relevel(bi_averages$indicator, fct_order))

# Visualising bi averages scaled
diff_bar <- ggplot(bi_averages, aes(x= indicator, y = value, label = value)) +
              geom_bar(stat = "identity", aes(fill = final_cluster), width = 0.5)  +
              scale_fill_manual(name = "Cluster",
                                values = c("1" = "#00ba38", "2" = "#f8766d")) +
              labs(title = "How do countires in clusters differ?",
                   subtitle = "Indicator avereges for countries per cluster") +
              coord_flip()
saveRDS(diff_bar, "graphs/diff_bar.rds")
plot(diff_bar)
```



Appendix

```
# Which countries are in these cluster?
cluster_1 <- data %>% filter(final_cluster == 1) %>% select(Country)
cluster_2 <- data %>% filter(final_cluster == 2) %>% select(Country)

k1 = cluster_1[1:(nrow(cluster_1)/2),] %>% rename("Cluster 1" = "Country")
k2 = cluster_1[(nrow(cluster_1)/2+1):nrow(cluster_1),] %>% rename("countries" = "Country")
k3 = cluster_2[1:(nrow(cluster_2)/2),] %>% rename("Cluster 2" = "Country")
```

```r
k4 = cluster_2[(nrow(cluster_2)/2+1):nrow(cluster_2),]%>% rename("countries" = "Country")
knitr::kable(list(k1, k2, k3, k4))
```

```r
# How long did the whole script take?
script_end <- Sys.time()

print(paste("Total script running time: ", round(difftime(script_end, script_start, units = "mins"), 1)
```

```
## [1] "Total script running time:  0  minutes"
```

Cluster 1	countries
Afghanistan	Lesotho
Bangladesh	Liberia
Benin	Malawi
Botswana	Mauritania
Burkina Faso	Mozambique
Burundi	Myanmar
Cambodia	Nepal
Cameroon	Niger
Chad	Nigeria
Comoros	Pakistan
Djibouti	Rwanda
Egypt	Senegal
Ethiopia	Sierra Leone
Ghana	Swaziland
Guatemala	Tajikistan
Guinea	Tanzania
Honduras	Togo
India	Uzbekistan
Indonesia	Yemen
Kenya	Zimbabwe

Cluster 2	countries
Albania	Latvia
Algeria	Lebanon
Argentina	Lithuania
Armenia	Luxembourg
Australia	Macedonia
Austria	Malaysia
Belarus	Mauritius
Belgium	Mexico
Bhutan	Mongolia
Bolivia	Montenegro
Brazil	Morocco
Bulgaria	Netherlands
Canada	New Zealand
Chile	Nicaragua
China	Norway
Colombia	Oman
Costa Rica	Panama
Croatia	Paraguay
Cyprus	Peru
Czech Republic	Philippines
Denmark	Poland
Dominican Republic	Portugal
Ecuador	Romania
El Salvador	Russia
Estonia	Serbia
Finland	Slovakia
France	Slovenia
Georgia	South Africa
Germany	Spain
Greece	Sri Lanka
Hungary	Suriname
Iceland	Sweden
Iran	Switzerland
Ireland	Thailand
Israel	Tunisia
Italy	Turkey
Japan	Ukraine
Kazakhstan	United Kingdom
Kyrgyzstan	Uruguay