# What makes people happy?
# Probe into the World Happiness Report 2024

Yuxin Gong

Student Number: 21088401

Course Number: STAT 847

Word Count: 1611 words

2024-04-22

On March 20th, fresh insights from the World Happiness Report 2024 were released, painting the picture of happiness trends across different ages and generation. The 2024 report ranked Canada as the 15th happiest country in the world, 2 places lower than the ranking in 2023 report. To probe into the factors that have an impact on the happiness level of countries, this project will use the data from the World Happiness Report 2024 as well as the data from World Bank to conduct relevant analyses and complete the Q1-Q6 in the requirements.

# 1. Data Description

The data used in this report are from the World Happiness Report 2024 and World Bank. Detailed data documentation can be found in Appendix A1. The World Happiness Report 2024 data incude the following 11 variables: country name, year, Life Ladder, Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity , Perceptions of corruption, Positive affect, and Negative affect. The World Bank data include the following 8 variables: country name, year, political factor, social protection, social inclusion, technology factor, legal factor, and environment factor. We also have the map data: shapefile from World Bank.

After cleaning and combing all the variables, we can get a dataset containing 17 variables, with life ladder as the response variable and the rest as predictors. Life ladder is the happiness level in the World Happiness Report. We can get an actual ladder variable (a categorical variable) by flooring the life ladder and make it a factor. This way, we have a dataset of 18 variables which contains two index variables: country name and year, one response variable in categorical version, one response variable in continuous version, and the rest 14 are explanatory variables.

# 2. Project Motivation, Goals, and Tasks (Q1)

With a dataset described above, and with the aim to further probe into the factors that impact country happiness levels, I put forward the following questions/tasks and approaches I have to answer them.

(1) Geographical visualization of the life happiness score. (map graph)

(2) What are some of the most important variables to explain happiness level? (Variable Importance – subset regression)

(3) Geographical visualization of the other 5 most important explanatory variables (map graph)

(4) Correlation of the 6 variables (Q2 and Q6, ggpairs and ggplot)

(5) Predict the happiness level (categorical version) using classification tree (Q2, classfication tree)

(6) Explain the happiness level using PCA (Q3, PCA)

(7) How does the world average happiness level develop over time? Are we in general happier than the old times after in 21st century? (time trend plot)

(8) How does the happiness level change before and after COVID? (t-test)

(9) Are people in developed countries happier than those in developing countries? (t-test)

The following analysis will address the questions (1), (2), (4), (5), (6).

## 3. Variables and Visualization (Q2, Q6)

First, I want to take a glance at the geographical distribution of the happiness level around the world. After merging the variable data into the map data, I can plot the map of world happiness in 2023. See Figure 1. We can see that in general, the happiness level in Noeth America, Europe, and Australia is generally higher than that in South America, Asia, and Africa.

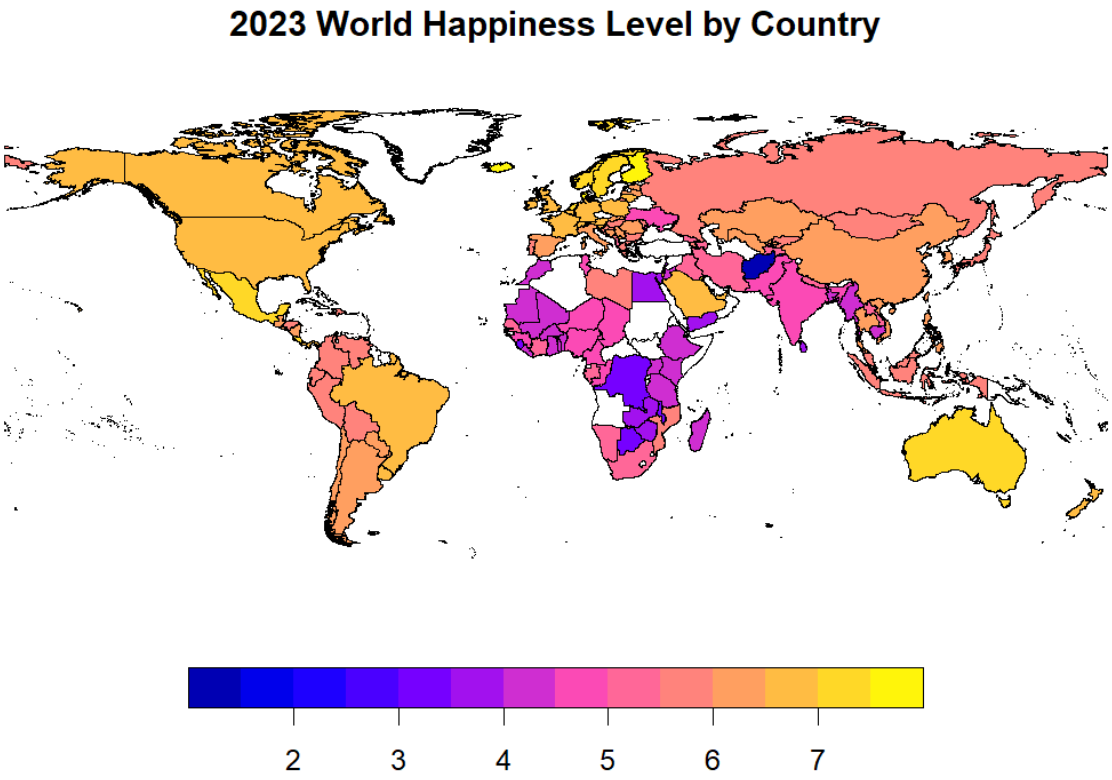



Figure 1: 2023 World Happiness Level by Country

To probe into the factors that impact the happiness levels of countries, I conducted a best subset selection on the 12 predictors (excluding the Positive Effect and Negative Effect, because they are, respectively, the average of previous-day affect measures for laughter, enjoyment, and doing interesting things, and the average of previous-day affect measures for worry, sadness, and anger. These are time series components and are not the study objects of this project). With life ladder being the response variable, the most important 5 explanatory variables from best subset selection are: log_gdp_per_capita, social_support, freedom_to_make_life_choices, political_stability, and environment, which together produce an adjusted R-squared of 0.3778. According to The World Happiness Report 2024, the R-squared reached by the variables used in the report excluding the effect variables is around 0.34, similar to the highest R-squared produced from best subset model selection. We can conclude that the report has also done its best to explain the world happiness level.

Hence, according to best subset selection, my most important 6 variables are: life_ladder, log_gdp_per_capita, social_support, freedom_to_make_life_choices, political_stability, and environment, in which life_ladder can be in either categorical version or continuous version. I will use the categorical version life_ladder in classsifcation tree analysis, and use the continuous version in the PCA analysis.

With the categorical version of life_ladder, in my most important 6 variables, life ladder is categorical, and the rest of the variables are continuous. A ggpairs plot is conducted to check the correlation and distribution of the variables. See Figure 2. As shown on the graph, between the five continuous variables, all the

correlation coefficients are positive and most are significantly positive. The categorical variable distribution is right skewed, so are most of its distributions in different category.

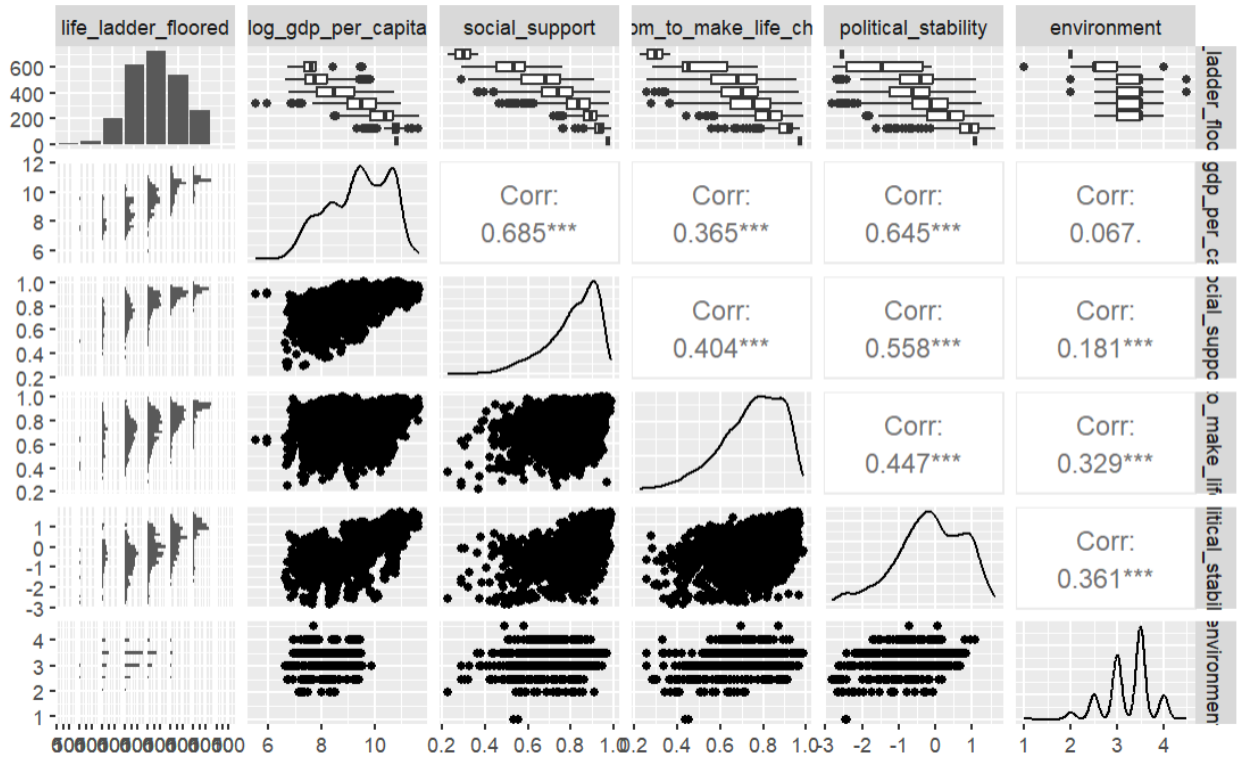



Figure 2: A Ggpairs Plot of the 6 Most Important Variables

To take a closer look at the correlation between the variables, I also built a corrplot and a ggplot to show the relationship. See Figure 3 and Figure 4.

As shown in both graphs, all correlations are positive.

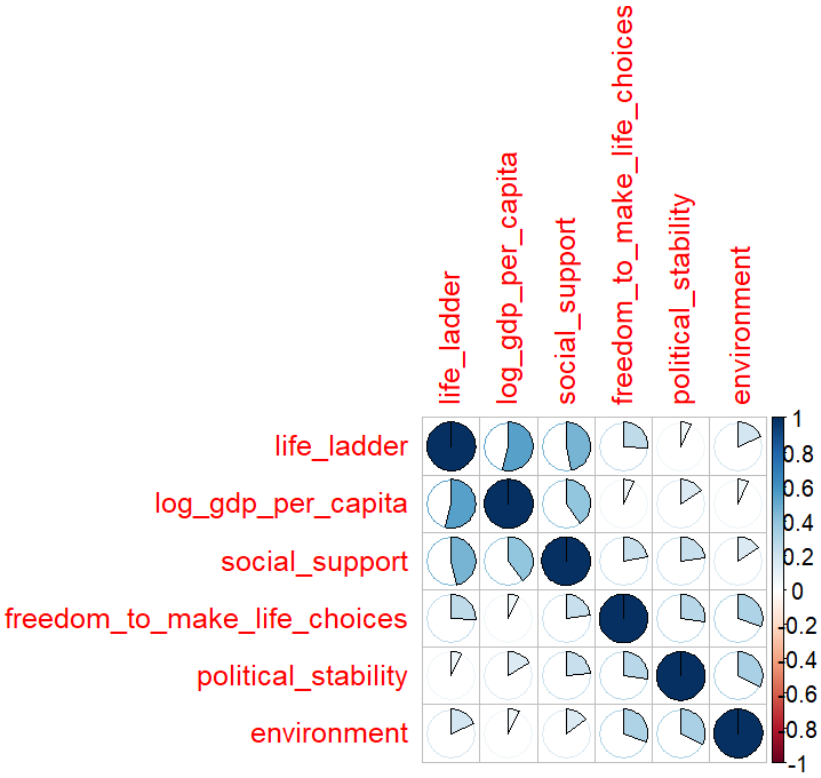



Figure 3: Corrplot Using the Method "Pie"

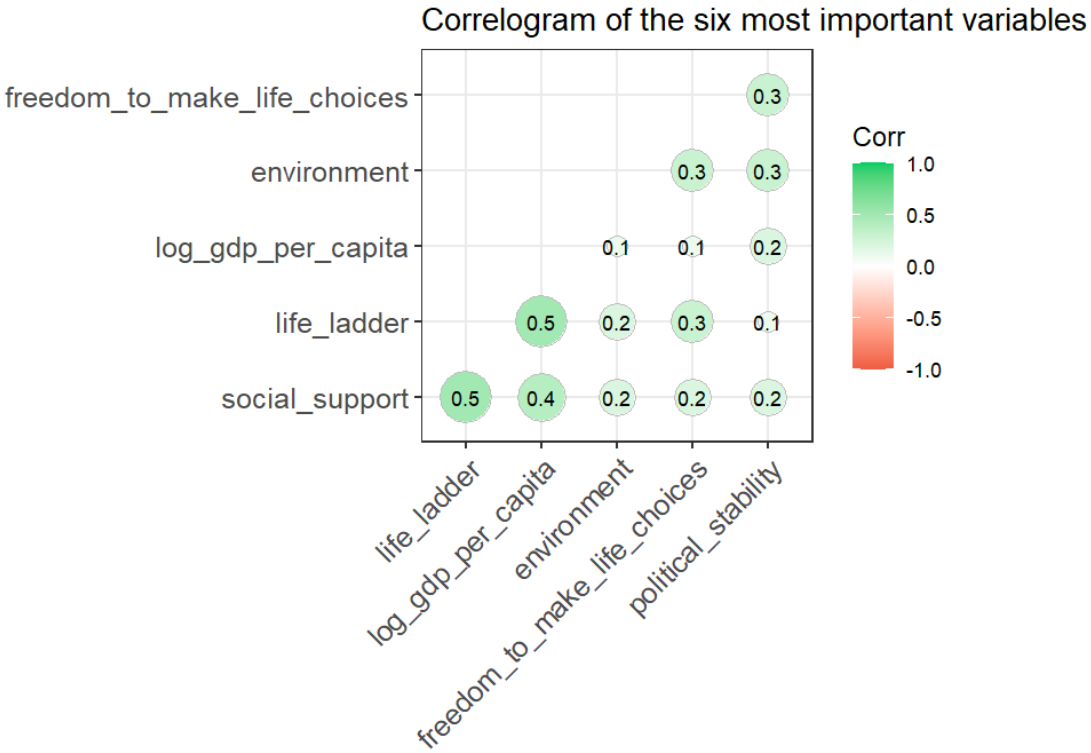



Figure 4: Ggplot: Correlogram of the Six Most Important Variables

# 4. Classification Tree (Q3)

Based on the variable selection above, I will do a classification tree of life_ladder (categorical version) as a function of the other five variables. I used 80% as training data and 20% as testing data. The classification tree plot is shown below. Only 4 out of 5 variables are used in the tree and only four predictors out of five are used and only 5 levels: level 3-7 are used. Therefore, we will only have predictions of level 3-7 based on this tree model.

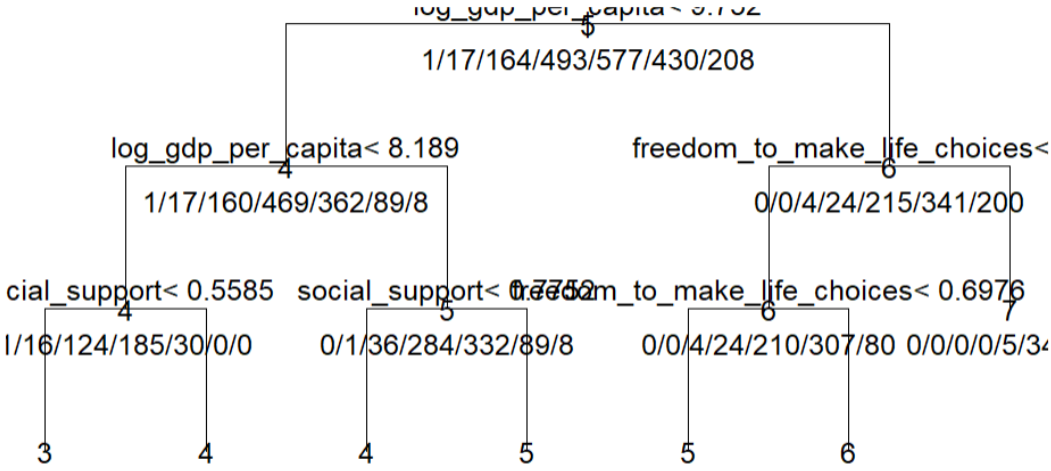



Figure 5: Classification Tree Plot (Ugly Version)

The confusion matrix results of this model is shown in Figure 7. The accuracy rate is 57.51%. Two prediction example down the tree are: The first prediction is 3, and the 10th prediction is 6.

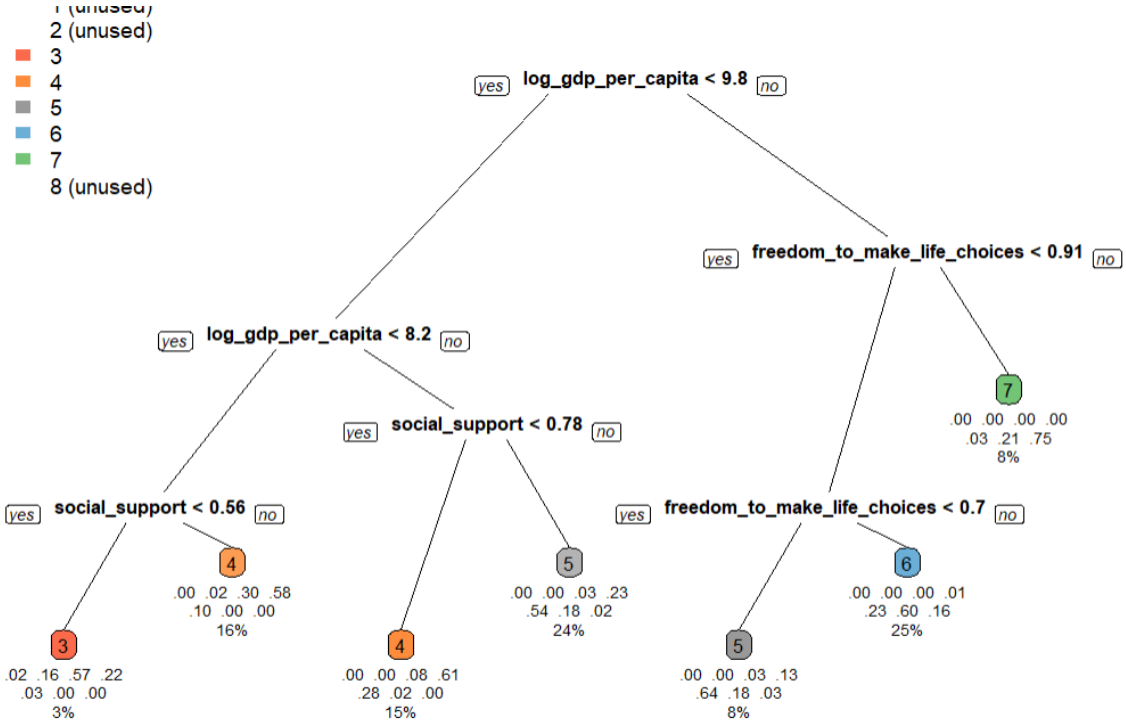



Figure 6: Classification Tree Plot (Beautified Version)

## 5. Continuous Model: PCA (Q4)

When we were selecting the most important variables, we did not make use of all the variables with best subset, and we can actually make use of all of them by using PCA. Here I use the continuous version of life_ladder as response variable, and the rest as explanatory variables, and fit a PCA model to explain the variable life_ladder. The PCA results are shown in Figure 8 and the PCA plot is in Figure 9. From Figure 8, we can see that the total (cumulative) variance explained by the first 10 principal components is as high as 0.92978 so it is enough that we use the first 10 PC. Using the first 10 PC, I then conducted a linear regression on the response variable life_ladder, getting an adjusted R-squared of 0.3675. Compared to the best subsets conducted above, which has an Adjusted R-squared of 0.3778, the PCA-generated variables perform slightly worse since they only have an Adjusted R-squared of 0.3675.

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5  6  7  8
         1  0  0  0  0  0  0  0  0
         2  0  0  0  0  0  0  0  0
         3  1  1  4  4  1  0  0  0
         4  0  3 29 88 25  1  0  0
         5  0  0  1 33 83 33  3  0
         6  0  0  0  1 36 65 18  0
         7  0  0  0  0  2  8 32  1
         8  0  0  0  0  0  0  0  0

Overall Statistics

               Accuracy : 0.5751
                 95% CI : (0.5291, 0.6201)
    No Information Rate : 0.3108
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4319

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
Sensitivity          0.000000 0.000000 0.117647   0.6984   0.5646   0.6075  0.60377 0.000000
Specificity          1.000000 1.000000 0.984055   0.8329   0.7853   0.8497  0.97381 1.000000
Pos Pred Value            NaN      NaN 0.363636   0.6027   0.5425   0.5417  0.74419      NaN
Neg Pred Value       0.997886 0.991543 0.935065   0.8838   0.8000   0.8810  0.95116 0.997886
Prevalence           0.002114 0.008457 0.071882   0.2664   0.3108   0.2262  0.11205 0.002114
Detection Rate       0.000000 0.000000 0.008457   0.1860   0.1755   0.1374  0.06765 0.000000
Detection Prevalence 0.000000 0.000000 0.023256   0.3087   0.3235   0.2537  0.09091 0.000000
Balanced Accuracy    0.500000 0.500000 0.550851   0.7656   0.6750   0.7286  0.78879 0.500000
```

Figure 7: Confusion Matrix of Classification Tree

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12    PC13    PC14
Standard deviation     2.0153 1.3694 1.2151 1.1112 1.03613 1.02418 0.85460 0.79840 0.68423 0.64052 0.59602 0.54273 0.50798 0.27430
Proportion of Variance 0.2901 0.1340 0.1055 0.0882 0.07668 0.07492 0.05217 0.04553 0.03344 0.02931 0.02537 0.02104 0.01843 0.00537
Cumulative Proportion  0.2901 0.4241 0.5295 0.6177 0.69441 0.76934 0.82150 0.86703 0.90048 0.92978 0.95515 0.97619 0.99463 1.00000
```

Figure 8: PCA Results

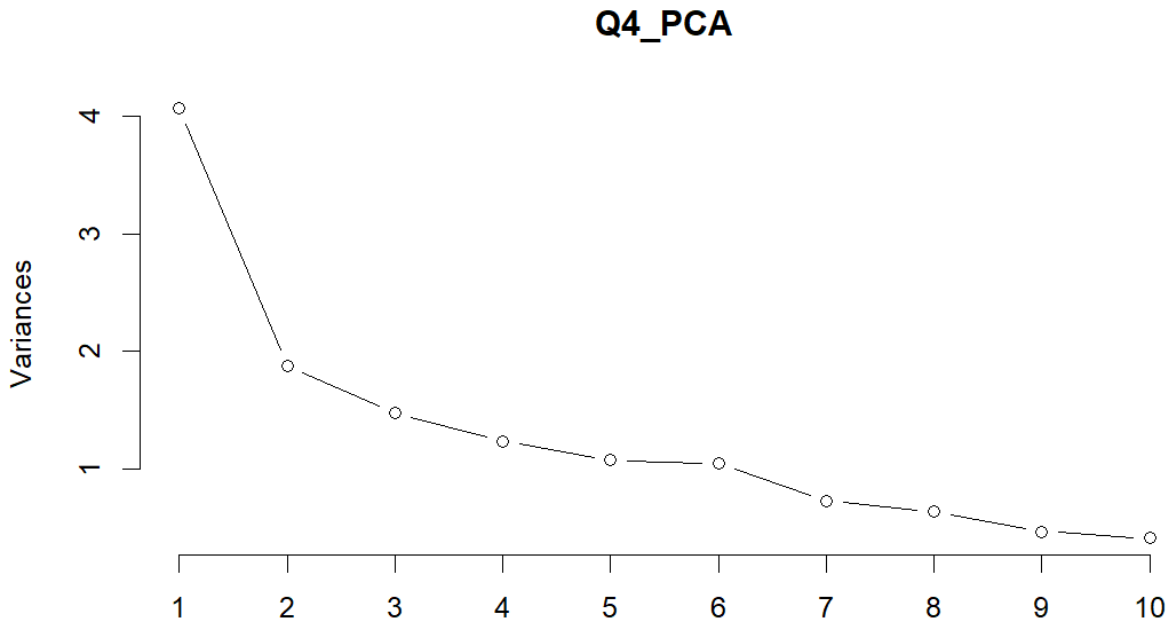



Figure 9: PCA Plot

# 6. Conclusions and Implications

Therefore, we can conclude that the most important 5 variables that have an impact on the country's happiness level are: log_gdp_per_capita, social_support, freedom_to_make_life_choices, political_stability, and environment. In order to boost the happiness level of Canada, relevant departments can make improvements on these five aspects.

# Appendix (Q5)

## A1 Data Documentation

### A1.1 Data Collection

The instructions below show the exact steps to get the original data.

(1) Life ladder and 6 explanatory variables used in the life happiness report (2005-2023) can be obtained by:
https://worldhappiness.report/ed/2024/#appendices-and-data -> Appendices & Data -> download "Data for Table 2.1"
This table contains the following variables: Country name, year, Life Ladder, Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity , Perceptions of corruption, Positive affect, and Negative affect. The detailed description and meaning of each of these variables can be found in https://worldhappiness.report/ed/2024/happiness-of-the-younger-the-

older-and-those-in-between/ -> "Technical Box 2: Detailed information about each of the predictors in Table 2.1"

Note that this table downloaded is an xls file.

(2) Other explanatory variables (PESTEL external environment variables):

Political Stability and Absence of Violence/Terrorism: Estimate, CPIA social protection rating (1=low to 6=high), CPIA policies for social inclusion/equity cluster average (1=low to 6=high), Scientific and technical journal articles, Rule of Law: Estimate, CPIA policy and institutions for environmental sustainability rating (1=low to 6=high).

Go to Website: https://data.worldbank.org/ DataBank -> World Development Indicators -> Databank -> Country "Select All", Series: the variables above, Time "Recent 25 years" -> Download options: CSV, Export range: Entire Dataset, Data format: List, Variable format: Both codes & name, NA preference: Blank, Text field delimiter: ", Metedata:No.

In this way, we will get a long table which contains: Country Name, Country Code, Series Name, Series Code, Time, Time Code, Value. We will need to convert it into a wide format later.

(3) Map file: shapefile
https://datacatalog.worldbank.org/search/dataset/0038272/World-Bank-Official-Boundaries (Sometimes the link does not work. If the link does not work, Google search "world bank shapefile" and click on the top link) download zip "World Country Polygons -Very High Definition" -> unzip the zip file.

We need to read the shapefile from the shapefile folder to get everything we need.

### A1.2 Data Cleaning and manipulation

The code below shows the full process of the data cleaning and manipulation process. All of the codes are reproducible.

# Yuxin Gong - Final Project What makes people happy: Probe into the World Happiness Report 2024

Yuxin Gong

2024-04-15

## Q5: Describe and show the code used to collect and clean the data.

### Collecting Data

See Data Documentation - Data Collection

### Cleaning data

```
######## Load all the libraries
library(sf)
```

```
## Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(readxl)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
report = read_excel("DataForTable2.1.xls")
worldbank_long = read.csv("worldbank_long2.csv")
geo = st_read("D:/Angie/Grad in UWaterloo/Term 2 W24/STAT847/Final/Data/shapefile/WB_countries_Admin0_10
```

```
## Reading layer `WB_countries_Admin0_10m' from data source
##   `D:\Angie\Grad in UWaterloo\Term 2 W24\STAT847\Final\Data\shapefile\WB_countries_Admin0_10m.shp'
##   using driver `ESRI Shapefile'
```

```
## Simple feature collection with 251 features and 52 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -59.47275 xmax: 180 ymax: 83.6341
## Geodetic CRS:  WGS 84
```

```
head(report, n = 20)
```

```
## # A tibble: 20 x 11
##    `Country name`  year `Life Ladder` `Log GDP per capita` `Social support`
##    <chr>          <dbl>         <dbl>                <dbl>            <dbl>
##  1 Afghanistan     2008          3.72                 7.35            0.451
##  2 Afghanistan     2009          4.40                 7.51            0.552
##  3 Afghanistan     2010          4.76                 7.61            0.539
##  4 Afghanistan     2011          3.83                 7.58            0.521
##  5 Afghanistan     2012          3.78                 7.66            0.521
##  6 Afghanistan     2013          3.57                 7.68            0.484
##  7 Afghanistan     2014          3.13                 7.67            0.526
##  8 Afghanistan     2015          3.98                 7.65            0.529
##  9 Afghanistan     2016          4.22                 7.65            0.559
## 10 Afghanistan     2017          2.66                 7.65            0.491
## 11 Afghanistan     2018          2.69                 7.63            0.508
## 12 Afghanistan     2019          2.38                 7.64            0.420
## 13 Afghanistan     2021          2.44                 7.32            0.454
## 14 Afghanistan     2022          1.28                NA               0.228
## 15 Afghanistan     2023          1.45                NA               0.368
## 16 Albania         2007          4.63                 9.12            0.821
## 17 Albania         2009          5.49                 9.24            0.833
## 18 Albania         2010          5.27                 9.28            0.733
## 19 Albania         2011          5.87                 9.31            0.759
## 20 Albania         2012          5.51                 9.33            0.785
## # i 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>
```

```
head(worldbank_long, n = 20)
```

```
##    Country.Name Country.Code
## 1   Afghanistan          AFG
## 2   Afghanistan          AFG
## 3   Afghanistan          AFG
## 4   Afghanistan          AFG
## 5   Afghanistan          AFG
## 6   Afghanistan          AFG
## 7   Afghanistan          AFG
## 8   Afghanistan          AFG
## 9   Afghanistan          AFG
## 10  Afghanistan          AFG
## 11  Afghanistan          AFG
## 12  Afghanistan          AFG
## 13  Afghanistan          AFG
## 14  Afghanistan          AFG
```

```
## 15  Afghanistan          AFG
## 16  Afghanistan          AFG
## 17  Afghanistan          AFG
## 18  Afghanistan          AFG
## 19  Afghanistan          AFG
## 20  Afghanistan          AFG
##                                                          Series.Name
## 1  Political Stability and Absence of Violence/Terrorism: Estimate
## 2  Political Stability and Absence of Violence/Terrorism: Estimate
## 3  Political Stability and Absence of Violence/Terrorism: Estimate
## 4  Political Stability and Absence of Violence/Terrorism: Estimate
## 5  Political Stability and Absence of Violence/Terrorism: Estimate
## 6  Political Stability and Absence of Violence/Terrorism: Estimate
## 7  Political Stability and Absence of Violence/Terrorism: Estimate
## 8  Political Stability and Absence of Violence/Terrorism: Estimate
## 9  Political Stability and Absence of Violence/Terrorism: Estimate
## 10 Political Stability and Absence of Violence/Terrorism: Estimate
## 11 Political Stability and Absence of Violence/Terrorism: Estimate
## 12 Political Stability and Absence of Violence/Terrorism: Estimate
## 13 Political Stability and Absence of Violence/Terrorism: Estimate
## 14 Political Stability and Absence of Violence/Terrorism: Estimate
## 15 Political Stability and Absence of Violence/Terrorism: Estimate
## 16 Political Stability and Absence of Violence/Terrorism: Estimate
## 17 Political Stability and Absence of Violence/Terrorism: Estimate
## 18 Political Stability and Absence of Violence/Terrorism: Estimate
## 19 Political Stability and Absence of Violence/Terrorism: Estimate
## 20                 CPIA social protection rating (1=low to 6=high)
##       Series.Code Time Time.Code     Value
## 1          PV.EST 2005    YR2005 -2.067510
## 2          PV.EST 2006    YR2006 -2.219135
## 3          PV.EST 2007    YR2007 -2.413373
## 4          PV.EST 2008    YR2008 -2.691361
## 5          PV.EST 2009    YR2009 -2.711421
## 6          PV.EST 2010    YR2010 -2.579152
## 7          PV.EST 2011    YR2011 -2.502060
## 8          PV.EST 2012    YR2012 -2.418561
## 9          PV.EST 2013    YR2013 -2.519349
## 10         PV.EST 2014    YR2014 -2.411068
## 11         PV.EST 2015    YR2015 -2.562625
## 12         PV.EST 2016    YR2016 -2.662156
## 13         PV.EST 2017    YR2017 -2.794974
## 14         PV.EST 2018    YR2018 -2.753262
## 15         PV.EST 2019    YR2019 -2.652407
## 16         PV.EST 2020    YR2020 -2.702632
## 17         PV.EST 2021    YR2021 -2.518530
## 18         PV.EST 2022    YR2022 -2.550802
## 19         PV.EST 2023    YR2023        NA
## 20 IQ.CPA.PROT.XQ 2005    YR2005        NA
```

```
head(geo, n = 10)
```

```
## Simple feature collection with 10 features and 52 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
```

```
## Bounding box:  xmin: -109.4537 ymin: -55.9185 xmax: 140.9776 ymax: 53.56944
## Geodetic CRS:  WGS 84
##    OBJECTID      featurecla LEVEL              TYPE
## 1         1 Admin-0 country     2 Sovereign country
## 2         2 Admin-0 country     2 Sovereign country
## 3         3 Admin-0 country     2 Sovereign country
## 4         4 Admin-0 country     2 Sovereign country
## 5         5 Admin-0 country     2 Sovereign country
## 6         6 Admin-0 country     2 Sovereign country
## 7         7 Admin-0 country     2 Sovereign country
## 8         8 Admin-0 country     2 Sovereign country
## 9         9 Admin-0 country     2           Country
## 10       10 Admin-0 country     2           Country
##                         FORMAL_EN FORMAL_FR    POP_EST POP_RANK GDP_MD_EST
## 1           Republic of Indonesia      <NA>  260580739       17    3028000
## 2                        Malaysia      <NA>   31381992       15     863000
## 3               Republic of Chile      <NA>   17789267       14     436100
## 4  Plurinational State of Bolivia      <NA>   11138234       14      78350
## 5                Republic of Peru      <NA>   31036656       15     410400
## 6              Argentine Republic      <NA>   44293293       15     879400
## 7              Republic of Cyprus      <NA>    1221549       12      29260
## 8               Republic of India      <NA> 1281935911       18    8721000
## 9      People's Republic of China      <NA> 1379302771       18   21140000
## 10               State of Israel      <NA>    8299706       13     297000
##    POP_YEAR LASTCENSUS GDP_YEAR                    ECONOMY
## 1      2017       2010     2016   4. Emerging region: MIKT
## 2      2017       2010     2016       6. Developing region
## 3      2017       2002     2016    5. Emerging region: G20
## 4      2017       2001     2016    5. Emerging region: G20
## 5      2017       2007     2016    5. Emerging region: G20
## 6      2017       2010     2016    5. Emerging region: G20
## 7      2017       2001     2016       6. Developing region
## 8      2017       2011     2016   3. Emerging region: BRIC
## 9      2017       2010     2016   3. Emerging region: BRIC
## 10     2017       2009     2016 2. Developed region: nonG7
##                 INCOME_GRP FIPS_10_ ISO_A2 ISO_A3 ISO_A3_EH ISO_N3 UN_A3 WB_A2
## 1   4. Lower middle income       ID     ID    IDN       IDN    360   360    ID
## 2   3. Upper middle income       MY     MY    MYS       MYS    458   458    MY
## 3   3. Upper middle income       CI     CL    CHL       CHL    152   152    CL
## 4   4. Lower middle income       BL     BO    BOL       BOL    068   068    BO
## 5   3. Upper middle income       PE     PE    PER       PER    604   604    PE
## 6   3. Upper middle income       AR     AR    ARG       ARG    032   032    AR
## 7  2. High income: nonOECD       CY     CY    CYP       CYP    196   196    CY
## 8   4. Lower middle income       IN     IN    IND       IND    356   356    IN
## 9   3. Upper middle income       CH     CN    CHN       CHN    156   156    CN
## 10    1. High income: OECD      -99     IL    ISR       ISR    376   376    IL
##    WB_A3     CONTINENT REGION_UN          SUBREGION                 REGION_WB
## 1    IDN          Asia      Asia South-Eastern Asia        East Asia & Pacific
## 2    MYS          Asia      Asia South-Eastern Asia        East Asia & Pacific
## 3    CHL South America  Americas      South America  Latin America & Caribbean
## 4    BOL South America  Americas      South America  Latin America & Caribbean
## 5    PER South America  Americas      South America  Latin America & Caribbean
## 6    ARG South America  Americas      South America  Latin America & Caribbean
## 7    CYP          Asia      Asia       Western Asia      Europe & Central Asia
```

```
## 8    IND          Asia      Asia      Southern Asia                South Asia
## 9    CHN          Asia      Asia       Eastern Asia        East Asia & Pacific
## 10   ISR          Asia      Asia       Western Asia Middle East & North Africa
##                  NAME_AR         NAME_BN             NAME_DE
## 1                                          Indonesien
## 2                                             Malaysia
## 3                                                   Chile
## 4                                              Bolivien
## 5                                                     Peru
## 6                                           Argentinien
## 7                                          Republik Zypern
## 8                                                 Indien
## 9                          Volksrepublik China
## 10                                                Israel
##                        NAME_EN                 NAME_ES
## 1                     Indonesia               Indonesia
## 2                      Malaysia                 Malasia
## 3                         Chile                   Chile
## 4                       Bolivia                 Bolivia
## 5                          Peru                    Perú
## 6                     Argentina               Argentina
## 7                        Cyprus                  Chipre
## 8                         India                   India
## 9  People's Republic of China República Popular China
## 10                        Israel                  Israel
##                          NAME_FR                  NAME_EL            NAME_HI
## 1                        Indonésie                 I
## 2                         Malaisie                  M
## 3                            Chili                      X
## 4                          Bolivie                   B
## 5                             Pérou                    Π
## 6                        Argentine                 A
## 7                           Chypre                    K
## 8                             Inde                     I
## 9  République populaire de Chine Λ   Δ        K
## 10                        Israël                    I
##      NAME_HU                 NAME_ID   NAME_IT        NAME_JA        NAME_KO
## 1  Indonézia               Indonesia Indonesia
## 2   Malajzia                Malaysia   Malesia
## 3      Chile                   Chili      Cile
## 4    Bolívia                 Bolivia   Bolivia
## 5       Peru                    Peru      Perù
## 6  Argentína               Argentina Argentina
## 7     Ciprus                  Siprus     Cipro
## 8      India                   India     India
## 9       Kína Republik Rakyat Tiongkok      Cina
## 10    Izrael                  Israel   Israele
##                NAME_NL                NAME_PL   NAME_PT
## 1            Indonesië               Indonezja Indonésia
## 2             Maleisië                 Malezja   Malásia
## 3                Chili                   Chile     Chile
## 4             Bolivia                 Boliwia   Bolívia
## 5                Peru                    Peru      Peru
## 6           Argentinië               Argentyna Argentina
```

```
## 7                Cyprus                      Cypr    Chipre
## 8                 India                     Indie     Índia
## 9  Volksrepubliek China Chińska Republika Ludowa     China
## 10               Israël                    Izrael    Israel
##                         NAME_RU    NAME_SV             NAME_TR
## 1                           Indonesien            Endonezya
## 2                             Malaysia              Malezya
## 3                                  Chile                Şili
## 4                               Bolivia              Bolivya
## 5                                   Peru                Peru
## 6                              Argentina             Arjantin
## 7                                  Cypern   Kıbrıs Cumhuriyeti
## 8                                 Indien            Hindistan
## 9                          Kina Çin Halk Cumhuriyeti
## 10                                Israel               İsrail
##                       NAME_VI        NAME_ZH   WB_NAME
## 1                    Indonesia         Indonesia
## 2                     Malaysia           Malaysia
## 3                        Chile                Chile
## 4                     Bolivia            Bolivia
## 5                        Peru                 Peru
## 6                    Argentina            Argentina
## 7                 Cộng hòa Síp             Cyprus
## 8                        Ấn Độ                India
## 9  Cộng hòa Nhân dân Trung Hoa          China
## 10                       Israel              Israel
##                                                 WB_RULES WB_REGION Shape_Leng
## 1                                                   None       EAP 495.029918
## 2                                                   None       EAP  68.456913
## 3                                                   None       LCR 416.997272
## 4                                                   None       LCR  54.345991
## 5                                                   None       LCR  73.262192
## 6                                                   None       LCR 151.513104
## 7                                                   None       ECA   9.596701
## 8  Stops South of Kashmir Line of Control with Pakistan       SOA 199.047024
## 9                                                   None       EAP 376.667095
## 10                                                  None     Other  11.130630
##    Shape_Area                       geometry
## 1  153.078608 MULTIPOLYGON (((117.7036 4....
## 2   26.703172 MULTIPOLYGON (((117.7036 4....
## 3   76.761813 MULTIPOLYGON (((-69.51009 -...
## 4   92.203587 MULTIPOLYGON (((-69.51009 -...
## 5  106.417089 MULTIPOLYGON (((-69.51009 -...
## 6  278.681073 MULTIPOLYGON (((-67.28475 -...
## 7    0.883311 MULTIPOLYGON (((32.6262 35....
## 8  272.304351 MULTIPOLYGON (((76.82459 35...
## 9  951.364850 MULTIPOLYGON (((110.6851 20...
## 10   1.954304 MULTIPOLYGON (((35.60385 33...
```

```
# First, let's make all the datasets have the reasonable format and
# names and the same variable name for country name so that we can
# merge them later.

# For report replace all the blank with '_'
```

```r
names(report)
```

```
##  [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"
```

```r
names(report) = gsub(" ", "_", names(report))
names(report)
```

```
##  [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"
```

```r
# make all strings lower case
names(report) = str_to_lower(names(report), locale = "en")
# I want to have an ordinal variable from 'life_laddor'
report$life_ladder_floored = floor(report$life_ladder)
head(report, n = 10)
```

```
## # A tibble: 10 x 12
##    country_name  year life_ladder log_gdp_per_capita social_support
##    <chr>        <dbl>       <dbl>              <dbl>          <dbl>
##  1 Afghanistan   2008        3.72               7.35          0.451
##  2 Afghanistan   2009        4.40               7.51          0.552
##  3 Afghanistan   2010        4.76               7.61          0.539
##  4 Afghanistan   2011        3.83               7.58          0.521
##  5 Afghanistan   2012        3.78               7.66          0.521
##  6 Afghanistan   2013        3.57               7.68          0.484
##  7 Afghanistan   2014        3.13               7.67          0.526
##  8 Afghanistan   2015        3.98               7.65          0.529
##  9 Afghanistan   2016        4.22               7.65          0.559
## 10 Afghanistan   2017        2.66               7.65          0.491
## # i 7 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>, life_ladder_floored <dbl>
```

```r
# Now the life_ladder_floored is treated as a categorical variable
# Check for NA values
anyNA(report)
```

```
## [1] TRUE
```

```r
length(is.na(report))
```

```
## [1] 28356
```

```r
nrow(na.omit(report))
```

```
## [1] 2097
```

```r
# At least we should have 2097 observations

# For worldbank
summary(worldbank_long)
```

```
##  Country.Name       Country.Code       Series.Name        Series.Code
##  Length:30329       Length:30329       Length:30329       Length:30329
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       Time       Time.Code             Value
##  Min.   :2005   Length:30329       Min.   :     -3.3
##  1st Qu.:2009   Class :character   1st Qu.:      0.3
##  Median :2014   Mode  :character   Median :      2.9
##  Mean   :2014                      Mean   :  15927.3
##  3rd Qu.:2019                      3rd Qu.:      3.6
##  Max.   :2023                      Max.   :2933010.7
##  NA's   :5                         NA's   :12693
```

```r
# replace all the '.' with '_', make everything lower case
names(worldbank_long) = c("country_name", "country_code", "series_name",
    "series_code", "year", "year_code", "value")
summary(worldbank_long)
```

```
##  country_name       country_code       series_name        series_code
##  Length:30329       Length:30329       Length:30329       Length:30329
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       year       year_code             value
##  Min.   :2005   Length:30329       Min.   :     -3.3
##  1st Qu.:2009   Class :character   1st Qu.:      0.3
##  Median :2014   Mode  :character   Median :      2.9
##  Mean   :2014                      Mean   :  15927.3
##  3rd Qu.:2019                      3rd Qu.:      3.6
##  Max.   :2023                      Max.   :2933010.7
##  NA's   :5                         NA's   :12693
```

```r
worldbank_long = worldbank_long %>%
    select(country_name, country_code, series_name, year, value)
# make the long format into wide format
anyNA(worldbank_long)
```

```
## [1] TRUE
```

```r
length(is.na(worldbank_long))
```

```
## [1] 151645
```

```r
# 227455
worldbank_long2 = na.omit(worldbank_long)
anyNA(worldbank_long2)
```

```
## [1] FALSE
```

```r
nrow(worldbank_long2)
```

```
## [1] 17636
```

```r
worldbank = pivot_wider(worldbank_long2, names_from = series_name, values_from = value)
worldbank
```

```
## # A tibble: 4,540 x 9
##    country_name country_code  year Political Stability ~1 CPIA social protecti~2
##    <chr>        <chr>        <int>                  <dbl>                  <dbl>
##  1 Afghanistan  AFG           2005                  -2.07                   NA
##  2 Afghanistan  AFG           2006                  -2.22                    2
##  3 Afghanistan  AFG           2007                  -2.41                    2
##  4 Afghanistan  AFG           2008                  -2.69                    2.5
##  5 Afghanistan  AFG           2009                  -2.71                    2.5
##  6 Afghanistan  AFG           2010                  -2.58                    2.5
##  7 Afghanistan  AFG           2011                  -2.50                    2.5
##  8 Afghanistan  AFG           2012                  -2.42                    2.5
##  9 Afghanistan  AFG           2013                  -2.52                    2.5
## 10 Afghanistan  AFG           2014                  -2.41                    2.5
## # i 4,530 more rows
## # i abbreviated names:
## #   1: `Political Stability and Absence of Violence/Terrorism: Estimate`,
## #   2: `CPIA social protection rating (1=low to 6=high)`
## # i 4 more variables:
## #   `CPIA policies for social inclusion/equity cluster average (1=low to 6=high)` <dbl>,
## #   `Scientific and technical journal articles` <dbl>, ...
```

```r
# now make all the names resonable
names(worldbank)
```

```
## [1] "country_name"
## [2] "country_code"
## [3] "year"
## [4] "Political Stability and Absence of Violence/Terrorism: Estimate"
## [5] "CPIA social protection rating (1=low to 6=high)"
## [6] "CPIA policies for social inclusion/equity cluster average (1=low to 6=high)"
## [7] "Scientific and technical journal articles"
## [8] "Rule of Law: Estimate"
## [9] "CPIA policy and institutions for environmental sustainability rating (1=low to 6=high)"
```

```
names(worldbank) = c("country_name", "country_code", "year", "political_stability",
    "social_protection", "social_inclusion", "technology", "legal", "environment")
names(worldbank)
```

```
## [1] "country_name"        "country_code"        "year"
## [4] "political_stability" "social_protection"   "social_inclusion"
## [7] "technology"          "legal"               "environment"
```

```
# for geo data
names(geo)
```

```
##  [1] "OBJECTID"   "featurecla" "LEVEL"      "TYPE"       "FORMAL_EN"
##  [6] "FORMAL_FR"  "POP_EST"    "POP_RANK"   "GDP_MD_EST" "POP_YEAR"
## [11] "LASTCENSUS" "GDP_YEAR"   "ECONOMY"    "INCOME_GRP" "FIPS_10_"
## [16] "ISO_A2"     "ISO_A3"     "ISO_A3_EH"  "ISO_N3"     "UN_A3"
## [21] "WB_A2"      "WB_A3"      "CONTINENT"  "REGION_UN"  "SUBREGION"
## [26] "REGION_WB"  "NAME_AR"    "NAME_BN"    "NAME_DE"    "NAME_EN"
## [31] "NAME_ES"    "NAME_FR"    "NAME_EL"    "NAME_HI"    "NAME_HU"
## [36] "NAME_ID"    "NAME_IT"    "NAME_JA"    "NAME_KO"    "NAME_NL"
## [41] "NAME_PL"    "NAME_PT"    "NAME_RU"    "NAME_SV"    "NAME_TR"
## [46] "NAME_VI"    "NAME_ZH"    "WB_NAME"    "WB_RULES"   "WB_REGION"
## [51] "Shape_Leng" "Shape_Area" "geometry"
```

```
# Only the NAME_EN will be of use in merging data later

# I want to merge the report and worldbank dataset according to
# country and year.
length(unique(report$country_name))
```

```
## [1] 165
```

```
unique(report$country_name)
```

```
##   [1] "Afghanistan"              "Albania"
##   [3] "Algeria"                  "Angola"
##   [5] "Argentina"                "Armenia"
##   [7] "Australia"                "Austria"
##   [9] "Azerbaijan"               "Bahrain"
##  [11] "Bangladesh"               "Belarus"
##  [13] "Belgium"                  "Belize"
##  [15] "Benin"                    "Bhutan"
```

```
##  [17] "Bolivia"                  "Bosnia and Herzegovina"
##  [19] "Botswana"                 "Brazil"
##  [21] "Bulgaria"                 "Burkina Faso"
##  [23] "Burundi"                  "Cambodia"
##  [25] "Cameroon"                 "Canada"
##  [27] "Central African Republic" "Chad"
##  [29] "Chile"                    "China"
##  [31] "Colombia"                 "Comoros"
##  [33] "Congo (Brazzaville)"      "Congo (Kinshasa)"
##  [35] "Costa Rica"               "Croatia"
##  [37] "Cuba"                     "Cyprus"
##  [39] "Czechia"                  "Denmark"
##  [41] "Djibouti"                 "Dominican Republic"
##  [43] "Ecuador"                  "Egypt"
##  [45] "El Salvador"              "Estonia"
##  [47] "Eswatini"                 "Ethiopia"
##  [49] "Finland"                  "France"
##  [51] "Gabon"                    "Gambia"
##  [53] "Georgia"                  "Germany"
##  [55] "Ghana"                    "Greece"
##  [57] "Guatemala"                "Guinea"
##  [59] "Guyana"                   "Haiti"
##  [61] "Honduras"                 "Hong Kong S.A.R. of China"
##  [63] "Hungary"                  "Iceland"
##  [65] "India"                    "Indonesia"
##  [67] "Iran"                     "Iraq"
##  [69] "Ireland"                  "Israel"
##  [71] "Italy"                    "Ivory Coast"
##  [73] "Jamaica"                  "Japan"
##  [75] "Jordan"                   "Kazakhstan"
##  [77] "Kenya"                    "Kosovo"
##  [79] "Kuwait"                   "Kyrgyzstan"
##  [81] "Laos"                     "Latvia"
##  [83] "Lebanon"                  "Lesotho"
##  [85] "Liberia"                  "Libya"
##  [87] "Lithuania"                "Luxembourg"
##  [89] "Madagascar"               "Malawi"
##  [91] "Malaysia"                 "Maldives"
##  [93] "Mali"                     "Malta"
##  [95] "Mauritania"               "Mauritius"
##  [97] "Mexico"                   "Moldova"
##  [99] "Mongolia"                 "Montenegro"
## [101] "Morocco"                  "Mozambique"
## [103] "Myanmar"                  "Namibia"
## [105] "Nepal"                    "Netherlands"
## [107] "New Zealand"              "Nicaragua"
## [109] "Niger"                    "Nigeria"
## [111] "North Macedonia"          "Norway"
## [113] "Oman"                     "Pakistan"
## [115] "Panama"                   "Paraguay"
## [117] "Peru"                     "Philippines"
## [119] "Poland"                   "Portugal"
## [121] "Qatar"                    "Romania"
## [123] "Russia"                   "Rwanda"
```

```
## [125] "Saudi Arabia"              "Senegal"
## [127] "Serbia"                    "Sierra Leone"
## [129] "Singapore"                 "Slovakia"
## [131] "Slovenia"                  "Somalia"
## [133] "Somaliland region"         "South Africa"
## [135] "South Korea"               "South Sudan"
## [137] "Spain"                     "Sri Lanka"
## [139] "State of Palestine"        "Sudan"
## [141] "Suriname"                  "Sweden"
## [143] "Switzerland"               "Syria"
## [145] "Taiwan Province of China"  "Tajikistan"
## [147] "Tanzania"                  "Thailand"
## [149] "Togo"                      "Trinidad and Tobago"
## [151] "Tunisia"                   "Turkmenistan"
## [153] "Türkiye"                   "Uganda"
## [155] "Ukraine"                   "United Arab Emirates"
## [157] "United Kingdom"            "United States"
## [159] "Uruguay"                   "Uzbekistan"
## [161] "Venezuela"                 "Vietnam"
## [163] "Yemen"                     "Zambia"
## [165] "Zimbabwe"
```

```
# 165 unique countries
length(unique(report$year))
```

```
## [1] 19
```

```
unique(report$year)
```

```
##  [1] 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2021 2022 2023
## [16] 2007 2020 2006 2005
```

```
# 19 years
length(unique(worldbank$country_name))
```

```
## [1] 253
```

```
length(unique(worldbank$year))
```

```
## [1] 18
```

```
unique(worldbank$year)
```

```
##  [1] 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
## [16] 2020 2021 2022
```

```
# 261 unique countries only 18 years. World bank does not have any
# data on year 2023 yet.

# left join worldbank to report on country_name and year I include
```

```r
# some factors based PESTL external environment analysis to see
# whether other external factors (except for the ones used in the
# report) will better explain the happiness level
full_report = left_join(report, worldbank, by = c("country_name", "year"))
nrow(full_report)
```

```
## [1] 2363
```

```r
# 2363 sample size
names(full_report)
```

```
##  [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"                 "life_ladder_floored"
## [13] "country_code"                    "political_stability"
## [15] "social_protection"               "social_inclusion"
## [17] "technology"                      "legal"
## [19] "environment"
```

```r
# Only keep the necessary columns
full_report2 = full_report %>%
    select(1:9, 12:19)
names(full_report2)
```

```
##  [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"       "life_ladder_floored"
## [11] "country_code"                    "political_stability"
## [13] "social_protection"               "social_inclusion"
## [15] "technology"                      "legal"
## [17] "environment"
```

```r
# Move response variables to the front


########## Geographical Visualization of 2023 happiness level
########## ############

# left join the geo data to report on year 2023 so that I can
# visualize 2023 happiness geographically later.
report_2023 = report %>%
    filter(year == 2023)
length(unique(report_2023$country_name))
```

```
## [1] 138
```

```r
report_2023_geo = left_join(geo, report_2023, by = c(NAME_EN = "country_name"))
summary(report_2023_geo)
```

```
##     OBJECTID      featurecla            LEVEL       TYPE
##  Min.   :  1.0   Length:251         Min.   :2   Length:251
##  1st Qu.: 63.5   Class :character   1st Qu.:2   Class :character
##  Median :126.0   Mode  :character   Median :2   Mode  :character
##  Mean   :126.0                      Mean   :2
##  3rd Qu.:188.5                      3rd Qu.:2
##  Max.   :251.0                      Max.   :2
##
##   FORMAL_EN          FORMAL_FR            POP_EST             POP_RANK
##  Length:251         Length:251         Min.   :0.000e+00   Min.   : 1.00
##  Class :character   Class :character   1st Qu.:1.573e+05   1st Qu.: 9.00
##  Mode  :character   Mode  :character   Median :4.926e+06   Median :12.00
##                                        Mean   :2.961e+07   Mean   :11.49
##                                        3rd Qu.:1.784e+07   3rd Qu.:14.00
##                                        Max.   :1.379e+09   Max.   :18.00
##
##    GDP_MD_EST          POP_YEAR      LASTCENSUS      GDP_YEAR
##  Min.   :       0   Min.   :   0   Min.   : -99   Min.   :   0
##  1st Qu.:    3167   1st Qu.:2017   1st Qu.:2001   1st Qu.:2016
##  Median :   35010   Median :2017   Median :2006   Median :2016
##  Mean   :  481559   Mean   :1945   Mean   :1687   Mean   :1935
##  3rd Qu.:  230418   3rd Qu.:2017   3rd Qu.:2010   3rd Qu.:2016
##  Max.   :21140000   Max.   :2017   Max.   :2012   Max.   :2016
##
##    ECONOMY           INCOME_GRP          FIPS_10_            ISO_A2
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     ISO_A3           ISO_A3_EH            ISO_N3             UN_A3
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     WB_A2              WB_A3            CONTINENT          REGION_UN
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   SUBREGION          REGION_WB           NAME_AR            NAME_BN
##  Length:251         Length:251         Length:251         Length:251
```

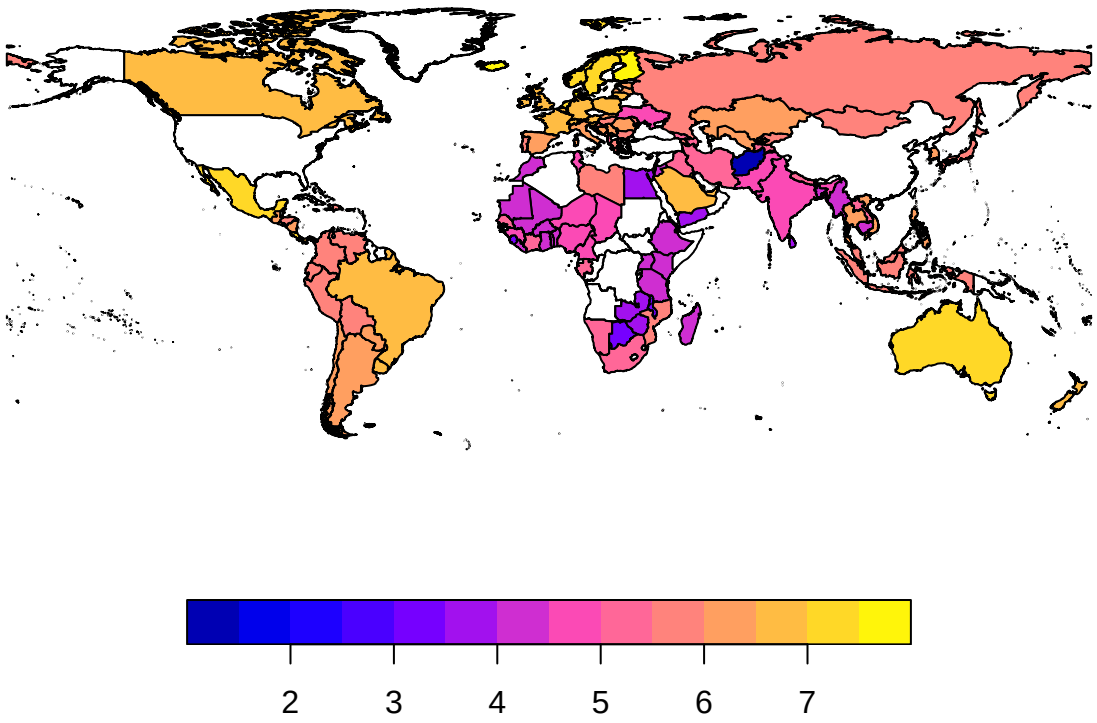
```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    NAME_DE            NAME_EN            NAME_ES            NAME_FR
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    NAME_EL            NAME_HI            NAME_HU            NAME_ID
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    NAME_IT            NAME_JA            NAME_KO            NAME_NL
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    NAME_PL            NAME_PT            NAME_RU            NAME_SV
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    NAME_TR            NAME_VI            NAME_ZH            WB_NAME
##  Length:251         Length:251         Length:251         Length:251
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    WB_RULES           WB_REGION            Shape_Leng          Shape_Area
##  Length:251         Length:251         Min.   :   0.0044   Min.   :   0.0000
##  Class :character   Class :character   1st Qu.:   2.4008   1st Qu.:   0.0512
##  Mode  :character   Mode  :character   Median :  18.8262   Median :   5.8821
##                                        Mean   :  63.1529   Mean   :  61.1288
##                                        3rd Qu.:  50.5104   3rd Qu.:  37.9906
##                                        Max.   :2573.7125   Max.   :2925.3326
##
```

```
##       year       life_ladder    log_gdp_per_capita social_support
##  Min.   :2023   Min.   :1.446   Min.   : 7.147     Min.   :0.3685
##  1st Qu.:2023   1st Qu.:4.675   1st Qu.: 8.670     1st Qu.:0.7108
##  Median :2023   Median :5.907   Median : 9.651     Median :0.8347
##  Mean   :2023   Mean   :5.678   Mean   : 9.573     Mean   :0.7956
##  3rd Qu.:2023   3rd Qu.:6.553   3rd Qu.:10.575     3rd Qu.:0.8949
##  Max.   :2023   Max.   :7.699   Max.   :11.676     Max.   :0.9788
##  NA's   :119    NA's   :119     NA's   :126        NA's   :119
##  healthy_life_expectancy_at_birth freedom_to_make_life_choices
##  Min.   :52.20                    Min.   :0.2283
##  1st Qu.:61.50                    1st Qu.:0.7417
##  Median :66.10                    Median :0.8261
##  Mean   :65.45                    Mean   :0.7969
##  3rd Qu.:70.35                    3rd Qu.:0.8766
##  Max.   :74.60                    Max.   :0.9648
##  NA's   :120                      NA's   :121
##    generosity       perceptions_of_corruption positive_affect  negative_affect
##  Min.   :-0.26821   Min.   :0.1525            Min.   :0.2605   Min.   :0.1138
##  1st Qu.:-0.07116   1st Qu.:0.6496            1st Qu.:0.5879   1st Qu.:0.2283
##  Median : 0.03108   Median :0.7656            Median :0.6709   Median :0.2771
##  Mean   : 0.03606   Mean   :0.7059            Mean   :0.6561   Mean   :0.2900
##  3rd Qu.: 0.14618   3rd Qu.:0.8360            3rd Qu.:0.7372   3rd Qu.:0.3534
##  Max.   : 0.58955   Max.   :0.9479            Max.   :0.8431   Max.   :0.5159
##  NA's   :126        NA's   :125               NA's   :119      NA's   :119
##  life_ladder_floored          geometry
##  Min.   :1.000       MULTIPOLYGON :251
##  1st Qu.:4.000       epsg:4326    :  0
##  Median :5.000       +proj=long...:  0
##  Mean   :5.159
##  3rd Qu.:6.000
##  Max.   :7.000
##  NA's   :119
```

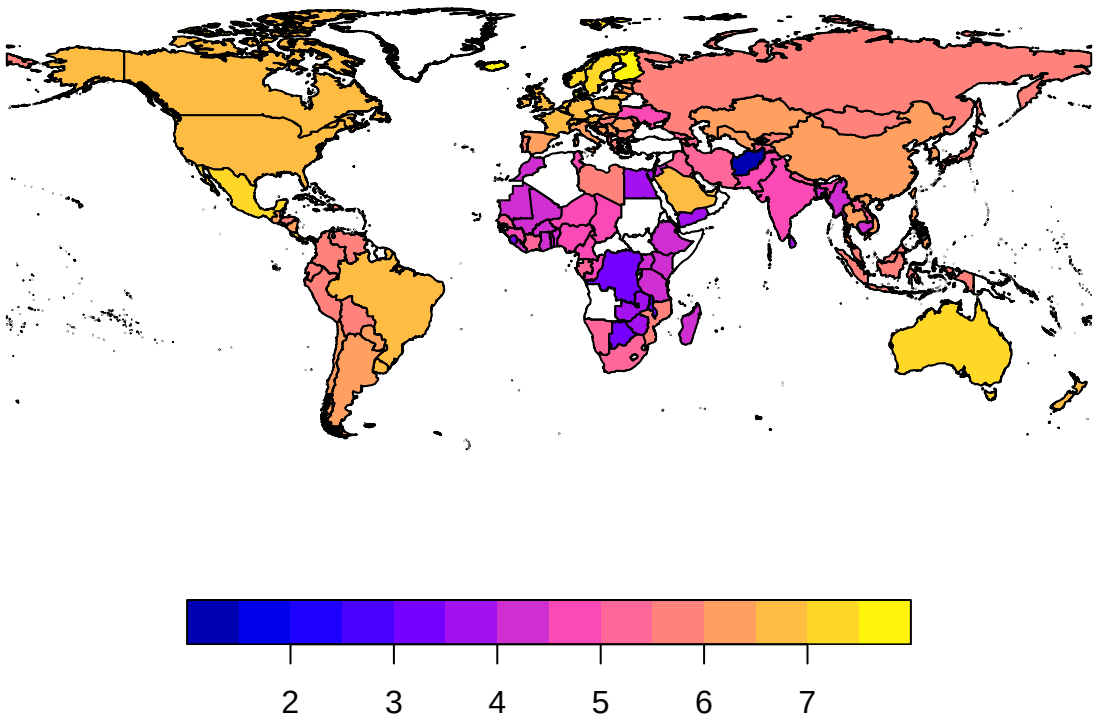
```
length(unique(report_2023_geo$count))
```

```
## [1] 0
```

```
plot(report_2023_geo["life_ladder"], main = "2023 World Happiness Level by Country")
```




```
# We can see that some countries are not matcched, we need to look
# into the names of some of those countries: the US, China, Algeria,
# Sudan, South Sudan, Congo, Angola unique(report_2023$country_name)
# The names in report_2023 is: 'United States' 'China' 'Congo
# (Brazzaville)' (this is Congo) 'Congo (Kinshasa)' (this is DR
# Congo). No country name for Algeria, Sudan, South Sudan, and Angola
# unique(geo$NAME_EN) The names in geo is: 'United States of
# America', 'People's Republic of China', 'Republic of the Congo',
# and 'Democratic Republic of the Congo' Now we replace the name in
# report so that we can merge
report_2023$country_name = ifelse(report_2023$country_name == "United States",
    "United States of America", report_2023$country_name)
report_2023$country_name = ifelse(report_2023$country_name == "China",
    "People's Republic of China", report_2023$country_name)
report_2023$country_name = ifelse(report_2023$country_name == "Congo (Brazzaville)",
    "Republic of the Congo", report_2023$country_name)
report_2023$country_name = ifelse(report_2023$country_name == "Congo (Kinshasa)",
    "Democratic Republic of the Congo", report_2023$country_name)
report_2023_geo2 = left_join(geo, report_2023, by = c(NAME_EN = "country_name"))
plot(report_2023_geo2["life_ladder"], main = "2023 World Happiness Level by Country")
```




```
# We can see that in general, the happiness level in Noeth America,
# Europe, and Australia is higher than that in South America, Asia,
# and Africa.
```

## Q1: Choose 6 Variables and Make a ggpairs graph

```
library(leaps)

######### Q1: Choose 6 variables ############### Choose 6 most
######### important variables: I want to use life_ladder_floor as
######### response variable (categorical), and the other 5 variables
######### as explanatory variables, for my classification tree

# Find 5 best variables with best subset: among all the variables
# (report variables + other variables)
response = full_report$life_ladder_floored
names(full_report)
```

```
##  [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"                  "life_ladder_floored"
## [13] "country_code"                     "political_stability"
## [15] "social_protection"                "social_inclusion"
## [17] "technology"                       "legal"
## [19] "environment"
```

```
predictors = full_report[, c(4:9, 14:19)]
# Perform best subset selection, choose 5 best explanatory variables
set.seed(847888)
best_subset = regsubsets(response ~ ., data = predictors, nvmax = 5, really.big = T)
# plot(best_subset, scale='adjr2')
bestsubset_sum = summary(best_subset)
bestsubset_sum$adjr2
```

```
## [1] 0.2282166 0.2989545 0.3289106 0.3432677 0.3495269
```

```
best_model_index = which.max(summary(best_subset)$adjr2)
best_model_index
```

```
## [1] 5
```

```
bestsubset_sum
```

```
## Subset selection object
## Call: regsubsets.formula(response ~ ., data = predictors, nvmax = 5,
##     really.big = T)
## 12 Variables  (and intercept)
##                                 Forced in Forced out
## log_gdp_per_capita                  FALSE      FALSE
## social_support                      FALSE      FALSE
## healthy_life_expectancy_at_birth    FALSE      FALSE
## freedom_to_make_life_choices        FALSE      FALSE
## generosity                          FALSE      FALSE
## perceptions_of_corruption           FALSE      FALSE
## political_stability                 FALSE      FALSE
## social_protection                   FALSE      FALSE
## social_inclusion                    FALSE      FALSE
## technology                          FALSE      FALSE
## legal                               FALSE      FALSE
## environment                         FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          log_gdp_per_capita social_support healthy_life_expectancy_at_birth
## 1  ( 1 ) "*"                " "            " "
## 2  ( 1 ) "*"                "*"            " "
## 3  ( 1 ) "*"                "*"            " "
## 4  ( 1 ) "*"                "*"            " "
## 5  ( 1 ) "*"                "*"            " "
##          freedom_to_make_life_choices generosity perceptions_of_corruption
## 1  ( 1 ) " "                          " "        " "
## 2  ( 1 ) " "                          " "        " "
```

```
## 3  ( 1 ) "*"                          " "         " "
## 4  ( 1 ) "*"                          " "         " "
## 5  ( 1 ) "*"                          " "         " "
##          political_stability social_protection social_inclusion technology
## 1  ( 1 ) " "                 " "                " "              " "
## 2  ( 1 ) " "                 " "                " "              " "
## 3  ( 1 ) " "                 " "                " "              " "
## 4  ( 1 ) "*"                 " "                " "              " "
## 5  ( 1 ) "*"                 " "                " "              " "
##          legal environment
## 1  ( 1 ) " "   " "
## 2  ( 1 ) " "   " "
## 3  ( 1 ) " "   " "
## 4  ( 1 ) " "   " "
## 5  ( 1 ) " "   "*"
```

```
# According to the model selected by the best subset model selection,
# the best five variables should be: log_gdp_per_capita,
# social_support, freedom_to_make_life_choices, political_stability,
# and environment
summary(lm(life_ladder_floored ~ log_gdp_per_capita + social_support +
    freedom_to_make_life_choices + political_stability + environment, data = full_report))
```

```
##
## Call:
## lm(formula = life_ladder_floored ~ log_gdp_per_capita + social_support +
##     freedom_to_make_life_choices + political_stability + environment,
##     data = full_report)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9585 -0.4213 -0.0437  0.4255  1.6964
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -2.65468    0.38556  -6.885 1.40e-11 ***
## log_gdp_per_capita           0.51047    0.04285  11.912  < 2e-16 ***
## social_support               1.70472    0.23015   7.407 4.16e-13 ***
## freedom_to_make_life_choices 1.15833    0.19300   6.002 3.30e-09 ***
## political_stability         -0.15840    0.03696  -4.286 2.10e-05 ***
## environment                  0.12035    0.05825   2.066   0.0392 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6399 on 630 degrees of freedom
##   (   1727    )
## Multiple R-squared:  0.3827, Adjusted R-squared:  0.3778
## F-statistic: 78.13 on 5 and 630 DF,  p-value: < 2.2e-16
```

```
# Adjusted R-squared: 0.3778 According to The World Happiness Report,
# the R-squared reached by the variables used in the report is around
# 0.34 (without the effect variables). Here the highest R-squared is
# similar to that, so we can conclude that the report has also done
```

```r
# its best to explain the world happiness level.  However, we have
# different best explanatory variables here. According to the model
# selected by the best subset model selection, the best five
# variables should be: log_gdp_per_capita, social_support,
# freedom_to_make_life_choices, political_stability, and environment
# Hence, my most important 6 variables are: life_ladder_floored,
# log_gdp_per_capita, social_support, freedom_to_make_life_choices,
# political_stability, and environment.  In the six variables:
# life_ladder_floored can be viewed as categorical, the rest are
# continuous.  Transform the variable into a categorical variable
six_var_categorical = full_report[, c(12, 4, 5, 7, 14, 19)]
six_var_categorical$life_ladder_floored = factor(six_var_categorical$life_ladder_floored)

names(full_report)
```

```
##  [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"                 "life_ladder_floored"
## [13] "country_code"                    "political_stability"
## [15] "social_protection"               "social_inclusion"
## [17] "technology"                      "legal"
## [19] "environment"
```

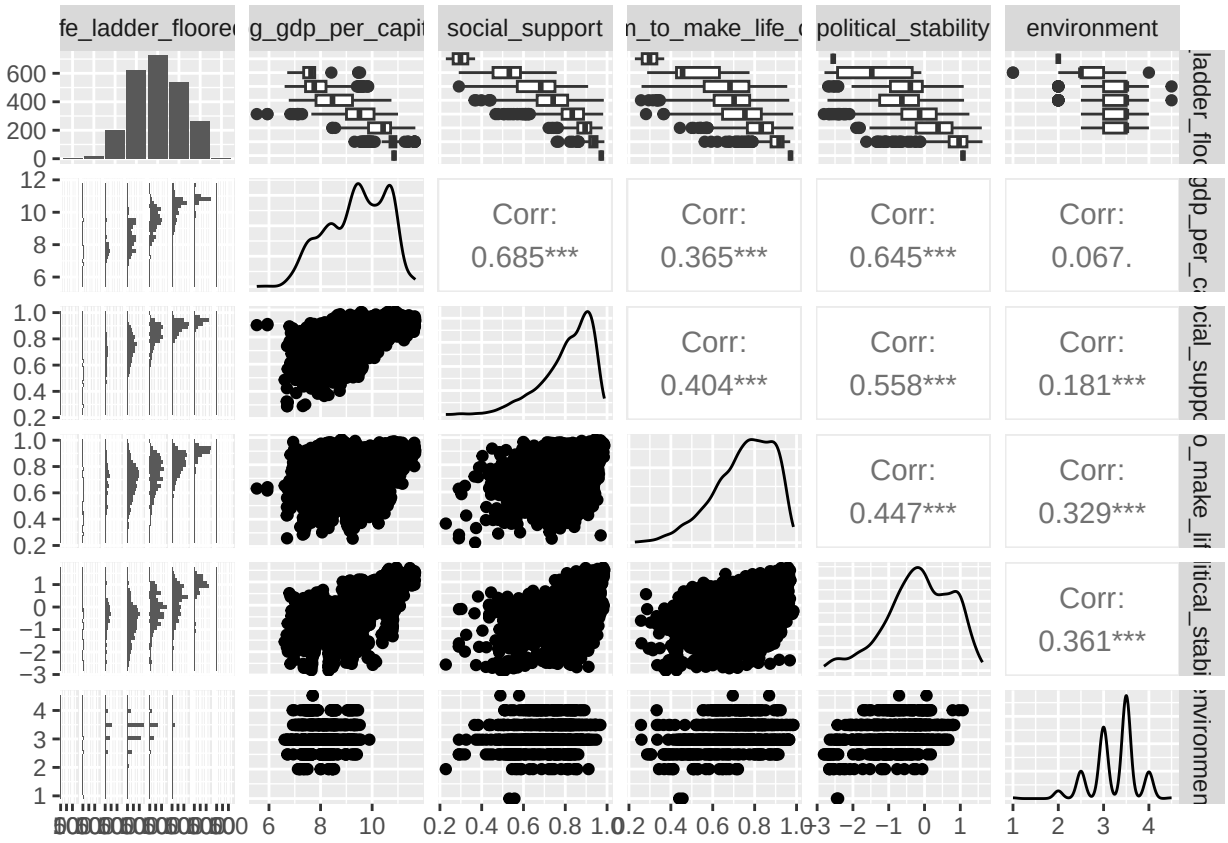
```r
six_var_continuous = full_report[, c(3, 4, 5, 7, 14, 19)]



############## ggpairs graph ############
library(GGally)
```

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

```r
ggpairs(six_var_categorical)
```

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

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




```
# As shown on the graph, between the five continuous variables, all
# the correlation coefficients are positive and most are
# significantly positive. The categorical variable distribution is
# right skewed, so are most of its distributions in different
# category.
```

## Q6: Build a ggplot: Build a visually impressive ggplot to show the relationship between at least three variables.

Now I want to build a ggplot and show the relationship between all six variables. Here I want to use the continuous version of response variable so that I can calculate the correlation coefficients.

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

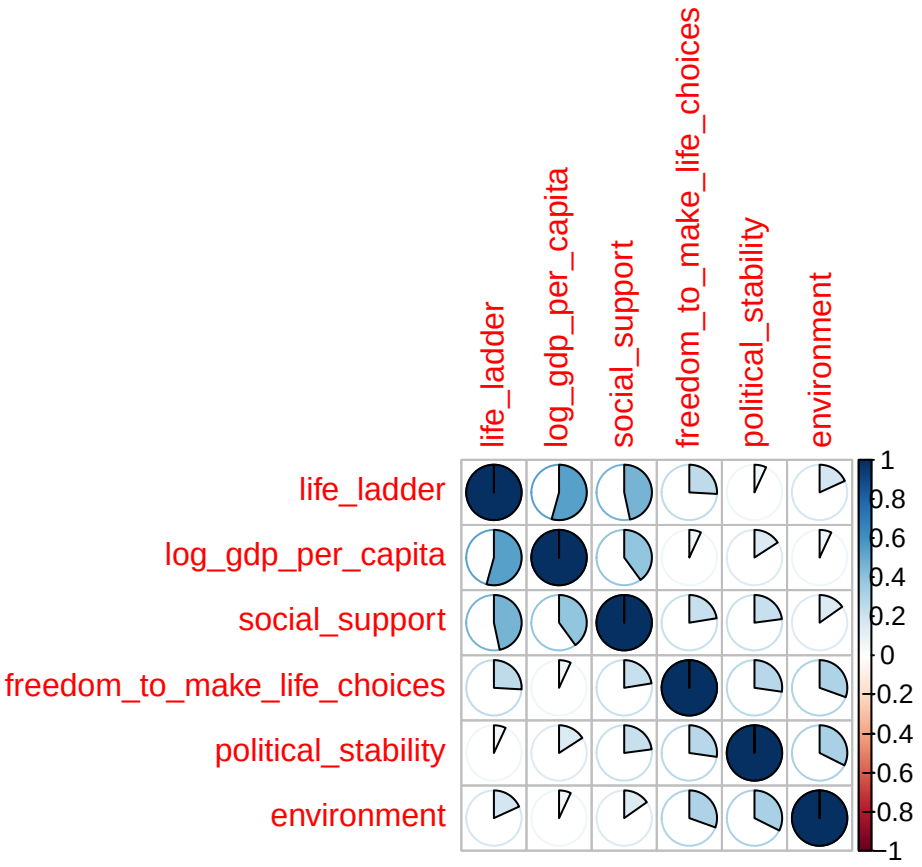
```
library(ggcorrplot)

relationship = na.omit(six_var_continuous)
nrow(relationship)
```

```
## [1] 636
```

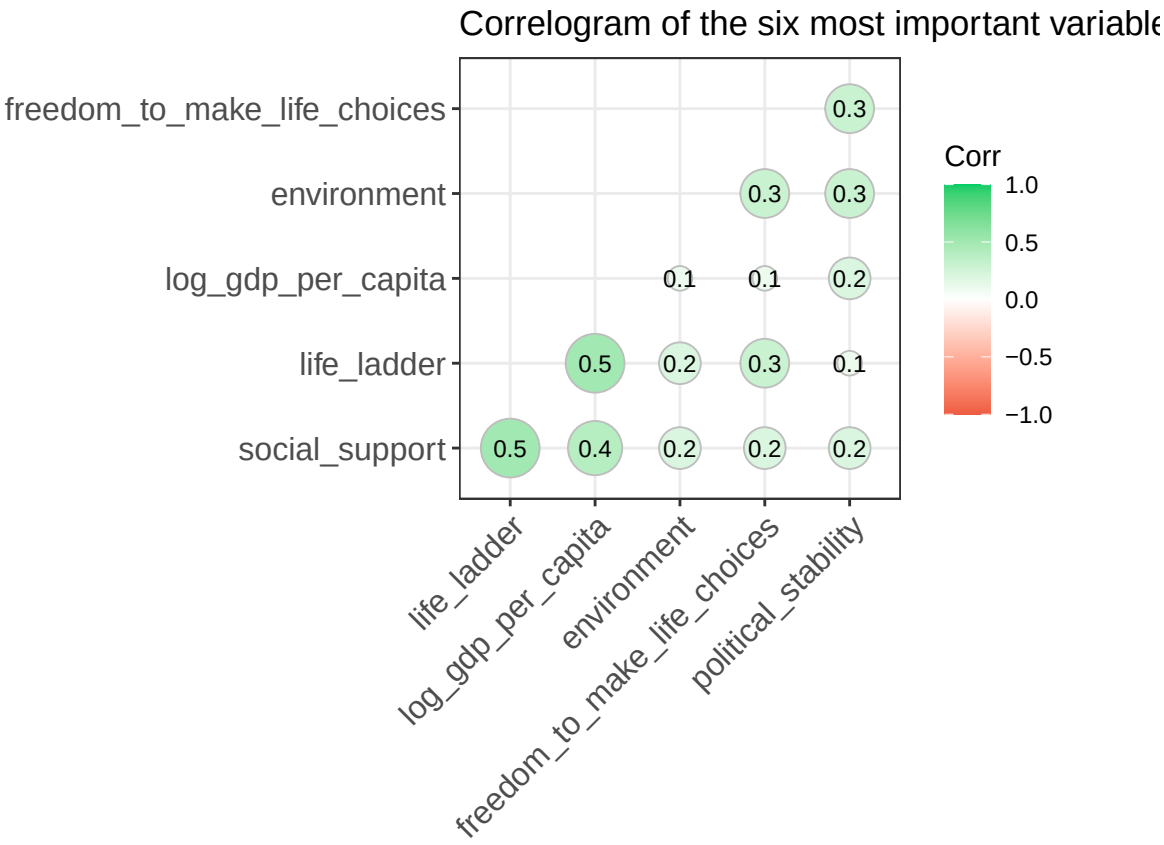
```r
# We can create a correlation plot with corrplot using the method
# 'pie'
M1 = cor(relationship)
M1
```

```
##                              life_ladder log_gdp_per_capita social_support
## life_ladder                   1.00000000         0.54311527      0.4664688
## log_gdp_per_capita            0.54311527         1.00000000      0.3998980
## social_support                0.46646881         0.39989804      1.0000000
## freedom_to_make_life_choices  0.25888620         0.06876333      0.2242692
## political_stability           0.06920026         0.15859095      0.2290513
## environment                   0.18184529         0.06948703      0.1525420
##                              freedom_to_make_life_choices political_stability
## life_ladder                                     0.25888620          0.06920026
## log_gdp_per_capita                              0.06876333          0.15859095
## social_support                                  0.22426921          0.22905127
## freedom_to_make_life_choices                    1.00000000          0.27267093
## political_stability                             0.27267093          1.00000000
## environment                                     0.30482850          0.32388596
##                              environment
## life_ladder                   0.18184529
## log_gdp_per_capita            0.06948703
## social_support                0.15254195
## freedom_to_make_life_choices  0.30482850
## political_stability           0.32388596
## environment                   1.00000000
```

```r
corrplot(M1, method = "pie")
```




```
# We can also use ggcorrplot to create a more reader-friendly visual
corr = round(cor(relationship), 1)
ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE, lab_size = 3,
    method = "circle", colors = c("tomato2", "white", "springgreen3"),
    title = "Correlogram of the six most important variables", ggtheme = theme_bw)
```

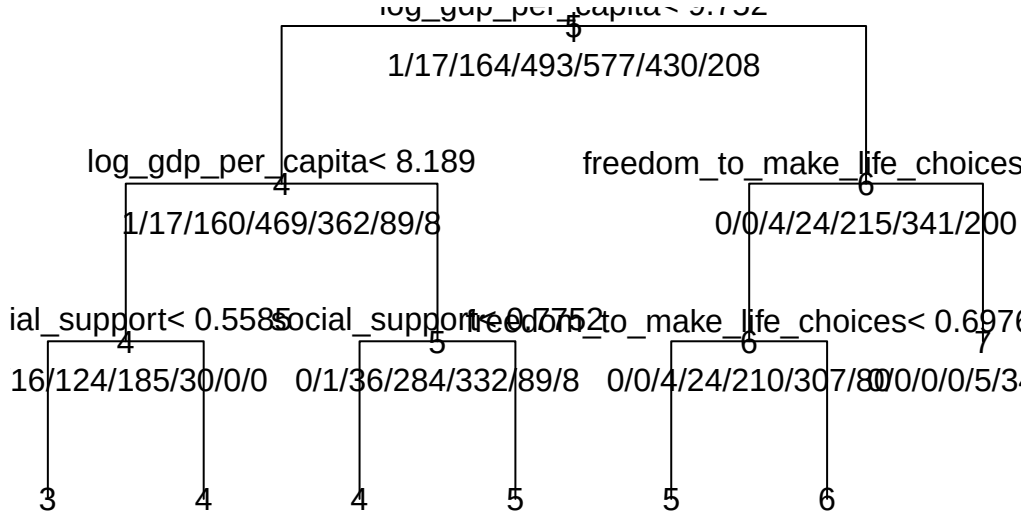



```
# We can tell that all correlations are positive.
```

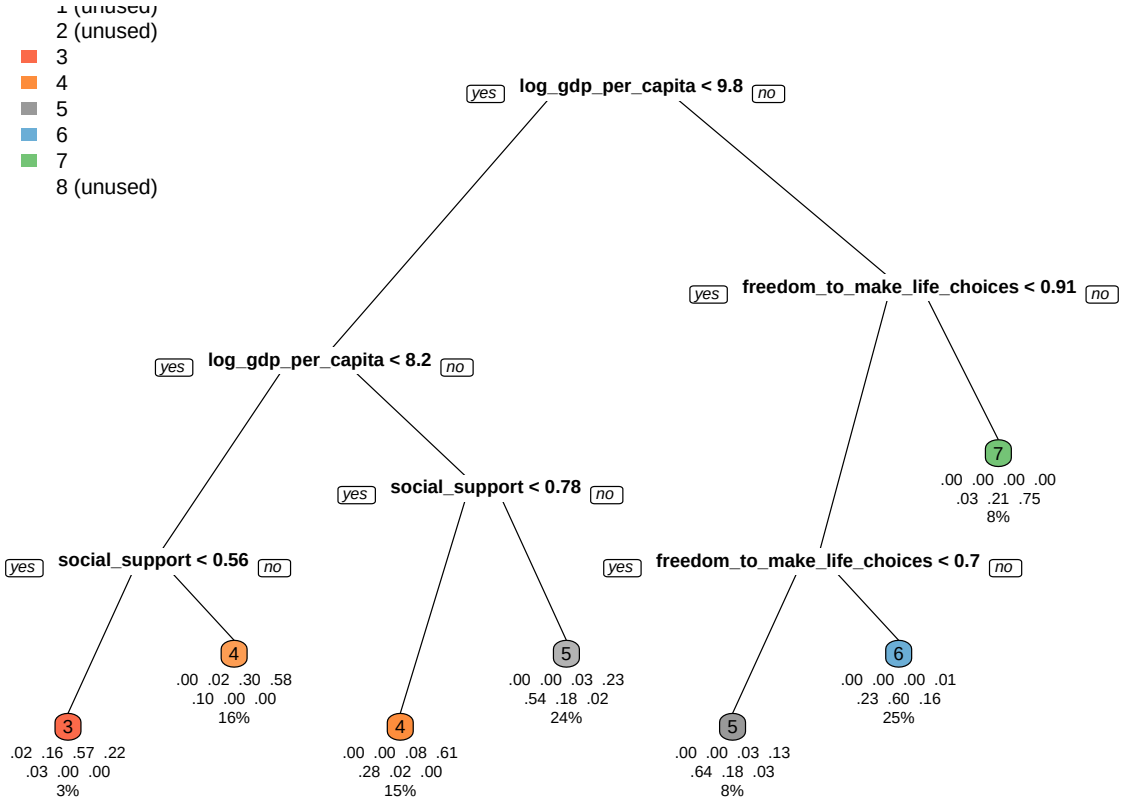
## Q3: Classification tree: pick a categorical variable from your variable list, predict it with the other five variables.

```
# Following the variable selection above, I will do a classification
# tree of life_ladder as a function of the other five variables.

library(rpart)
library(rpart.plot)

X = six_var_categorical[, c("log_gdp_per_capita", "social_support", "freedom_to_make_life_choices",
    "political_stability", "environment")]
y = six_var_categorical$life_ladder_floored

# Split the data into training and testing sets
set.seed(123)  # For reproducibility
train_index = sample(1:nrow(six_var_categorical), 0.8 * nrow(six_var_categorical))  # 80% for training,
X_train = X[train_index, ]
X_test = X[-train_index, ]
y_train = y[train_index]
y_test = y[-train_index]
```

```r
train_data = cbind(X_train, y_train)


# Build the classification tree
tree_model = rpart(y_train ~ log_gdp_per_capita + social_support + freedom_to_make_life_choices +
    political_stability + environment, data = train_data, method = "class")

# Plot the tree ugly version
plot(tree_model, uniform = TRUE)
text(tree_model, use.n = TRUE, all = TRUE, cex = 1)
```




```r
# beautified version
rpart.plot(tree_model, yesno = 2, type = 0, extra = 104, under = TRUE,
    fallen.leaves = FALSE)
```




```
# Only four predictors out of five are used and only 5 levels: level
# 3-7 are used. Therefore, we will only have predictions of level 3-7
# based on this tree model.

# Make predictions
predictions = predict(tree_model, newdata = data.frame(X_test), type = "class")

# Calculate confusion matrix
conf_matrix = table(y_test, predictions)
conf_matrix
```

```
##       predictions
## y_test  1  2  3  4  5  6  7  8
##      1  0  0  1  0  0  0  0  0
##      2  0  0  1  3  0  0  0  0
##      3  0  0  4 29  1  0  0  0
##      4  0  0  4 88 33  1  0  0
##      5  0  0  1 25 83 36  2  0
##      6  0  0  0  1 33 65  8  0
##      7  0  0  0  0  3 18 32  0
##      8  0  0  0  0  0  0  1  0
```

```
# Another way to calculate the confusion matrix
library(caret)
```

```
##      lattice
```

```
##
##     'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
conf_matrix <- confusionMatrix(data = predictions, reference = y_test)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8
##          1  0  0  0  0  0  0  0  0
##          2  0  0  0  0  0  0  0  0
##          3  1  1  4  4  1  0  0  0
##          4  0  3 29 88 25  1  0  0
##          5  0  0  1 33 83 33  3  0
##          6  0  0  0  1 36 65 18  0
##          7  0  0  0  0  2  8 32  1
##          8  0  0  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.5751
##                  95% CI : (0.5291, 0.6201)
##     No Information Rate : 0.3108
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4319
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity          0.000000 0.000000 0.117647   0.6984   0.5646   0.6075
## Specificity          1.000000 1.000000 0.984055   0.8329   0.7853   0.8497
## Pos Pred Value            NaN      NaN 0.363636   0.6027   0.5425   0.5417
## Neg Pred Value       0.997886 0.991543 0.935065   0.8838   0.8000   0.8810
## Prevalence           0.002114 0.008457 0.071882   0.2664   0.3108   0.2262
## Detection Rate       0.000000 0.000000 0.008457   0.1860   0.1755   0.1374
## Detection Prevalence 0.000000 0.000000 0.023256   0.3087   0.3235   0.2537
## Balanced Accuracy    0.500000 0.500000 0.550851   0.7656   0.6750   0.7286
##                      Class: 7 Class: 8
## Sensitivity           0.60377 0.000000
## Specificity           0.97381 1.000000
## Pos Pred Value        0.74419      NaN
## Neg Pred Value        0.95116 0.997886
## Prevalence            0.11205 0.002114
## Detection Rate        0.06765 0.000000
## Detection Prevalence  0.09091 0.000000
## Balanced Accuracy     0.78879 0.500000
```

```r
# This code will build a classification tree using the rpart package,
# plot it using rpart.plot, and then make predictions on the test
# set. Finally, it calculates the confusion matrix.

# To follow down the tree for example predictions, we use the predict
# function: Example prediction 1
example1 = data.frame(X_test[1, ])
prediction1 = predict(tree_model, newdata = example1, type = "class")
print("Example 1 Prediction:")
```

```
## [1] "Example 1 Prediction:"
```

```r
print(prediction1)
```

```
## 1
## 3
## Levels: 1 2 3 4 5 6 7 8
```

```r
# The first prediction is 3

# Example prediction 2
example2 = data.frame(X_test[10, ])
prediction2 = predict(tree_model, newdata = example2, type = "class")
print("Example 2 Prediction:")
```

```
## [1] "Example 2 Prediction:"
```

```r
print(prediction2)
```

```
## 1
## 6
## Levels: 1 2 3 4 5 6 7 8
```

```r
# The 10th prediction is 6
```

## Q4: pick a continuous variable, build a model using model selection and dimension reduction tools.

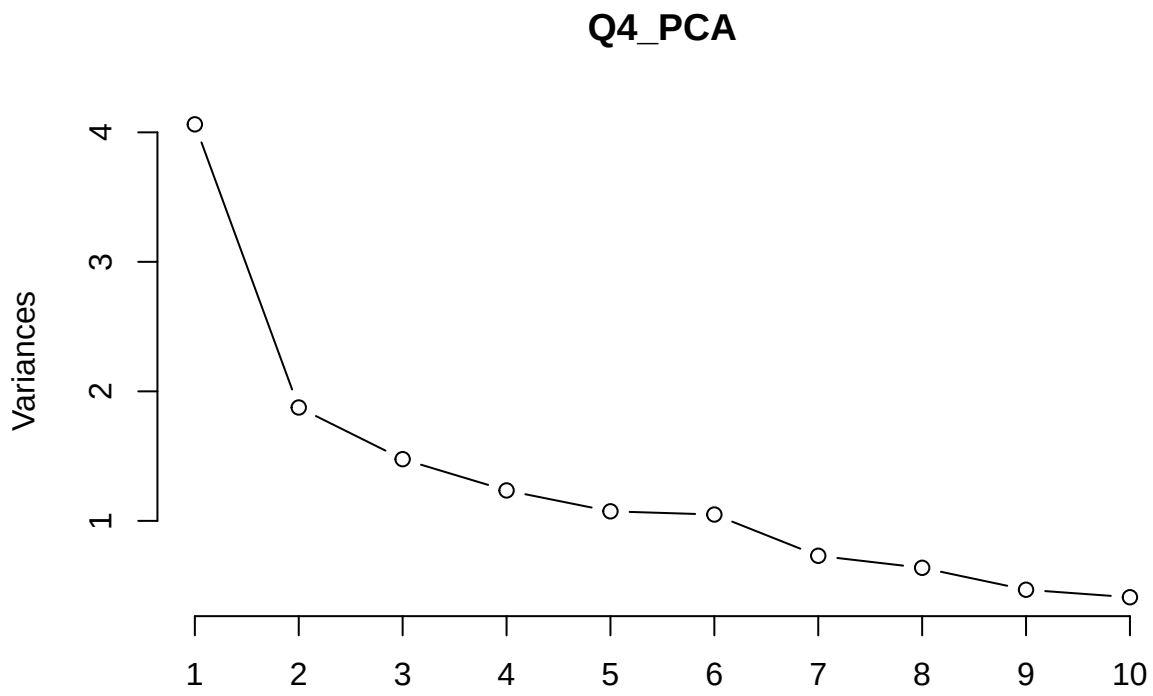
```r
# When we were selecting the most important variables, we can
# actually make use of all of them by using PCA.  Here I use the
# continuous version of life_ladder as response variable, and the
# rest as explanatory variables, and fit a PCA model to explain the
# variable life_ladder

Q4 = full_report[, c(3:11, 14:19)]
names(Q4)
```

```
##  [1] "life_ladder"                    "log_gdp_per_capita"
##  [3] "social_support"                 "healthy_life_expectancy_at_birth"
##  [5] "freedom_to_make_life_choices"   "generosity"
##  [7] "perceptions_of_corruption"      "positive_affect"
##  [9] "negative_affect"                "political_stability"
## [11] "social_protection"              "social_inclusion"
## [13] "technology"                     "legal"
## [15] "environment"
```

```
library(stats)
# 1. Extract predictor variables
predictors = Q4[, !(names(Q4) %in% c("life_ladder"))]
# 2. Perform PCA and standardize all the predictors
Q4_PCA = prcomp(~., data = predictors, scale = TRUE)
summary(Q4_PCA)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.0153 1.3694 1.2151 1.1112 1.03613 1.02418 0.85460
## Proportion of Variance 0.2901 0.1340 0.1055 0.0882 0.07668 0.07492 0.05217
## Cumulative Proportion  0.2901 0.4241 0.5295 0.6177 0.69441 0.76934 0.82150
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.79840 0.68423 0.64052 0.59602 0.54273 0.50798 0.27430
## Proportion of Variance 0.04553 0.03344 0.02931 0.02537 0.02104 0.01843 0.00537
## Cumulative Proportion  0.86703 0.90048 0.92978 0.95515 0.97619 0.99463 1.00000
```

```
# We can see that the total (cumulative) variance explained by the
# first 10 principal components is as high as 0.92978 It is enough
# that we use the first 10 PC.
plot(Q4_PCA, type = "lines")
```




```
# Now we fit a model based on the results of PCA Extract the loadings
# matrix from Q4_PCA. This contains the loadings matrix, which
# represents the weights of the original variables on each principal
# component.
Q4_loadings_10 = Q4_PCA$rotation[, 1:10]
Q4_scaled_predictor = scale(predictors)
# Get the scaled PC values
Q4_PC_scaled = Q4_scaled_predictor %*% Q4_loadings_10
# Unscale the PC values = PC*PC_sd + PC_mean, because PC are all
# centered, means are all 0. we only need PC*PC_sd
Q4_PC_unscaled = Q4_PC_scaled * Q4_PCA$sdev
Q4_dat = as.data.frame(cbind(Q4[, 1], Q4_PC_scaled))
head(Q4_dat)
```

```
##   life_ladder       PC1          PC2       PC3        PC4        PC5        PC6
## 1    3.723590 -5.534765  0.818129006 1.0679933 -1.4896959 -0.1958943 -0.6277448
## 2    4.401778 -5.130759  0.212530840 0.6087360 -1.2753063 -0.2213845 -0.9458892
## 3    4.758381 -5.110417  0.440437548 0.8682533 -1.2616520 -0.6628492 -0.7482431
## 4    3.831719 -5.281874  0.844779478 0.6951901 -1.1069075 -0.6979486 -0.8569621
## 5    3.782938 -4.915211 -0.009845828 0.6971329 -1.1071351 -0.1020360 -0.7706431
## 6    3.572100 -5.092191  0.795251655 0.8481510 -0.8605506 -0.3804411 -0.7253653
##           PC7       PC8         PC9        PC10
## 1  0.39376309 -1.634370 -1.14699987  0.53084275
## 2  0.18010891 -1.670377 -0.83640506  0.30002641
## 3 -0.01821867 -1.299232 -0.07674962  0.01467479
## 4  0.37428588 -1.622908  0.15506139 -0.27304169
```

```
## 5  0.23007120 -1.798240  0.85826651  0.15516913
## 6 -0.09303453 -1.301650  0.30929990  0.43748172
```

```
# Perform regression
Q4_lm = lm(life_ladder ~ ., data = Q4_dat)
summary(Q4_lm)
```

```
##
## Call:
## lm(formula = life_ladder ~ ., data = Q4_dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86725 -0.39728 -0.01927  0.42007  1.67563
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.064148   0.089520  56.570  < 2e-16 ***
## PC1          0.157178   0.019112   8.224 1.49e-15 ***
## PC2          0.002523   0.039183   0.064  0.94868
## PC3         -0.298117   0.026389 -11.297  < 2e-16 ***
## PC4         -0.139650   0.051294  -2.723  0.00669 **
## PC5          0.171596   0.082141   2.089  0.03717 *
## PC6         -0.013783   0.066193  -0.208  0.83514
## PC7         -0.192564   0.036050  -5.342 1.36e-07 ***
## PC8          0.085296   0.072258   1.180  0.23834
## PC9         -0.040394   0.039394  -1.025  0.30563
## PC10        -0.058178   0.041507  -1.402  0.16159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.581 on 538 degrees of freedom
##   (   1814    )
## Multiple R-squared:  0.3791, Adjusted R-squared:  0.3675
## F-statistic: 32.85 on 10 and 538 DF,  p-value: < 2.2e-16
```

```
# Adjusted R-squared: 0.3675

# Compared to the best subsets conducted above, which has an Adjusted
# R-squared of 0.3778, the PCA generated variables perform slightly
# worse since they only have an Adjusted R-squared of 0.3675.
```