Happiness Score Prediction

RABIN NEPAL (U00901360)

2023-11-26

Necessary Library Imports

Data Import and Cleanup

```
# get filenames from data directory
files = list.files(DATA_DIR)
print(files)
```

```
## [1] "2015.csv" "2016.csv" "2017.csv" "2018.csv" "2019.csv" "2020.csv" "2021.csv"
## [8] "2022.csv" "2023.csv"
```

```
# read csv files for world happiness data for each year
# data_2015 = read.csv("2015.csv")
# data_2016 = read.csv("2016.csv")
# data_2017 = read.csv("2017.csv")
# data_2018 = read.csv("2018.csv")
# data_2019 = read.csv("2019.csv")
# data_2020 = read.csv("2020.csv")
# data_2021 = read.csv("2021.csv")
# data_2022 = read.csv("2022.csv")
# data_2023 = read.csv("2023.csv")

# read csv files for world happiness data for each year
for(filename in files){
  year = substring(filename,1,4)
  # set name of variables dynamically as data_{year}
  variable_name = paste0("data_",year)
  VARIABLE_NAMES = c(VARIABLE_NAMES,variable_name)
  # print(variable_name)

  # read csv file
  year_data = read.csv(filename,stringsAsFactors = F)
  assign(variable_name,year_data)

  # see dimension of data for each year
  print(paste("Shape of data for ", year, "is :", dim(year_data)[1], "rows and", dim(year_data)[2], "co
}
```

```
## [1] "Shape of data for  2015 is : 158 rows and 12 columns."
## [1] "Shape of data for  2016 is : 157 rows and 13 columns."
## [1] "Shape of data for  2017 is : 155 rows and 12 columns."
## [1] "Shape of data for  2018 is : 156 rows and 9 columns."
## [1] "Shape of data for  2019 is : 156 rows and 9 columns."
## [1] "Shape of data for  2020 is : 153 rows and 20 columns."
## [1] "Shape of data for  2021 is : 149 rows and 20 columns."
## [1] "Shape of data for  2022 is : 146 rows and 12 columns."
## [1] "Shape of data for  2023 is : 137 rows and 19 columns."
```

```r
# see samples of each year data
# head(data_2015)
for(i in 1:TOTAL_YEARS){
  print(paste("A sample of data for year", 2014+i,":"))
  dummy_year = get(VARIABLE_NAMES[i])
  glimpse(dummy_year)
}
```

```
## [1] "A sample of data for year 2015 :"
## Rows: 158
## Columns: 12
## $ Country           <chr> "Switzerland", "Iceland", "Denmark", "Norway", "Cana~
## $ Region            <chr> "Western Europe", "Western Europe", "Western Europe"~
## $ Happiness.Rank    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Happiness.Score   <dbl> 7.587, 7.561, 7.527, 7.522, 7.427, 7.406, 7.378, 7.3~
## $ Standard.Error    <dbl> 0.03411, 0.04884, 0.03328, 0.03880, 0.03553, 0.03140~
## $ GDP.per.Capita    <dbl> 1.39651, 1.30232, 1.32548, 1.45900, 1.32629, 1.29025~
## $ Social.Support    <dbl> 1.34951, 1.40223, 1.36058, 1.33095, 1.32261, 1.31826~
## $ Life.Expectancy   <dbl> 0.94143, 0.94784, 0.87464, 0.88521, 0.90563, 0.88911~
## $ Freedom           <dbl> 0.66557, 0.62877, 0.64938, 0.66973, 0.63297, 0.64169~
## $ Corruption        <dbl> 0.41978, 0.14145, 0.48357, 0.36503, 0.32957, 0.41372~
## $ Generosity        <dbl> 0.29678, 0.43630, 0.34139, 0.34699, 0.45811, 0.23351~
## $ Dystopia.Residual <dbl> 2.51738, 2.70201, 2.49204, 2.46531, 2.45176, 2.61955~
## [1] "A sample of data for year 2016 :"
## Rows: 157
## Columns: 13
## $ Country                   <chr> "Denmark", "Switzerland", "Iceland", "Norway~
## $ Region                    <chr> "Western Europe", "Western Europe", "Western~
## $ Happiness.Rank            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1~
## $ Happiness.Score           <dbl> 7.526, 7.509, 7.501, 7.498, 7.413, 7.404, 7.~
## $ Lower.Confidence.Interval <dbl> 7.460, 7.428, 7.333, 7.421, 7.351, 7.335, 7.~
## $ Upper.Confidence.Interval <dbl> 7.592, 7.590, 7.669, 7.575, 7.475, 7.473, 7.~
## $ GDP.per.Capita            <dbl> 1.44178, 1.52733, 1.42666, 1.57744, 1.40598,~
## $ Social.Support            <dbl> 1.16374, 1.14524, 1.18326, 1.12690, 1.13464,~
## $ Life.Expectancy           <dbl> 0.79504, 0.86303, 0.86733, 0.79579, 0.81091,~
## $ Freedom                   <dbl> 0.57941, 0.58557, 0.56624, 0.59609, 0.57104,~
## $ Corruption                <dbl> 0.44453, 0.41203, 0.14975, 0.35776, 0.41004,~
## $ Generosity                <dbl> 0.36171, 0.28083, 0.47678, 0.37895, 0.25492,~
## $ Dystopia.Residual         <dbl> 2.73939, 2.69463, 2.83137, 2.66465, 2.82596,~
## [1] "A sample of data for year 2017 :"
## Rows: 155
## Columns: 12
## $ Country           <chr> "Norway", "Denmark", "Iceland", "Switzerland", "Finl~
## $ Happiness.Rank    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Happiness.Score   <dbl> 7.537, 7.522, 7.504, 7.494, 7.469, 7.377, 7.316, 7.3~
## $ Whisker.high      <dbl> 7.594445, 7.581728, 7.622030, 7.561772, 7.527542, 7.~
## $ Whisker.low       <dbl> 7.479556, 7.462272, 7.385970, 7.426227, 7.410458, 7.~
## $ GDP.per.Capita    <dbl> 1.616463, 1.482383, 1.480633, 1.564980, 1.443572, 1.~
## $ Social.Support    <dbl> 1.533524, 1.551122, 1.610574, 1.516912, 1.540247, 1.~
## $ Life.Expectancy   <dbl> 0.7966665, 0.7925655, 0.8335521, 0.8581313, 0.809157~
## $ Freedom           <dbl> 0.6354226, 0.6260067, 0.6271626, 0.6200706, 0.617950~
## $ Generosity        <dbl> 0.36201224, 0.35528049, 0.47554022, 0.29054928, 0.24~
## $ Corruption        <dbl> 0.31596383, 0.40077007, 0.15352656, 0.36700729, 0.38~
## $ Dystopia.Residual <dbl> 2.277027, 2.313707, 2.322715, 2.276716, 2.430182, 2.~
## [1] "A sample of data for year 2018 :"
## Rows: 156
## Columns: 9
## $ Overall.rank    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ Country         <chr> "Finland", "Norway", "Denmark", "Iceland", "Switzerlan~
## $ Happiness.Score <dbl> 7.632, 7.594, 7.555, 7.495, 7.487, 7.441, 7.328, 7.324~
## $ GDP.per.Capita  <dbl> 1.305, 1.456, 1.351, 1.343, 1.420, 1.361, 1.330, 1.268~
## $ Social.Support  <dbl> 1.592, 1.582, 1.590, 1.644, 1.549, 1.488, 1.532, 1.601~
## $ Life.Expectancy <dbl> 0.874, 0.861, 0.868, 0.914, 0.927, 0.878, 0.896, 0.876~
## $ Freedom         <dbl> 0.681, 0.686, 0.683, 0.677, 0.660, 0.638, 0.653, 0.669~
## $ Generosity      <dbl> 0.202, 0.286, 0.284, 0.353, 0.256, 0.333, 0.321, 0.365~
## $ Corruption      <chr> "0.393", "0.340", "0.408", "0.138", "0.357", "0.295", ~
## [1] "A sample of data for year 2019 :"
## Rows: 156
## Columns: 9
## $ Overall.rank    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ Country         <chr> "Finland", "Denmark", "Norway", "Iceland", "Netherland~
## $ Happiness.Score <dbl> 7.769, 7.600, 7.554, 7.494, 7.488, 7.480, 7.343, 7.307~
## $ GDP.per.Capita  <dbl> 1.340, 1.383, 1.488, 1.380, 1.396, 1.452, 1.387, 1.303~
## $ Social.Support  <dbl> 1.587, 1.573, 1.582, 1.624, 1.522, 1.526, 1.487, 1.557~
## $ Life.Expectancy <dbl> 0.986, 0.996, 1.028, 1.026, 0.999, 1.052, 1.009, 1.026~
## $ Freedom         <dbl> 0.596, 0.592, 0.603, 0.591, 0.557, 0.572, 0.574, 0.585~
## $ Generosity      <dbl> 0.153, 0.252, 0.271, 0.354, 0.322, 0.263, 0.267, 0.330~
## $ Corruption      <dbl> 0.393, 0.410, 0.341, 0.118, 0.298, 0.343, 0.373, 0.380~
## [1] "A sample of data for year 2020 :"
## Rows: 153
## Columns: 20
## $ Country                                    <chr> "Finland", "Denmark", "Swit~
## $ Regional.indicator                         <chr> "Western Europe", "Western ~
## $ Happiness.Score                            <dbl> 7.8087, 7.6456, 7.5599, 7.5~
## $ Standard.error.of.ladder.score             <dbl> 0.03115630, 0.03349229, 0.0~
## $ upperwhisker                               <dbl> 7.869766, 7.711245, 7.62852~
## $ lowerwhisker                               <dbl> 7.747634, 7.579955, 7.49127~
## $ GDP.per.Capita                             <dbl> 10.639267, 10.774001, 10.97~
## $ Social.Support                             <dbl> 0.9543297, 0.9559908, 0.942~
## $ Life.Expectancy                            <dbl> 71.90083, 72.40250, 74.1024~
## $ Freedom                                    <dbl> 0.9491722, 0.9514443, 0.921~
## $ Generosity                                 <dbl> -0.059482019, 0.066201776, ~
## $ Corruption                                 <dbl> 0.1954446, 0.1684895, 0.303~
## $ Ladder.score.in.Dystopia                   <dbl> 1.972317, 1.972317, 1.97231~
## $ Explained.by..Log.GDP.per.capita           <dbl> 1.2851895, 1.3269485, 1.390~
## $ Explained.by..Social.support               <dbl> 1.499526, 1.503449, 1.47240~
## $ Explained.by..Healthy.life.expectancy      <dbl> 0.9612714, 0.9793326, 1.040~
## $ Explained.by..Freedom.to.make.life.choices <dbl> 0.6623167, 0.6650399, 0.628~
## $ Explained.by..Generosity                   <dbl> 0.15967044, 0.24279340, 0.2~
## $ Explained.by..Perceptions.of.corruption    <dbl> 0.47785726, 0.49526033, 0.4~
## $ Dystopia...residual                        <dbl> 2.762835, 2.432741, 2.35026~
## [1] "A sample of data for year 2021 :"
## Rows: 149
## Columns: 20
## $ Country                                    <chr> "Finland", "Denmark", "Swit~
## $ Regional.indicator                         <chr> "Western Europe", "Western ~
## $ Happiness.Score                            <dbl> 7.842, 7.620, 7.571, 7.554,~
## $ Standard.error.of.ladder.score             <dbl> 0.032, 0.035, 0.036, 0.059,~
## $ upperwhisker                               <dbl> 7.904, 7.687, 7.643, 7.670,~
## $ lowerwhisker                               <dbl> 7.780, 7.552, 7.500, 7.438,~
## $ GDP.per.Capita                             <dbl> 10.775, 10.933, 11.117, 10.~
## $ Social.Support                             <dbl> 0.954, 0.954, 0.942, 0.983,~
## $ Life.Expectancy                            <dbl> 72.000, 72.700, 74.400, 73.~
## $ Freedom                                    <dbl> 0.949, 0.946, 0.919, 0.955,~
## $ Generosity                                 <dbl> -0.098, 0.030, 0.025, 0.160~
## $ Corruption                                 <dbl> 0.186, 0.179, 0.292, 0.673,~
## $ Ladder.score.in.Dystopia                   <dbl> 2.43, 2.43, 2.43, 2.43, 2.4~
## $ Explained.by..Log.GDP.per.capita           <dbl> 1.446, 1.502, 1.566, 1.482,~
## $ Explained.by..Social.support               <dbl> 1.106, 1.108, 1.079, 1.172,~
## $ Explained.by..Healthy.life.expectancy      <dbl> 0.741, 0.763, 0.816, 0.772,~
## $ Explained.by..Freedom.to.make.life.choices <dbl> 0.691, 0.686, 0.653, 0.698,~
## $ Explained.by..Generosity                   <dbl> 0.124, 0.208, 0.204, 0.293,~
## $ Explained.by..Perceptions.of.corruption    <dbl> 0.481, 0.485, 0.413, 0.170,~
## $ Dystopia...residual                        <dbl> 3.253, 2.868, 2.839, 2.967,~
## [1] "A sample of data for year 2022 :"
## Rows: 146
## Columns: 12
## $ RANK                       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ~
## $ Country                    <chr> "Finland", "Denmark", "Iceland", "Switzerla~
## $ Happiness.Score            <dbl> 7.821, 7.636, 7.557, 7.512, 7.415, 7.404, 7~
## $ Whisker.high               <dbl> 7.886, 7.710, 7.651, 7.586, 7.471, 7.501, 7~
## $ Whisker.low                <dbl> 7.756, 7.563, 7.464, 7.437, 7.359, 7.307, 7~
## $ Dystopia..1.83....residual <dbl> 2.518, 2.226, 2.320, 2.153, 2.137, 2.042, 2~
## $ GDP.per.Capita             <dbl> 1.892, 1.953, 1.936, 2.026, 1.945, 2.209, 1~
## $ Social.Support             <dbl> 1.258, 1.243, 1.320, 1.226, 1.206, 1.155, 1~
## $ Life.Expectancy            <dbl> 0.775, 0.777, 0.803, 0.822, 0.787, 0.790, 0~
## $ Freedom                    <dbl> 0.736, 0.719, 0.718, 0.677, 0.651, 0.700, 0~
## $ Generosity                 <dbl> 0.109, 0.188, 0.270, 0.147, 0.271, 0.120, 0~
## $ Corruption                 <dbl> 0.534, 0.532, 0.191, 0.461, 0.419, 0.388, 0~
## [1] "A sample of data for year 2023 :"
## Rows: 137
## Columns: 19
## $ Country                                    <chr> "Finland", "Denmark", "Icel~
## $ Happiness.Score                            <dbl> 7.804, 7.586, 7.530, 7.473,~
## $ Standard.error.of.ladder.score             <dbl> 0.036, 0.041, 0.049, 0.032,~
## $ upperwhisker                               <dbl> 7.875, 7.667, 7.625, 7.535,~
## $ lowerwhisker                               <dbl> 7.733, 7.506, 7.434, 7.411,~
## $ GDP.per.Capita                             <dbl> 10.792, 10.962, 10.896, 10.~
## $ Social.Support                             <dbl> 0.969, 0.954, 0.983, 0.943,~
## $ Life.Expectancy                            <dbl> 71.150, 71.250, 72.050, 72.~
## $ Freedom                                    <dbl> 0.961, 0.934, 0.936, 0.809,~
## $ Generosity                                 <dbl> -0.019, 0.134, 0.211, -0.02~
## $ Corruption                                 <dbl> 0.182, 0.196, 0.668, 0.708,~
## $ Ladder.score.in.Dystopia                   <dbl> 1.778, 1.778, 1.778, 1.778,~
## $ Explained.by..Log.GDP.per.capita           <dbl> 1.888, 1.949, 1.926, 1.833,~
## $ Explained.by..Social.support               <dbl> 1.585, 1.548, 1.620, 1.521,~
## $ Explained.by..Healthy.life.expectancy      <dbl> 0.535, 0.537, 0.559, 0.577,~
## $ Explained.by..Freedom.to.make.life.choices <dbl> 0.772, 0.734, 0.738, 0.569,~
## $ Explained.by..Generosity                   <dbl> 0.126, 0.208, 0.250, 0.124,~
## $ Explained.by..Perceptions.of.corruption    <dbl> 0.535, 0.525, 0.187, 0.158,~
## $ Dystopia...residual                        <dbl> 2.363, 2.084, 2.250, 2.691,~
```

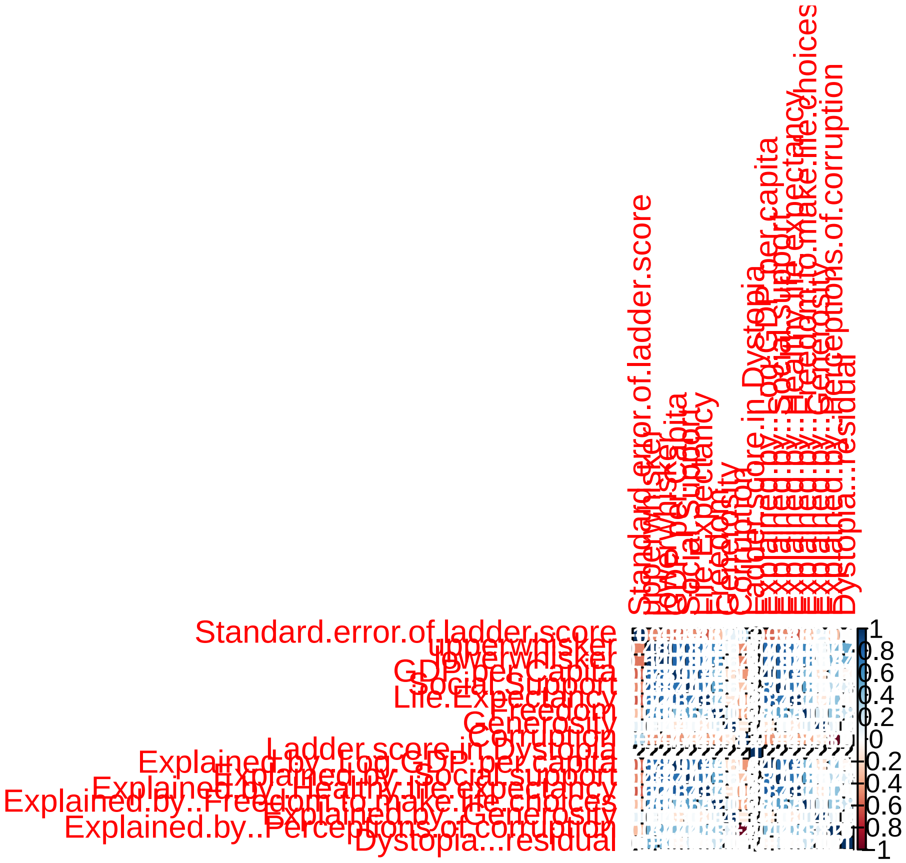
```
# remove null values from dataset
for(i in 1:TOTAL_YEARS){
  dummy_year = get(VARIABLE_NAMES[i])
  if(any(is.na(dummy_year))){
    print("Found NA values and removing them...")
    dummy_year = na.omit(dummy_year)
    assign(VARIABLE_NAMES[i],dummy_year)
  }
}
```

```
## [1] "Found NA values and removing them..."
```

Data Exploration

```
corrplot(corr = cor(data_2023[,-c(1,2)]),method = "color", outline = T,addCoef.col = "white", number.di
```

```
## Warning in cor(data_2023[, -c(1, 2)]): the standard deviation is zero
```



```
# lets first see the columns for each year data
col_2015 = c(colnames(data_2015))
col_2016 = colnames(data_2016)
col_2017 = colnames(data_2017)
```

```
col_2018 = colnames(data_2018)
col_2019 = colnames(data_2019)
col_2020 = colnames(data_2020)
col_2021 = colnames(data_2021)
col_2022 = colnames(data_2022)
col_2023 = colnames(data_2023)

# lets find which columns are consistent among all these years
all_col = list(col_2015,col_2016,col_2017,col_2018,col_2019,col_2020,col_2021,col_2022,col_2023)
common_values = Reduce(intersect, all_col)
print(common_values) # gives 0 : means the columns have been named differently in the dataset
```

```
## [1] "Country"         "Happiness.Score" "GDP.per.Capita"  "Social.Support"
## [5] "Life.Expectancy" "Freedom"         "Corruption"      "Generosity"
```

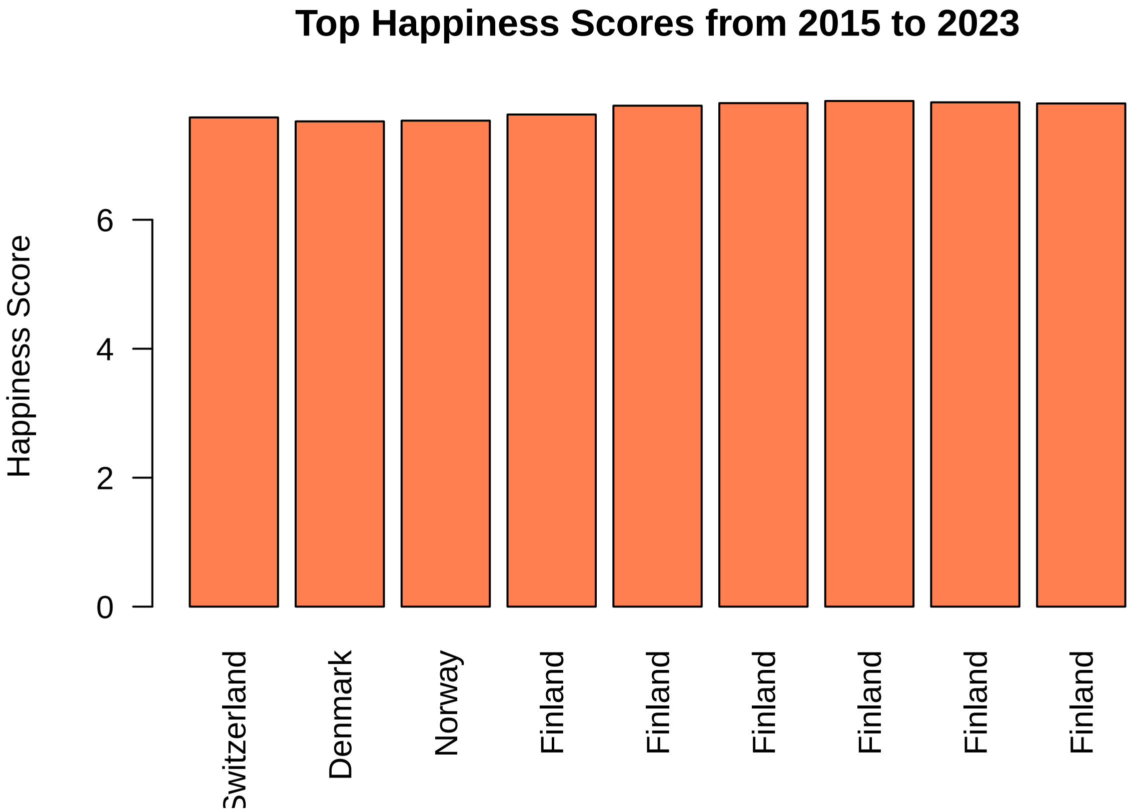
```
# After exploring the column names we can clearly see that the same variables are named differenty in t

# lets make the column names uniform
column_names = c("Country", "Happiness.Score", "GDP.per.Capita", "Social.Support", "Life.Expectancy","F

# print(common_values)
# gives all the values in column_names

# filter the values that are essential for the study (as mentioned on the world happiness report append

for(i in 1:TOTAL_YEARS){
  dummy_year = get(VARIABLE_NAMES[i])
  # select according to columns names
  dummy_year = dummy_year[column_names]
  # sort by column name
  dummy_year = dummy_year[,order(colnames(dummy_year))]
  assign(VARIABLE_NAMES[i], dummy_year )
  }


# lets see which countries has held the No.1 rank in happiness over the years
happy_countries = c()
happy_countries_score = c()

sad_countries = c()
sad_countries_score = c()

for(i in 1:TOTAL_YEARS){
  dummy_year = get(VARIABLE_NAMES[i])
  happy_countries[i] = dummy_year$Country[1]
  happy_countries_score[i] = dummy_year$Happiness.Score[1]
  sad_countries[i] = dummy_year$Country[nrow(dummy_year)]
  sad_countries_score[i] = dummy_year$Happiness.Score[nrow(dummy_year)]
}



print(happy_countries)
```

```
## [1] "Switzerland" "Denmark"     "Norway"      "Finland"     "Finland"
## [6] "Finland"     "Finland"     "Finland"     "Finland"
```

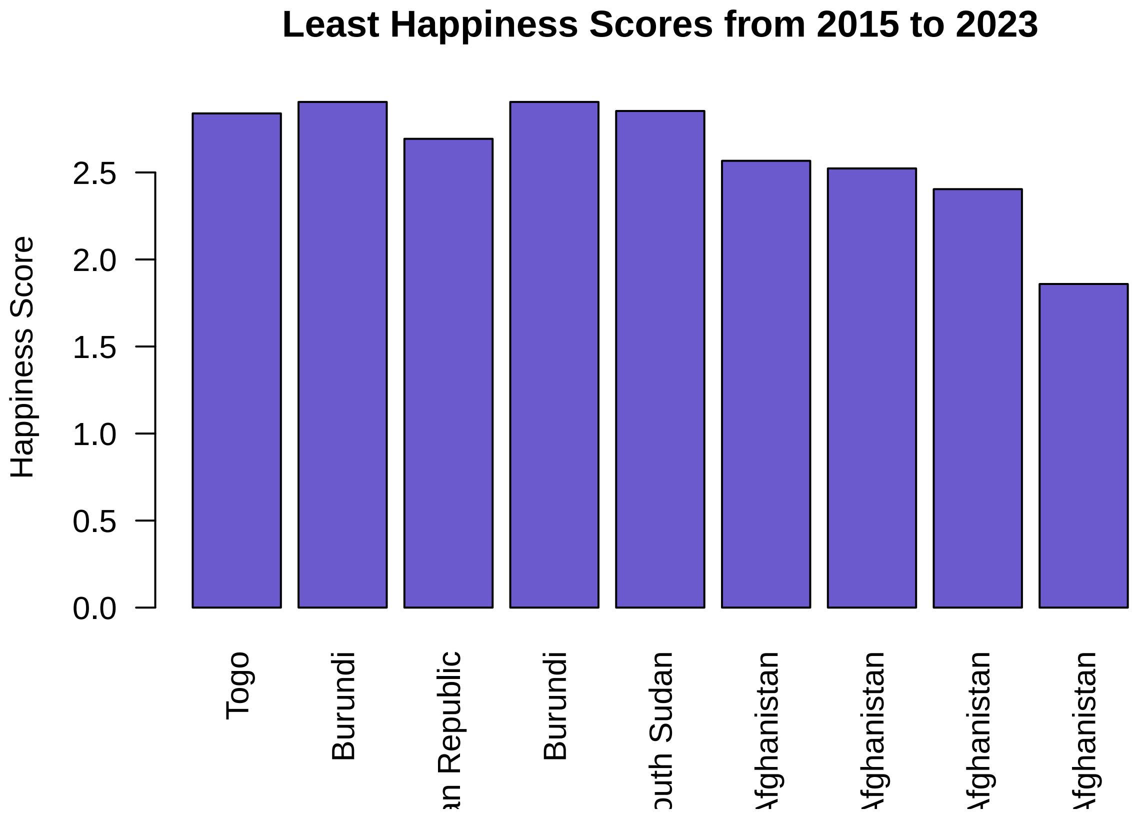
```
print(sad_countries)
```

```
## [1] "Togo"                     "Burundi"
## [3] "Central African Republic" "Burundi"
## [5] "South Sudan"              "Afghanistan"
## [7] "Afghanistan"              "Afghanistan"
## [9] "Afghanistan"
```

```
# see trend of happiness along the years
barplot(happy_countries_score,names.arg=happy_countries,main = "Top Happiness Scores from 2015 to 2023"
```



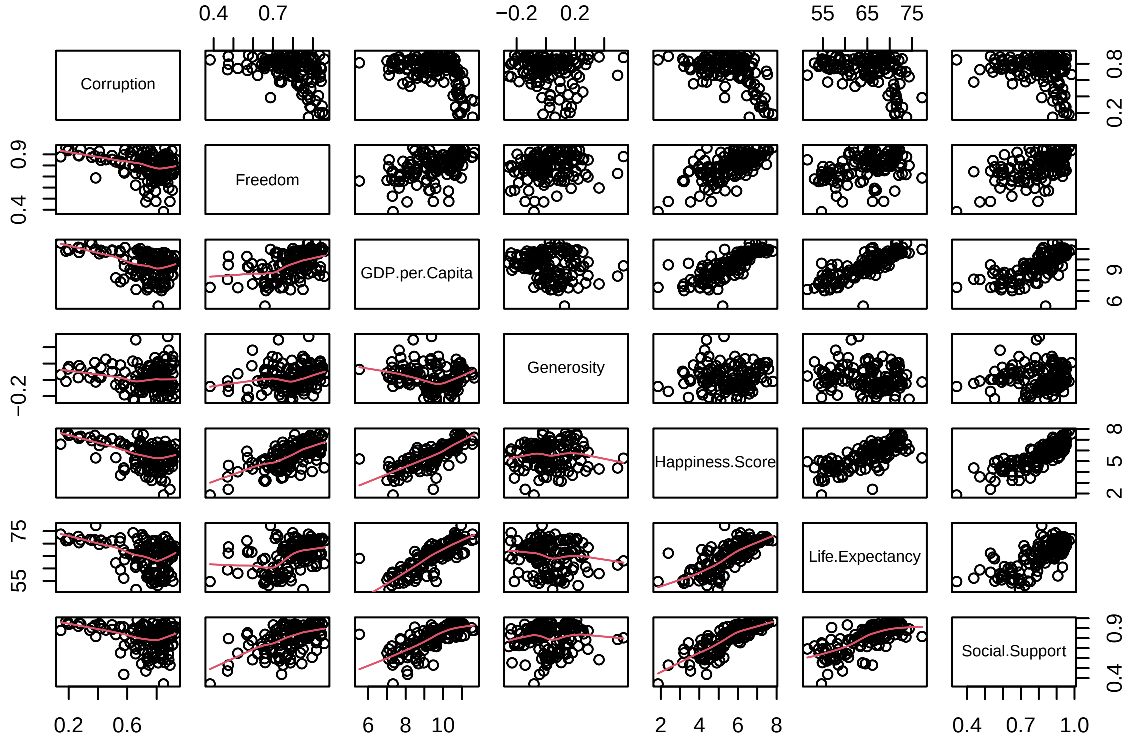
```
# see trend of sadness along the years
barplot(sad_countries_score,names.arg=sad_countries,main = "Least Happiness Scores from 2015 to 2023", 
```



```
# lets explore in details for a particular year
# summary of data
summary(data_2023)
```

```
##    Corruption       Country             Freedom       GDP.per.Capita
##  Min.   :0.1460   Length:136         Min.   :0.3820   Min.   : 5.527
##  1st Qu.:0.6660   Class :character   1st Qu.:0.7262   1st Qu.: 8.587
##  Median :0.7725   Mode  :character   Median :0.8010   Median : 9.575
##  Mean   :0.7246                      Mean   :0.7881   Mean   : 9.455
##  3rd Qu.:0.8460                      3rd Qu.:0.8748   3rd Qu.:10.540
##  Max.   :0.9290                      Max.   :0.9610   Max.   :11.660
##    Generosity        Happiness.Score Life.Expectancy Social.Support
##  Min.   :-0.25400   Min.   :1.859   Min.   :51.53   Min.   :0.3410
##  1st Qu.:-0.07100   1st Qu.:4.702   1st Qu.:60.65   1st Qu.:0.7210
##  Median : 0.00200   Median :5.694   Median :65.84   Median :0.8265
##  Mean   : 0.02357   Mean   :5.544   Mean   :64.97   Mean   :0.7986
##  3rd Qu.: 0.11750   3rd Qu.:6.343   3rd Qu.:69.41   3rd Qu.:0.8960
##  Max.   : 0.53100   Max.   :7.804   Max.   :77.28   Max.   :0.9830
```

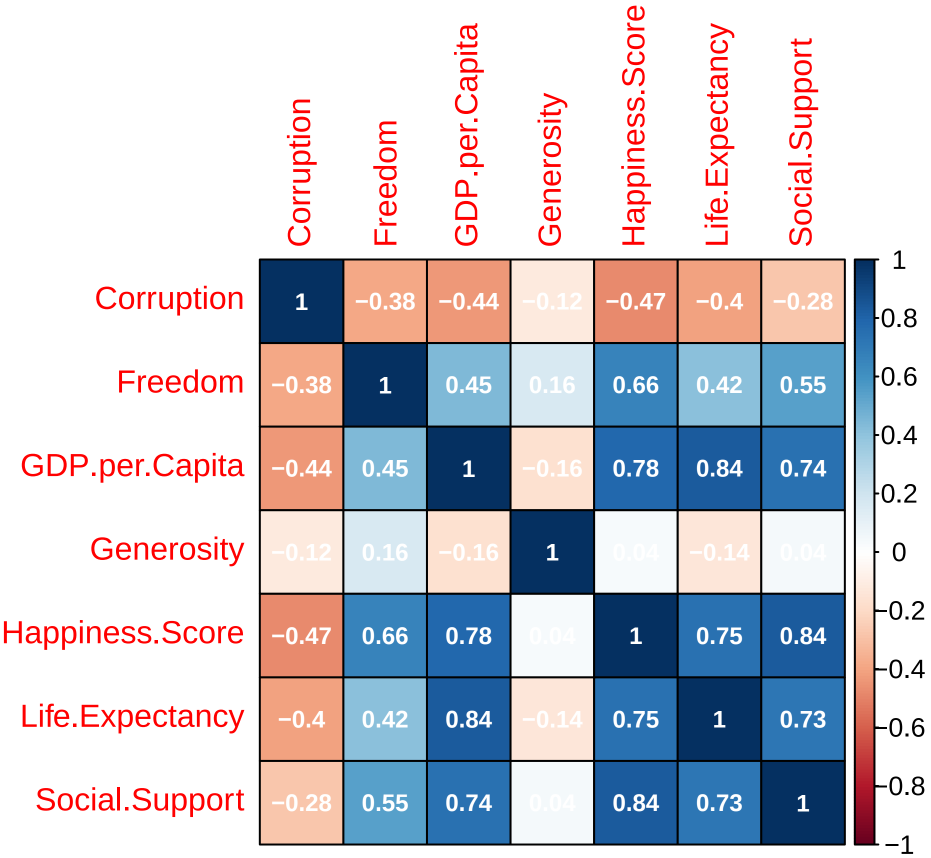
```
#see relationship between all variables in data
pairs(data_2023[,-2],lower.panel=panel.smooth)
```



```r
# see relation between happiness and different factors
cor(data_2023[,-2])
```

```
##                  Corruption    Freedom GDP.per.Capita  Generosity
## Corruption        1.0000000 -0.3814667     -0.4353726 -0.11830016
## Freedom          -0.3814667  1.0000000      0.4494623  0.16466639
## GDP.per.Capita   -0.4353726  0.4494623      1.0000000 -0.16229841
## Generosity       -0.1183002  0.1646664     -0.1622984  1.00000000
## Happiness.Score  -0.4705609  0.6619596      0.7838363  0.03982861
## Life.Expectancy  -0.4043965  0.4150920      0.8375331 -0.13503599
## Social.Support   -0.2752549  0.5463187      0.7417623  0.04052801
##                  Happiness.Score Life.Expectancy Social.Support
## Corruption           -0.47056085      -0.4043965    -0.27525487
## Freedom               0.66195958       0.4150920     0.54631867
## GDP.per.Capita        0.78383629       0.8375331     0.74176233
## Generosity            0.03982861      -0.1350360     0.04052801
## Happiness.Score       1.00000000       0.7469282     0.83805645
## Life.Expectancy       0.74692816       1.0000000     0.72552383
## Social.Support        0.83805645       0.7255238     1.00000000
```

```r
corrplot(corr = cor(data_2023[,-2]),method = "color", outline = T,addCoef.col = "white", number.digits
```



```
# top 5 happiest and unhappiest countries in 2023

# top 5 generous and ungenerous countries in 2023

# top 5 healthiest and unhealthiest countries in 2023

# top 5 richest and poorest countries in 2023

# top 5 free and restricted countries in 2023

# top 5 corrupted and uncorrupted countries in 2023
```

Data Clean-up for Regression

Train-test Split

Linear Regression Model

Lasso Regression Model

Ridge Regression Model

K-Nearest Neighbour Model

Comparison of Results

Accuracies

Models	Accuracy
Logistic Regression	79.38%
LDA	84.53%
LDA (0.3)	84.53%
Naive Bayes	83.50%
KNN	67.01%

Plots

k-fold Cross Validation

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

References:

remove this link in final publication * https://www.analyticsvidhya.com/blog/2022/01/different-types-of-regression-models/ * Dataset: https://medium.com/@TheSussex/is-happiness-free-bb9ef70d0bd5 https://www.kaggle.com/code/mshinde10/predicting-world-happiness https://www.kaggle.com/code/kagleo123/world-happiness-record-eda-ml-regression https://www.kaggle.com/datasets/unsdsn/world-happiness/code?datasetId=894&sortBy=dateRun https://www.kaggle.com/datasets/unsdsn/world-happiness/code?datasetId=894&sortBy=dateRun

Dataset for 2020: https://www.kaggle.com/datasets/londeen/world-happiness-report-2020

a https://worldhappiness.report/data/