Exploratory Data Analysis [github link]

1. Correlation setiap variables

Dependent Variable : Life.expectancy
Independet Variables : Year, Adult.Mortality, infant.deaths, Alcohol, percentage.expenditure, Hepatitis.B, Measles, BMI, under.five.deaths, Polio, Total.expenditure, Diphtheria, HIV.AIDS, GDP, Population, thinness..1.19.years, thinness.5.9.years, Income.composition.of.resources, Schooling

```
df = read.csv("/Users/shafaqonitatingmail.com/Documents/Semester 4/Data Mining and Visualization/Life Expectancy Data.csv")
```

```
head(df)
```

Description: df [6 × 22]

	Country <chr>	Year <int>	Status <chr>	Life.expectancy <dbl>	Adult.Mortality <int>	infant.deaths <int>	Alcohol <dbl>	percentage.expenditure <dbl>	Hepatitis.B <int>
1	Afghanistan	2015	Developing	65.0	263	62	0.01	71.279624	65
2	Afghanistan	2014	Developing	59.9	271	64	0.01	73.523582	62
3	Afghanistan	2013	Developing	59.9	268	66	0.01	73.219243	64
4	Afghanistan	2012	Developing	59.5	272	69	0.01	78.184215	67
5	Afghanistan	2011	Developing	59.2	275	71	0.01	7.097109	68
6	Afghanistan	2010	Developing	58.8	279	74	0.01	79.679367	66

6 rows | 1–10 of 22 columns

With the `head()` function above, we can identify columns that are numerical and categorical.

```
colSums(is.na(df))
```

```
 [1] "Country"                  "Year"                     "Status"
 [4] "Life.expectancy"          "Adult.Mortality"          "infant.deaths"
 [7] "Alcohol"                  "percentage.expenditure"   "Hepatitis.B"
[10] "Measles"                  "BMI"                      "under.five.deaths"
[13] "Polio"                    "Total.expenditure"        "Diphtheria"
[16] "HIV.AIDS"                 "GDP"                      "Population"
[19] "thinness..1.19.years"     "thinness.5.9.years"       "Income.composition.of.resources"
[22] "Schooling"
```

the code above is used to display the name of each column in the dataset. This dataset contains 22 columns of information about variables that affect life expectancy.

```
colSums(is.na(df))
```

```
                  Country                      Year                    Status           Life.expectancy
                        0                         0                         0                        10
           Adult.Mortality                 infant.deaths                   Alcohol   percentage.expenditure
                       10                         0                       194                         0
              Hepatitis.B                      Measles                       BMI          under.five.deaths
                      553                         0                        34                         0
                    Polio            Total.expenditure                 Diphtheria                   HIV.AIDS
                       19                       226                        19                         0
                      GDP                Population          thinness..1.19.years       thinness.5.9.years
                      448                       652                        34                        34
Income.composition.of.resources              Schooling
                      167                         0
```

with the function `colSums(is.na(df))` above, we can identify the number of null-values in each column of df. The `is.na()` function is used to check whether there are null values, then the `colSums()` function is used to sum the null-values in each df column.

Next, we will select columns in df that have numerical data type. This is needed because the `cor()` function will calculate the correlation between numerical variables so the variables to be calculated must be numerical.

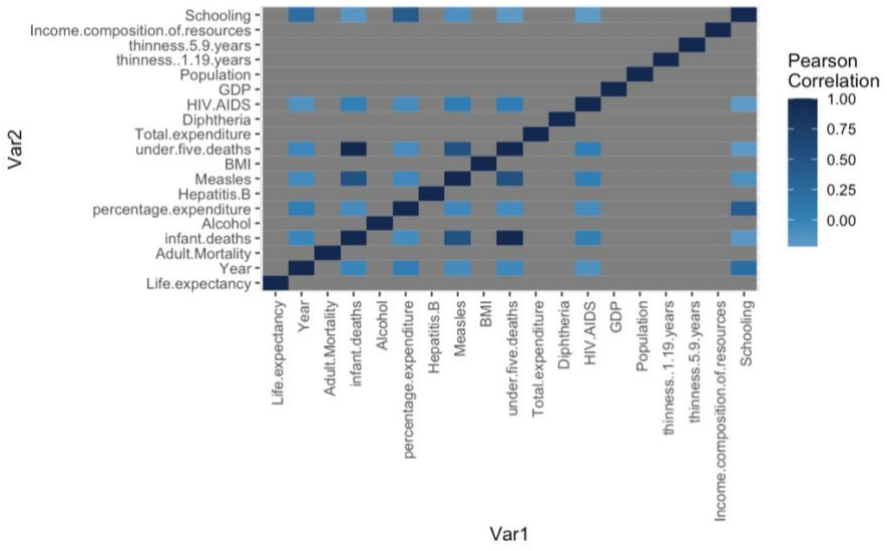
```
df_numerical = df[, c("Life.expectancy", "Year", "Adult.Mortality", "infant.deaths",
"Alcohol", "percentage.expenditure", "Hepatitis.B", "Measles", "BMI",
"under.five.deaths", "Total.expenditure", "Diphtheria", "HIV.AIDS", "GDP", "Population",
"thinness..1.19.years", "thinness.5.9.years", "Income.composition.of.resources",
"Schooling")]
cor(df_numerical)
```

```
                                 Life.expectancy       Year Adult.Mortality infant.deaths Alcohol percentage.expenditure Hepatitis.B
Life.expectancy                                1         NA              NA           NA      NA                      NA          NA
Year                                          NA 1.00000000              NA  -0.03741493      NA              0.03139998          NA
Adult.Mortality                               NA         NA               1           NA      NA                      NA          NA
infant.deaths                                 NA -0.03741493              NA  1.00000000      NA             -0.08561222          NA
Alcohol                                       NA         NA              NA           NA       1                      NA          NA
percentage.expenditure                        NA  0.0313998              NA  -0.08561222      NA              1.00000000          NA
Hepatitis.B                                   NA         NA              NA           NA      NA                      NA           1
Measles                                       NA -0.08249298              NA   0.50112834      NA             -0.05659568          NA
BMI                                           NA         NA              NA           NA      NA                      NA          NA
under.five.deaths                             NA -0.04293699              NA   0.99662888      NA             -0.08785231          NA
Total.expenditure                             NA         NA              NA           NA      NA                      NA          NA
Diphtheria                                    NA         NA              NA           NA      NA                      NA          NA
HIV.AIDS                                      NA -0.13974136              NA   0.02523132      NA             -0.09785682          NA
GDP                                           NA         NA              NA           NA      NA                      NA          NA
Population                                    NA         NA              NA           NA      NA                      NA          NA
thinness..1.19.years                          NA         NA              NA           NA      NA                      NA          NA
thinness.5.9.years                            NA         NA              NA           NA      NA                      NA          NA
Income.composition.of.resources               NA         NA              NA           NA      NA                      NA          NA
Schooling_                                    NA  0.2034712              NA  -0.19175731      NA              0.3881049          NA
                                 Measles BMI under.five.deaths Total.expenditure Diphtheria    HIV.AIDS GDP Population thinness..1.19.years
Life.expectancy                       NA  NA                NA                NA         NA          NA  NA         NA                    NA
Year                          -0.08249298  NA       -0.04293699                NA         NA -0.13974136  NA         NA                    NA
Adult.Mortality                       NA  NA                NA                NA         NA          NA  NA         NA                    NA
infant.deaths                 0.50112834  NA        0.99662888                NA         NA  0.02523132  NA         NA                    NA
Alcohol                               NA  NA                NA                NA         NA          NA  NA         NA                    NA
percentage.expenditure        -0.05659568  NA       -0.08785231                NA         NA -0.09785682  NA         NA                    NA
Hepatitis.B                           NA  NA                NA                NA         NA          NA  NA         NA                    NA
Measles                       1.00000000  NA        0.50780871                NA         NA  0.03089872  NA         NA                    NA
BMI                                   NA   1                NA                NA         NA          NA  NA         NA                    NA
under.five.deaths             0.50780871  NA        1.00000000                NA         NA  0.03806151  NA         NA                    NA
Total.expenditure                     NA  NA                NA                 1         NA          NA  NA         NA                    NA
Diphtheria                            NA  NA                NA                NA          1          NA  NA         NA                    NA
HIV.AIDS                      0.03089872  NA        0.03806151                NA         NA  1.00000000  NA         NA                    NA
GDP                                   NA  NA                NA                NA         NA          NA   1         NA                    NA
Population                            NA  NA                NA                NA         NA          NA  NA          1                    NA
thinness..1.19.years                  NA  NA                NA                NA         NA          NA  NA         NA                     1
thinness.5.9.years                    NA  NA                NA                NA         NA          NA  NA         NA                    NA
Income.composition.of.resources       NA  NA                NA                NA         NA          NA  NA         NA                    NA
Schooling                     -0.12260854  NA       -0.20711142                NA         NA -0.21861975  NA         NA                    NA
                                 thinness.5.9.years Income.composition.of.resources  Schooling
Life.expectancy                                  NA                              NA         NA
Year                                             NA                              NA  0.2034712
Adult.Mortality                                  NA                              NA         NA
infant.deaths                                    NA                              NA -0.1917573
Alcohol                                          NA                              NA         NA
percentage.expenditure                           NA                              NA  0.3881050
Hepatitis.B                                      NA                              NA         NA
Measles                                          NA                              NA -0.1226085
BMI                                              NA                              NA         NA
under.five.deaths                                NA                              NA -0.2071114
Total.expenditure                                NA                              NA         NA
Diphtheria                                       NA                              NA         NA
HIV.AIDS                                         NA                              NA -0.2186198
GDP                                              NA                              NA         NA
Population                                       NA                              NA         NA
thinness..1.19.years                             NA                              NA         NA
thinness.5.9.years                                1                              NA         NA
Income.composition.of.resources                  NA                               1         NA
Schooling                                        NA                              NA  1.0000000
```

It is found that there are a lot of NAs which means there are a lot of missing values or null-values between the columns. This will certainly affect the data correlation plot.

```
library(reshape2)
library(ggplot2)

corr_melted <- melt(cor(df_numerical))
ggplot(corr_melted, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "#DEFCF9", mid = "#3282B8", high = "#142850",
                       name = "Pearson\nCorrelation") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.7, hjust = 1))
```



The plot shows that the correlation is very poor because there are a lot of null-values depicted in grey.

To fix this problem, we need to impute null-values, which is the process of filling in the null-values for each data that has no value and replacing them with the mean, median, mode or other methods. This needs to be done so that the data can be analyzed better.

In this case, we will impute the null values with the mean value in each column. The first step is we will select the numerical variable by using the `sapply()` and `is.numeric()` functions which will be stored in a new data frame called `df_num`. Next, we will calculate the average by ignoring null-values using `na.rm = TRUE` in the `colMeans()` function.

```
df_num <- df[, sapply(df, is.numeric)]
colMeans(df_num, na.rm = TRUE)
```

Year	Life.expectancy	Adult.Mortality	infant.deaths	Alcohol
2.007519e+03	6.922493e+01	1.647964e+02	3.030395e+01	4.602861e+00
percentage.expenditure	Hepatitis.B	Measles	BMI	under.five.deaths
7.382513e+02	8.094046e+01	2.419592e+03	3.832125e+01	4.203574e+01
Polio	Total.expenditure	Diphtheria	HIV.AIDS	GDP
8.255019e+01	5.938190e+00	8.232408e+01	1.742103e+00	7.483158e+03
Population	thinness..1.19.years	thinness.5.9.years	Income.composition.of.resources	Schooling
1.275338e+07	4.839704e+00	4.870317e+00	6.275511e-01	1.199279e+01

The next step is that we will iterate over df_num to impute null-values with the mean or average value of each column. If we get columns in df_num that have null-values, we will impute values to fill the null-values.

```
for(i in names(df_num)) {
  df_num[, i][is.na(df_num[, i])] = mean(df_num[, i], na.rm=TRUE)
}
colSums(is.na(df_num))
```

Year	Life.expectancy	Adult.Mortality	infant.deaths	Alcohol
0	0	0	0	0
percentage.expenditure	Hepatitis.B	Measles	BMI	under.five.deaths
0	0	0	0	0
Polio	Total.expenditure	Diphtheria	HIV.AIDS	GDP
0	0	0	0	0
Population	thinness..1.19.years	thinness.5.9.years	Income.composition.of.resources	Schooling
0	0	0	0	0

By running imputation on the null-values, it is found that there are no longer any columns that have null-values according to the output above.

Next, we will perform calculations to find the correlation value for each variable as in the beginning using the `cor()` function.

```
print(cor(df_num))
```

	Year	Life.expectancy	Adult.Mortality	infant.deaths	Alcohol	percentage.expenditure	Hepatitis.B	Measles	BMI
Year	1.00000000	0.1696226	-0.07886077	-0.03741493	-0.04816759	0.03139998	0.08939827	-0.08249298	0.1083267
Life.expectancy	0.16962259	1.0000000	-0.69635931	-0.19653500	0.39159834	0.38179117	0.20377144	-0.15757382	0.5592553
Adult.Mortality	-0.07886077	-0.6963593	1.00000000	0.07874713	-0.19040781	-0.24281353	-0.13859091	0.03117404	-0.3814494
infant.deaths	-0.03741493	-0.1965350	0.07874713	1.00000000	-0.11381227	-0.08561222	-0.17878339	0.50112834	-0.2272200
Alcohol	-0.04816759	0.3915983	-0.19040781	-0.11381227	1.00000000	0.33963429	0.07544715	-0.05105499	0.3180703
percentage.expenditure	0.03139998	0.3817912	-0.24281353	-0.08561222	0.33963429	1.00000000	0.01167932	-0.05659568	0.2285372
Hepatitis.B	0.08939827	0.2037714	-0.13859091	-0.17878339	0.07544715	0.01167932	1.00000000	-0.09031694	0.1349285
Measles	-0.08249298	-0.1575738	0.03117404	0.50112834	-0.05105499	-0.05659568	-0.09031694	1.00000000	-0.1759253
BMI	0.10832670	0.5592553	-0.38144941	-0.22721997	0.31807030	0.22853723	0.13492851	-0.17592529	1.0000000
under.five.deaths	-0.04293699	-0.2225030	0.09413509	0.99662888	-0.11077713	-0.08785231	-0.18441262	0.50780871	-0.2375859
Polio	0.09381958	0.4615738	-0.27269358	-0.17067376	0.21374404	0.14720343	0.40851923	-0.13614628	0.2821559
Total.expenditure	0.08186014	0.2079806	-0.11087454	-0.12656412	0.29489812	0.17341415	0.05008430	-0.10456872	0.2318144
Diphtheria	0.13385337	0.4754184	-0.27301389	-0.17515631	0.21524194	0.14356978	0.49995767	-0.14186137	0.2810588
HIV.AIDS	-0.13974136	-0.5564568	0.52372692	0.02523132	-0.04864971	-0.09785682	-0.10240544	0.03089872	-0.2435476
GDP	0.09335084	0.4304930	-0.27705292	-0.10710904	0.31859116	0.88814032	0.06231758	-0.06805959	0.2766447
Population	0.01495063	-0.0196377	-0.01250145	0.54852167	-0.03076466	-0.02464822	-0.10981088	0.23624988	-0.0632376
thinness..1.19.years	-0.04759213	-0.4721619	0.29986267	0.46559015	-0.41694557	-0.25119000	-0.10514361	0.22474217	-0.5320247
thinness.5.9.years	-0.05062670	-0.4666292	0.30536641	0.47122795	-0.40588068	-0.25272486	-0.10833424	0.22100716	-0.5389106
Income.composition.of.resources	0.23633317	0.6924828	-0.44006205	-0.14366278	0.41609923	0.38037355	0.15099204	-0.11576407	0.4798374
Schooling	0.20347119	0.7150663	-0.43510845	-0.19175731	0.49754628	0.38810498	0.17175483	-0.12260854	0.5081055

	under.five.deaths	Polio	Total.expenditure	Diphtheria	HIV.AIDS	GDP	Population	thinness..1.19.years
Year	-0.04293699	0.09381958	0.08186014	0.13385337	-0.13974136	0.09335084	0.014950626	-0.04759213
Life.expectancy	-0.22250302	0.46157378	0.20798062	0.47541838	-0.55645682	0.43049302	-0.019637702	-0.47216188
Adult.Mortality	0.09413509	-0.27269358	-0.11087454	-0.27301389	0.52372692	-0.27705292	-0.012501453	0.29986267
infant.deaths	0.99662888	-0.17067376	-0.12656412	-0.17515631	0.02523132	-0.10710904	0.548521669	0.46559015
Alcohol	-0.11077713	0.21374404	0.29489812	0.21524194	-0.04864971	0.31859116	-0.030764657	-0.41694557
percentage.expenditure	-0.08785231	0.14720343	0.17341415	0.14356978	-0.09785682	0.88814032	-0.024648218	-0.25119000
Hepatitis.B	-0.18441262	0.40851923	0.05008430	0.49995767	-0.10240544	0.06231758	-0.109810878	-0.10514361
Measles	0.50780871	-0.13614628	-0.10456872	-0.14186137	0.03089872	-0.06805959	0.236249877	0.22474217
BMI	-0.23758588	0.28215592	0.23181440	0.28105881	-0.24354758	0.27664474	-0.063237604	-0.53202475
under.five.deaths	1.00000000	-0.18870311	-0.12826939	-0.19565055	0.03806151	-0.11064032	0.535864022	0.46762640
Polio	-0.18870311	1.00000000	0.13012949	0.67355332	-0.15948864	0.19398031	-0.034881799	-0.21993756
Total.expenditure	-0.12826939	0.13012949	1.00000000	0.14559660	-0.00138272	0.12146709	-0.066698201	-0.26872376
Diphtheria	-0.19565055	0.67355332	0.14559660	1.00000000	-0.16478684	0.18279456	-0.025457689	-0.22781986
HIV.AIDS	0.03806151	-0.15948864	-0.00138272	-0.16478684	1.00000000	-0.13451379	-0.027318431	0.20392213
GDP	-0.11064032	0.19398031	0.12146709	0.18279456	-0.13451379	1.00000000	-0.025611731	-0.26774463
Population	0.53586402	-0.03488180	-0.06669820	-0.02545769	-0.02731843	-0.02561173	1.000000000	0.23611740
thinness..1.19.years	0.46762640	-0.21993756	-0.26872376	-0.22781986	0.20392213	-0.26774463	0.236117395	1.00000000
thinness.5.9.years	0.47209862	-0.22071010	-0.27524000	-0.22110531	0.20713956	-0.27239955	0.233940899	0.93910199
Income.composition.of.resources	-0.16153341	0.35539764	0.14909473	0.37172915	-0.24745353	0.44031708	-0.007951445	-0.40666173
Schooling	-0.20711142	0.38583163	0.21831013	0.38994380	-0.21861975	0.42948916	-0.029464903	-0.44614009

	thinness.5.9.years	Income.composition.of.resources	Schooling
Year	-0.0506267	0.236333172	0.2034712
Life.expectancy	-0.4666292	0.692482805	0.7150663
Adult.Mortality	0.3053664	-0.440062048	-0.4351085
infant.deaths	0.4712279	-0.143662780	-0.1917573
Alcohol	-0.4058807	0.416099225	0.4975463
percentage.expenditure	-0.2527249	0.380373551	0.3881050
Hepatitis.B	-0.1083342	0.150992044	0.1717548
Measles	0.2210072	-0.115764074	-0.1226085
BMI	-0.5389106	0.479837369	0.5081055
under.five.deaths	0.4720986	-0.161533413	-0.2071114
Polio	-0.2207101	0.355397638	0.3858316
Total.expenditure	-0.2752400	0.149094726	0.2183101
Diphtheria	-0.2211053	0.371729147	0.3899438
HIV.AIDS	0.2071396	-0.247453534	-0.2186198
GDP	-0.2723995	0.440317075	0.4294892
Population	0.2339409	-0.007951445	-0.0294649
thinness..1.19.years	0.9391020	-0.406661733	-0.4461401
thinness.5.9.years	1.0000000	-0.395778501	-0.4357772
Income.composition.of.resources	-0.3957785	1.000000000	0.7962071
Schooling	-0.4357772	0.796207053	1.0000000

By imputing the values in the null-values, the resulting correlation values no longer have null-values according to the output above. This will certainly help in plotting the correlation and in conducting further analysis of the data.

Then we will do melting or convert the correlation matrix format into a long format using the reshape2 library so that it will produce a data frame with three columns, namely Var1, Var2, and value, where Var1 and Var2 represent the column names in the converted correlation matrix and value is the correlation value between Var1 and Var2. After that, we can confirm by using the `head()` function.

```
library(dplyr)

corr_melted <- melt(cor(df_num))
corr_melted_sorted <- corr_melted %>% arrange(desc(value))
head(corr_melted_sorted, 40)
```

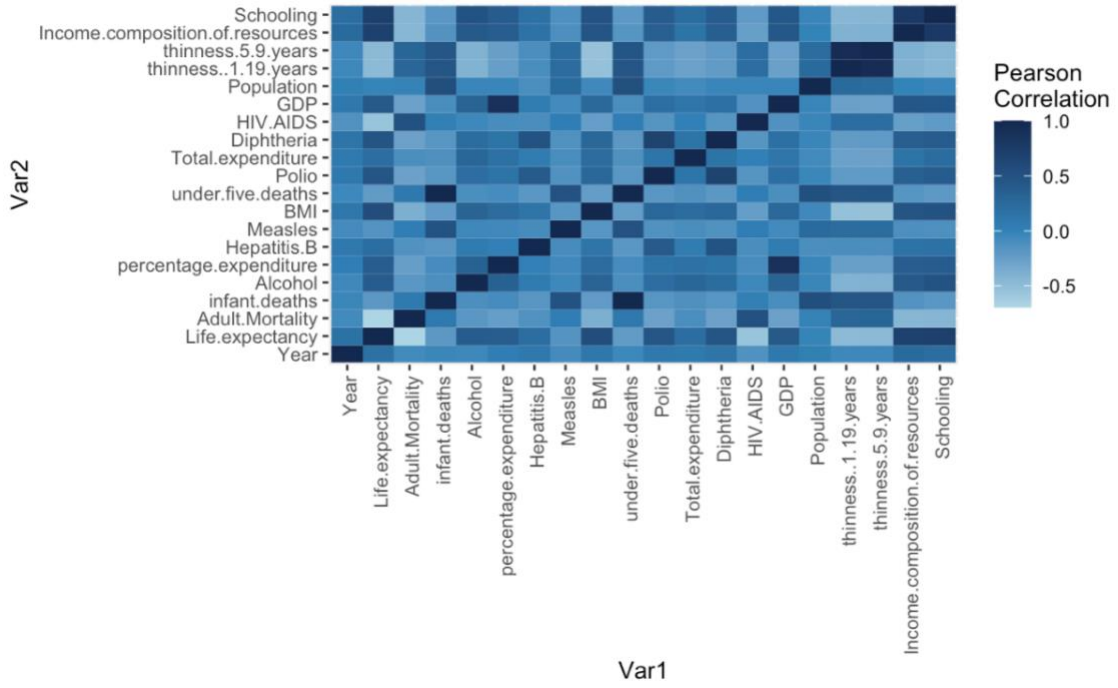
	Var1 <fctr>	Var2 <fctr>	value <dbl>
1	Year	Year	1.0000000
2	Life.expectancy	Life.expectancy	1.0000000
3	Adult.Mortality	Adult.Mortality	1.0000000
4	infant.deaths	infant.deaths	1.0000000
5	Alcohol	Alcohol	1.0000000
6	percentage.expenditure	percentage.expenditure	1.0000000
7	Hepatitis.B	Hepatitis.B	1.0000000
8	Measles	Measles	1.0000000
9	BMI	BMI	1.0000000
10	under.five.deaths	under.five.deaths	1.0000000
11	Polio	Polio	1.0000000
12	Total.expenditure	Total.expenditure	1.0000000
13	Diphtheria	Diphtheria	1.0000000
14	HIV.AIDS	HIV.AIDS	1.0000000
15	GDP	GDP	1.0000000
16	Population	Population	1.0000000
17	thinness..1.19.years	thinness..1.19.years	1.0000000
18	thinness.5.9.years	thinness.5.9.years	1.0000000
19	Income.composition.of.resources	Income.composition.of.resources	1.0000000
20	Schooling	Schooling	1.0000000

21	under.five.deaths	infant.deaths	0.9966289
22	infant.deaths	under.five.deaths	0.9966289
23	thinness.5.9.years	thinness..1.19.years	0.9391020
24	thinness..1.19.years	thinness.5.9.years	0.9391020
25	GDP	percentage.expenditure	0.8881403
26	percentage.expenditure	GDP	0.8881403
27	Schooling	Income.composition.of.resources	0.7962071
28	Income.composition.of.resources	Schooling	0.7962071
29	Schooling	Life.expectancy	0.7150663
30	Life.expectancy	Schooling	0.7150663
31	Income.composition.of.resources	Life.expectancy	0.6924828
32	Life.expectancy	Income.composition.of.resources	0.6924828
33	Diphtheria	Polio	0.6735533
34	Polio	Diphtheria	0.6735533
35	BMI	Life.expectancy	0.5592553
36	Life.expectancy	BMI	0.5592553
37	Population	infant.deaths	0.5485217
38	infant.deaths	Population	0.5485217
39	Population	under.five.deaths	0.5358640
40	under.five.deaths	Population	0.5358640

The output shows the 40 highest correlation values, but we only focus on the 21st row because the previous row is the correlation between the variable and the variable itself. The top three shows that there is a high correlation in under.five.deaths with infant.deaths, in thinness.5.9.years with thinness..1.19.years, and in GDP with percentage.expenditure. This output also shows a high correlation in our dependet variable (Life.Expectancy) with the variables Schooling, Income.composition.of.resources, BMI, and GDP.

Next, we will prove it by plotting the correlation value of each two numerical variables in our data using ggplot.

```
library(ggplot2)
ggplot(corr_melted, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "#DEFCF9", mid = "#3282B8", high = "#142850",
                       name = "Pearson\nCorrelation") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.7, hjust = 1))
```



Based on the plot, we can conclude that there is a high correlation between the dependent variables (Life.expectancy) with Schooling, Income.composition.of.resources, BMI, and GDP and this is in accordance with the data that we have sorted above.

In addition to the resulting correlations to the dependent variables, it was also found that there is:
- High correlation between under.five deaths and infant.deaths
- High correlation between under.five deaths and Measles
- High correlation between infant.deaths and Measles
- High correlation between Schooling and percentage.expenditure
- High correlation between Schooling and Year
- etc.

2. Statistical Testing

Now, we will use several statistical tests to answer our research questions or hypothesis, which in this section we have 4 research questions. Using the power of statistical analysis, we will gain deeper insights and meaningful conclusions from the data at hand.

1) How is the distribution of life expectancy in this dataset divided into specific quantiles?

In accordance with the question above, we will find the distribution of the variable life.expectancy in our dataset divided into specific quantiles. With the code below, we will calculate the quantile of the life.expectancy variable

```
df_stat <- df_num
quantiles <- quantile(df_stat$Life.expectancy, probs = seq(0, 1, 0.25))
quantiles
```

```
  0%  25%  50%  75% 100% 
36.3 63.2 72.0 75.6 89.0 
```

This output shows that at the 25% quantile, the life expectancy value is 63.2, which means that 25% of the population in the dataset has a life expectancy below this value. Then at the 50% quantile, the life expectancy value is 72.0, which means that 50% of the population in the dataset has a life expectancy below this value. Then at the 75% quantile, the life expectancy value is 75.6, which means that 75% of the population in the dataset has a life expectancy below this value, and so on.

2) Is the minimum value of the life expectancy variable is in a normal distribution with the mean and standard deviation of the variable?

In accordance with the question above, we will find out whether the minimum value of the life.expectancy variable is in the normal distribution or not. With the code below, we will determine the extent to which the minimum value of the life expectancy variable falls within the normal distribution by using the mean and standard deviation of the variable itself. The result of this code is pMin, which represents the probability of the minimum value being in a normal distribution, where the higher pMin, the higher the probability that the minimum value of the variable comes from a normal distribution.

```
minLE <- min(df_stat$Life.expectancy)
maxLE <- max(df_stat$Life.expectancy)
meanLE <- mean(df_stat$Life.expectancy)
sdLE <- sd(df_stat$Life.expectancy)
pMin <- pnorm( q = minLE, mean = meanLE, sd = sdLE)
pMin
```

```
[1] 0.0002670969
```

And the answer to this research question is no because we can see that the pMin value is so low that it indicates that the probability that the minimum value of the variable does not come from a normal distribution.

3) What are the quantile values of a normal distribution with specified mean and standard deviation, for ten different probabilities, ranging from 0 to 1 with intervals of 0.1?

In accordance with the question above, we will find out whether the minimum value of the life.expectancy variable is in the normal distribution or not. With the code below, we will find out how the quantile values produced by the normal distribution are

```
qnorm(mean = meanLE, sd = sdLE, p = seq(0, 1, 0.1))
```

```
 [1]     -Inf 57.04040 61.22310 64.23912 66.81620 69.22493 71.63366 74.21074 77.22676 81.40946      Inf
```

The output shows that at the 10% Quantile the life expectancy value is 57.04040, which is about 10% of the data below this value. At the 20% Quantile the life expectancy value is 61.22310, which is about 20% of the data below this value. At the 30% quantile the life expectancy value is 64.23912, which is about 30% of the data below this value. At the 40% Quantile the life expectancy value is 66.81620, which is about 40% of the data below this value. At the 50% Quantile, the life expectancy value is 69.22493, which is about 50% of the data below this value, etc. Where -inf indicates the lowest value that can be obtained in a normal distribution and inf otherwise.

4) Is there a significant difference between the average life expectancy in this dataset and the specified population mean value (70 years)?

Furthermore, according to the question above we will use the previous code to evaluate whether there is a significant difference between the average life expectancy in the dataset and the specified population mean value (69.22493 years) using t-test.

In this case, the null hypothesis (H0) is that the average life expectancy in the dataset is equal to 69.22493, while the alternative hypothesis (H1) is that the average life expectancy in the dataset is significantly different from 69.22493.

```
t.test(df_stat$Life.expectancy, mu = 69.22493, alternative = "two.sided")
```

```
        One Sample t-test

data:  df_stat$Life.expectancy
t = 9.6575e-06, df = 2937, p-value = 1
alternative hypothesis: true mean is not equal to 69.22493
95 percent confidence interval:
 68.88100 69.56886
sample estimates:
mean of x
 69.22493
```

The above output shows that there is not enough statistical evidence to support a significant difference between the average life expectancy in the dataset and the specified mean value of 69.22493. This is indicated by the p-value of 1, which indicates that there is not enough statistical evidence to conclude that there is a significant difference between the average life expectancy in the dataset and the specified mean value. Thus, we can conclude that with 95% confidence the average life expectancy is between 68.88100 and 69.56886 so we can accept the null hypothesis (H0).

3. Predictive Modelling using Random Forest

Step 1: Normalize the data to have the same range of values

Normalization is necessary in order to convert the values in the dataset into a uniform scale. In this normalization we will use the min-max scaling method. First, we need to define the Max and Min.

In this code, we will define Max by passing the maximum value of each column of dataset df_num represented by number 2 and Min by passing the minimum value of each column of dataset df_num represented by number 2.

```
Max <- apply(df_num, 2, max) #counts the maximum value of each column in the dataset
Min <- apply(df_num, 2, min) #counts the minimum value of each column in the dataset
```

The following is the output of the maximum value of each column in the dataset df_num

```
Max
```

Year	Life.expectancy	Adult.Mortality	infant.deaths
2.015000e+03	8.900000e+01	7.230000e+02	1.800000e+03
Alcohol	percentage.expenditure	Hepatitis.B	Measles
1.787000e+01	1.947991e+04	9.900000e+01	2.121830e+05
BMI	under.five.deaths	Polio	Total.expenditure
8.730000e+01	2.500000e+03	9.900000e+01	1.760000e+01
Diphtheria	HIV.AIDS	GDP	Population
9.900000e+01	5.060000e+01	1.191727e+05	1.293859e+09
thinness..1.19.years	thinness.5.9.years	Income.composition.of.resources	Schooling
2.770000e+01	2.860000e+01	9.480000e-01	2.070000e+01

The following is the output of the minimum value of each column in the dataset df_num

```
Min
```

Year	Life.expectancy	Adult.Mortality	infant.deaths
2000.00000	36.30000	1.00000	0.00000
Alcohol	percentage.expenditure	Hepatitis.B	Measles
0.01000	0.00000	1.00000	0.00000
BMI	under.five.deaths	Polio	Total.expenditure
1.00000	0.00000	3.00000	0.37000
Diphtheria	HIV.AIDS	GDP	Population
2.00000	0.10000	1.68135	34.00000
thinness..1.19.years	thinness.5.9.years	Income.composition.of.resources	Schooling
0.10000	0.10000	0.00000	0.00000

Now, we will normalize the data with the min-max scaling method and store it as a data frame. The scale() function itself is used to normalize the data in df_num by shifting and adjusting the range of variable values. The center parameter will indicate that the minimum value of each variable will be used as the center value. Then, the scale parameter will determine the value range of each variable after normalization.

```
scaled <- as.data.frame(scale(df_num, center = Min, scale = Max - Min))
head(scaled)
```

Description: df [6 × 20]

	Year <dbl>	Life.expectancy <dbl>	Adult.Mortality <dbl>	infant.deaths <dbl>	Alcohol <dbl>	percentage.expenditure <dbl>	Hepatitis.B <dbl>
1	1.0000000	0.5445920	0.3628809	0.03444444	0	0.0036591349	0.6530612
2	0.9333333	0.4478178	0.3739612	0.03555556	0	0.0037743283	0.6224490
3	0.8666667	0.4478178	0.3698061	0.03666667	0	0.0037587051	0.6428571
4	0.8000000	0.4402277	0.3753463	0.03833333	0	0.0040135816	0.6734694
5	0.7333333	0.4345351	0.3795014	0.03944444	0	0.0003643296	0.6836735
6	0.6666667	0.4269450	0.3850416	0.04111111	0	0.0040903352	0.6632653

6 rows | 1–8 of 20 columns

Step 2: Split the data into train and test

We will split df_num with a split ratio of 80%:20%, where the train dataset is 80% of the total data in df_num and the test dataset is the remaining 20% of the df_num data.

```
library(caTools)
set.seed(42)
```

```
split = sample.split(scaled$Life.expectancy, SplitRatio = 0.80)

train <- subset(scaled, split == TRUE)
test <- subset(scaled, split == FALSE)
```

Step 3: Create a formula for what columns will be fit to the model so that they are not written one by one. Where in this formula the dependent variable is Life.Expectancy and the independent variables are variables other than Life.Expectancy or we can see the output below:

```
f <- as.formula(paste("Life.expectancy ~", paste(n[!n %in% "Life.expectancy"], collapse
= " + ")))
f
```

```
Life.expectancy ~ Year + Adult.Mortality + infant.deaths + Alcohol +
    percentage.expenditure + Hepatitis.B + Measles + BMI + under.five.deaths +
    Polio + Total.expenditure + Diphtheria + HIV.AIDS + GDP +
    Population + thinness..1.19.years + thinness.5.9.years +
    Income.composition.of.resources + Schooling
```

Step 4: Predictive modeling using Random Forest

Loads the randomForest library into R and creates a random forest model using the randomForest() function, where the parameters are the formula above and the data to be trained.

```
library(randomForest)

modelRF <- randomForest(f,data=train)
print(modelRF)
```

```
Call:
 randomForest(formula = f, data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 0.001176477
                    % Var explained: 96.44
```

The output shows the result of regression modeling using the Random Forest algorithm with 500 trees and 6 variables tried on each split (node). The model produces a Mean of squared residuals value of 0.001176477 and % Var explained (percentage of variation explained by the model) of 96.44.

Importance() function will calculate the importance score of independent variables in our model, this score measures how often each variable is used in the formation of decisions by each variable.

```
# importance of each predictor
importance(modelRF)
```

```
                               IncNodePurity
Year                               0.5935961
Adult.Mortality                   15.9738233
infant.deaths                      1.3228608
Alcohol                            0.9236581
percentage.expenditure             0.5259169
Hepatitis.B                        0.2693687
Measles                            0.3550443
BMI                                2.7760678
under.five.deaths                  2.7341605
Polio                              1.2390567
Total.expenditure                  0.5262947
Diphtheria                         1.3340237
HIV.AIDS                          21.4801717
GDP                                0.7373768
Population                         0.3000999
thinness..1.19.years               1.3513500
thinness.5.9.years                 1.9439790
Income.composition.of.resources   15.1867229
Schooling                          7.4610601
```

From the output above we can see that importance() function returns an array containing the importance value of each feature in the random forest model. The importance value represents the relative contribution of each feature in making accurate predictions. If variables with high scores are not used in our model, the results will change significantly, so it is important to know which variables contribute the most, which in this case are HIV.AIDS, Adult.Mortality, and Income.composition.of.resources.

Then, we will predict our model using the test dataset and store it in the predictedRF variable. Then, we will combine the predictedRF and test$Life.expectancy columns into one data frame, where predictedRF contains the predicted value generated by the model, while test$Life.expectancy contains the actual value of the response variable. Then we will name it "Predicted Value" for the prediction and "Ground Truth" for the actual value.

```
#create a data frame to compare the predicted value with the ground truth
predictedRF <- predict(modelRF,test)
resultsRF <- cbind(predictedRF,test$Life.expectancy)

colnames(resultsRF) <- c('Predicted Value','Ground Truth')
resultsRF <- as.data.frame(resultsRF)
resultsRF
```

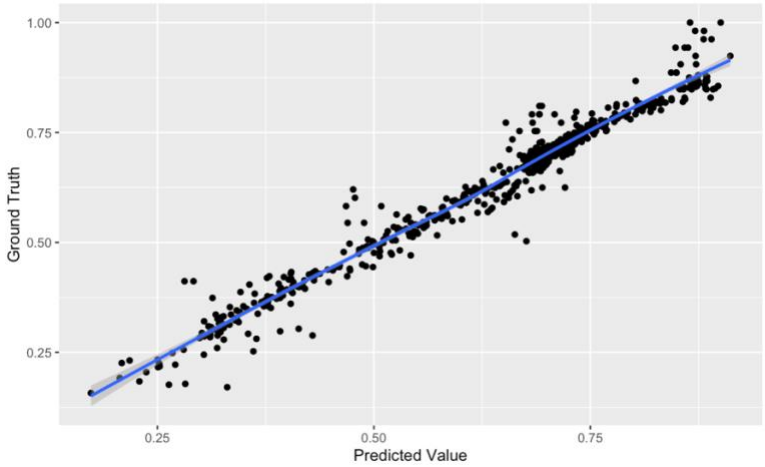
Description: df [589 × 2]

	Predicted Value <dbl>	Ground Truth <dbl>
1	0.4887059	0.5445920
9	0.4172121	0.4022770
12	0.3994631	0.3927894
16	0.3810085	0.3510436
17	0.7848963	0.7874763
20	0.7679567	0.7703985
30	0.6971667	0.7020873
34	0.7252810	0.7419355
45	0.6855418	0.6717268
50	0.3547385	0.2922201

1–10 of 589 rows

Next, we will plot between the predicted value (x-axis) and the ground truth (y-axis) using a scatter plot.

```
plotRF <- ggplot(resultsRF,aes(x=resultsRF$`Predicted Value`,y=resultsRF$`Ground
Truth`)) + geom_point() + stat_smooth() + xlab("Predicted Value") + ylab("Ground Truth")
plotRF
```



From the plot, we can see the distribution of data on the predicted value and ground truth and the line from our model, which is random forest. It can be seen that the distribution of the points is close to the line so this can be an indication of the good performance of the random forest model in predicting the target value.

A linear pattern like the output above can also be assumed that there is a significant correlation between the predicted value and the ground truth and the random forest model has good performance.

This can also be proven by calculating the Mean Square Error with the code below

```
#Calculate Mean Squared Error
MSE_RF <- sum((resultsRF$`Predicted Value` - resultsRF$`Ground Truth`)^2)/nrow(test)
MSE_RF
```

```
[1] 0.001205333
```

The Mean Squared Error result of 0.001205333 shows that the random forest model that has been made has a good ability to predict the value of Life expectancy. The lower the MSE value, the smaller the difference between the actual value and the predicted value, so it can be concluded that the model has a high level of accuracy in predicting the value of Life expectancy.

Step 4: Python Equivalent

The first thing to do is to get a dataframe that has been imputed with mean and normalized data and convert it into a csv dataset. in this case, we will convert the `scaled` dataframe into a csv dataset that we can later directly use for the python equivalent of the random forest regression model. So, the code will be like this

```
write.csv(scaled, "/Users/shafaqonitatingmail.com/Documents/Semester 4/Data Mining and Visualization/scaled.csv", row.names = FALSE)
```

After running the code, the csv dataset will automatically be downloaded into the path we listed and we can immediately do the python equivalent.

Import Libraries

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
```

Load Dataset

```
df = pd.read_csv("scaled.csv")
```

```
df.head() # check if the dataset has been loaded and check the overall output
```

	Year	Life.expectancy	Adult.Mortality	infant.deaths	Alcohol	percentage.expenditure	Hepatitis.B	Measles	BMI	under.five.deaths
0	1.000000	0.544592	0.362881	0.034444	0.0	0.003659	0.653061	0.005439	0.209733	0.0332
1	0.933333	0.447818	0.373961	0.035556	0.0	0.003774	0.622449	0.002319	0.203940	0.0344
2	0.866667	0.447818	0.369806	0.036667	0.0	0.003759	0.642857	0.002027	0.198146	0.0356
3	0.800000	0.440228	0.375346	0.038333	0.0	0.004014	0.673469	0.013135	0.192352	0.0372
4	0.733333	0.434535	0.379501	0.039444	0.0	0.000364	0.683673	0.014200	0.187717	0.0388

Split Dataset: Split the dataset into 80% for train data and 20% for test data

```
X = df.drop("Life.expectancy", axis=1) # independent variables
y = df["Life.expectancy"] # dependent variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Create Model: Initializes a random forest regressor object in Python using scikit-learn library

```
rf = RandomForestRegressor(random_state=42)
```

Training Model: Fits or training a random forest model using the training data

```
rf.fit(X_train, y_train)
```

```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

Get Feature Importance: Calculates the feature importances of a random forest model

```
importances = pd.Series(rf.feature_importances_,
index=X_train.columns).sort_values(ascending=False)
print(importances)
```

```
HIV.AIDS                         0.594121
Adult.Mortality                  0.159651
Income.composition.of.resources  0.143005
Schooling                        0.018830
BMI                              0.014034
under.five.deaths                0.010756
thinness.5.9.years               0.008680
Alcohol                          0.007130
Year                             0.007001
thinness..1.19.years             0.005650
Total.expenditure                0.005315
Measles                          0.003892
GDP                              0.003641
Polio                            0.003590
infant.deaths                    0.003420
percentage.expenditure           0.003385
Diphtheria                       0.003132
Population                       0.002891
Hepatitis.B                      0.001875
dtype: float64
```

In accordance with the importances using R, the 3 variables that greatly contribute to this model are HIV.AIDS, Adult.Mortality, and Income.composition.of.resources.

Predict Model using dataset test: The trained model will make predictions using the test data

```
y_pred = rf.predict(X_test)
```

Results: Create a data frame consisting of predicted value and ground truth

```
resultsRF = pd.DataFrame({"Predicted Value": y_pred, "Ground Truth": y_test})
resultsRF
```

	Predicted Value	Ground Truth
2546	0.698880	0.709677
650	0.754440	0.751423
1740	0.728330	0.719165
177	0.753966	0.768501
1377	0.323529	0.296015
...	...	...
859	0.410949	0.444023
1405	0.709848	0.705882
962	0.543188	0.470588
2658	0.654630	0.631879
2272	0.714421	0.715370

588 rows × 2 columns

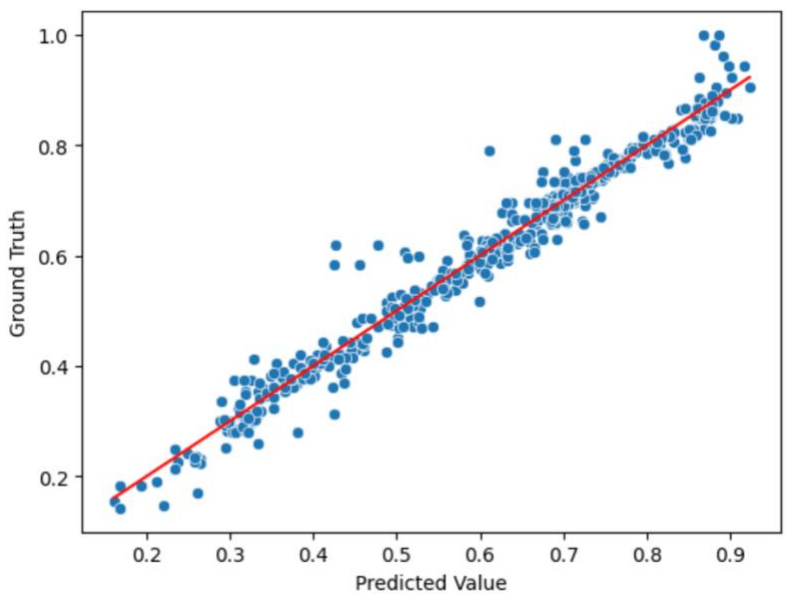
Plot Predicted vs. Ground Truth Values: Plot the result using scatterplot

```
# create scatter plot
sns.scatterplot(x="Predicted Value", y="Ground Truth", data=resultsRF)

# add model line
sns.lineplot(x=resultsRF['Predicted Value'], y=resultsRF['Predicted Value'],
color='red')

# set axis labels
plt.xlabel("Predicted Value")
plt.ylabel("Ground Truth")

# show plot
plt.show()
```



Calculate Mean Squared Error (MSE):

```
MSE_RF = mean_squared_error(y_test, y_pred)
print("Mean Squared Error: ", MSE_RF)
```

```
Mean Squared Error:  0.000978457455815839
```

The Mean Squared Error result of 0.000978457455815839 shows that the random forest model that has been made has a good ability to predict the value of Life expectancy. The lower the MSE value, the smaller the difference between the actual value and the predicted value, so it can be concluded that the model has a high level of accuracy in predicting the value of Life expectancy.

After obtaining the MSE of the two models created with R and python we can see the difference produced by both. This difference can happen for several reasons, one of which is due to the different functions and libraries used in Python and R. The MSE generated by the model created with python has a smaller value where the performance of the model in predicting the target value on data that has never been seen before (testing data) is better than the model created with R.