ANA 505-2 GR5 PROJECT [Jihoon, Ashish, Elizabeth]

2024-12-06

II. Data Import:

EX1: Read the file “Life Expectancy Data.csv” into RStudio

```
Life_Expectancy <- read.csv("/Users/eli/Downloads/Life Expectancy Data.csv")
# Read the CSV file into a data frame named 'Life_Expectancy'Data.csv’
```

III. Exploratory Data Analysis (EDA):

EX2: Generate the following descriptive and summary statistics

1. Describe

```
summary(Life_Expectancy) # Generate summary statistics for each column in the
'Life_Expectancy' data frame, including minimum, 1st quartile, median, mean,
3rd quartile, and maximum values.
```

```
##    Country               Year         Status          Life.expectancy
##  Length:2938        Min.   :2000   Length:2938        Min.   :36.30
##  Class :character   1st Qu.:2004   Class :character   1st Qu.:63.10
##  Mode  :character   Median :2008   Mode  :character   Median :72.10
##                     Mean   :2008                      Mean   :69.22
##                     3rd Qu.:2012                      3rd Qu.:75.70
##                     Max.   :2015                      Max.   :89.00
##                                                       NA's   :10
##  Adult.Mortality infant.deaths       Alcohol        percentage.expenditure
##  Min.   :  1.0   Min.   :   0.0   Min.   : 0.0100   Min.   :    0.000
##  1st Qu.: 74.0   1st Qu.:   0.0   1st Qu.: 0.8775   1st Qu.:    4.685
##  Median :144.0   Median :   3.0   Median : 3.7550   Median :   64.913
##  Mean   :164.8   Mean   :  30.3   Mean   : 4.6029   Mean   :  738.251
##  3rd Qu.:228.0   3rd Qu.:  22.0   3rd Qu.: 7.7025   3rd Qu.:  441.534
##  Max.   :723.0   Max.   :1800.0   Max.   :17.8700   Max.   :19479.912
##  NA's   :10                       NA's   :194
##   Hepatitis.B       Measles               BMI        under.five.deaths
##  Min.   : 1.00   Min.   :     0.0   Min.   : 1.00   Min.   :   0.00
##  1st Qu.:77.00   1st Qu.:     0.0   1st Qu.:19.30   1st Qu.:   0.00
##  Median :92.00   Median :    17.0   Median :43.50   Median :   4.00
##  Mean   :80.94   Mean   :  2419.6   Mean   :38.32   Mean   :  42.04
##  3rd Qu.:97.00   3rd Qu.:   360.2   3rd Qu.:56.20   3rd Qu.:  28.00
##  Max.   :99.00   Max.   :212183.0   Max.   :87.30   Max.   :2500.00
##  NA's   :553                        NA's   :34
##      Polio       Total.expenditure   Diphtheria       HIV.AIDS
```

```
##  Min.   : 3.00   Min.   : 0.370    Min.   : 2.00   Min.   : 0.100
##  1st Qu.:78.00   1st Qu.: 4.260    1st Qu.:78.00   1st Qu.: 0.100
##  Median :93.00   Median : 5.755    Median :93.00   Median : 0.100
##  Mean   :82.55   Mean   : 5.938    Mean   :82.32   Mean   : 1.742
##  3rd Qu.:97.00   3rd Qu.: 7.492    3rd Qu.:97.00   3rd Qu.: 0.800
##  Max.   :99.00   Max.   :17.600    Max.   :99.00   Max.   :50.600
##  NA's   :19      NA's   :226       NA's   :19
##       GDP              Population        thinness..1.19.years
##  Min.   :     1.68   Min.   :3.400e+01   Min.   : 0.10
##  1st Qu.:   463.94   1st Qu.:1.958e+05   1st Qu.: 1.60
##  Median :  1766.95   Median :1.387e+06   Median : 3.30
##  Mean   :  7483.16   Mean   :1.275e+07   Mean   : 4.84
##  3rd Qu.:  5910.81   3rd Qu.:7.420e+06   3rd Qu.: 7.20
##  Max.   :119172.74   Max.   :1.294e+09   Max.   :27.70
##  NA's   :448         NA's   :652         NA's   :34
##  thinness.5.9.years Income.composition.of.resources   Schooling
##  Min.   : 0.10      Min.   :0.0000                   Min.   : 0.00
##  1st Qu.: 1.50      1st Qu.:0.4930                   1st Qu.:10.10
##  Median : 3.30      Median :0.6770                   Median :12.30
##  Mean   : 4.87      Mean   :0.6276                   Mean   :11.99
##  3rd Qu.: 7.20      3rd Qu.:0.7790                   3rd Qu.:14.30
##  Max.   :28.60      Max.   :0.9480                   Max.   :20.70
##  NA's   :34         NA's   :167                      NA's   :163
```

2. Sum, mean, median, std for the following columns

```
#1) D: Life Expectancy
sum(Life_Expectancy$Life.expectancy, na.rm = TRUE) # Calculate the total sum
of the Life Expectancy column, ignoring missing values

## [1] 202690.6

mean(Life_Expectancy$Life.expectancy, na.rm = TRUE) # Calculate the mean of
the Life Expectancy column, ignoring missing values

## [1] 69.22493

median(Life_Expectancy$Life.expectancy, na.rm = TRUE) # Calculate the median
of the Life Expectancy column, ignoring missing values

## [1] 72.1

sd(Life_Expectancy$Life.expectancy, na.rm = TRUE) # Calculate the standard
deviation of the Life Expectancy column, ignoring missing values

## [1] 9.523867

#2) E: Adult Mortality
sum(Life_Expectancy$Adult.Mortality, na.rm = TRUE) # Calculate the total sum
of the Adult Mortality column, ignoring missing values

## [1] 482524
```

```r
mean(Life_Expectancy$Adult.Mortality, na.rm = TRUE) # Calculate the mean of the Adult Mortality column, ignoring missing values
```

```
## [1] 164.7964
```

```r
median(Life_Expectancy$Adult.Mortality, na.rm = TRUE) # Calculate the median of the Adult Mortality column, ignoring missing values
```

```
## [1] 144
```

```r
sd(Life_Expectancy$Adult.Mortality, na.rm = TRUE) # Calculate the standard deviation of the Adult Mortality column, ignoring missing values
```

```
## [1] 124.2921
```

```r
#3) F: Infant Death
sum(Life_Expectancy$infant.deaths) # Calculate the total sum of the infant deaths column, ignoring missing values
```

```
## [1] 89033
```

```r
mean(Life_Expectancy$infant.deaths) # Calculate the mean of the infant deaths column, ignoring missing values
```

```
## [1] 30.30395
```

```r
median(Life_Expectancy$infant.deaths) # Calculate the median of the infant deaths column, ignoring missing values
```

```
## [1] 3
```

```r
sd(Life_Expectancy$infant.deaths) # Calculate the standard deviation of the infant deaths column, ignoring missing values
```

```
## [1] 117.9265
```

```r
#4) I: Hepatitis B
sum(Life_Expectancy$Hepatitis.B, na.rm = TRUE) # Calculate the total sum of the Hepatitis B column, ignoring missing values
```

```
## [1] 193043
```

```r
mean(Life_Expectancy$Hepatitis.B, na.rm = TRUE)  # Calculate the mean of the Hepatitis B column, ignoring missing values
```

```
## [1] 80.94046
```

```r
median(Life_Expectancy$Hepatitis.B, na.rm = TRUE) # Calculate the median of the Hepatitis B column, ignoring missing values
```

```
## [1] 92
```

```r
sd(Life_Expectancy$Hepatitis.B, na.rm = TRUE)  # Calculate the standard deviation of the Hepatitis B column, ignoring missing values
```

```
## [1] 25.07002

#Note: some of these variables have missing values, na.rm = TRUE removes the
missing values
```

EX3: Return to a dataframe a full correlation of Life Expectancy and Measles in 2015.

```
LE_2015 <- subset(Life_Expectancy, Year == 2015) # Filter the dataset
'Life_Expectancy' to include only rows where the Year is 2015.
head(LE_2015) #prints the first six rows of the 2015 subset

##                Country Year     Status Life.expectancy Adult.Mortality
## 1          Afghanistan 2015 Developing            65.0             263
## 17             Albania 2015 Developing            77.8              74
## 33             Algeria 2015 Developing            75.6              19
## 49              Angola 2015 Developing            52.4             335
## 65 Antigua and Barbuda 2015 Developing            76.4              13
## 81           Argentina 2015 Developing            76.3             116
##    infant.deaths Alcohol percentage.expenditure Hepatitis.B Measles  BMI
## 1             62    0.01               71.27962          65    1154 19.1
## 17             0    4.60              364.97523          99       0 58.0
## 33            21      NA                0.00000          95      63 59.5
## 49            66      NA                0.00000          64     118 23.3
## 65             0      NA                0.00000          99       0 47.7
## 81             8      NA                0.00000          94       0 62.8
##    under.five.deaths Polio Total.expenditure Diphtheria HIV.AIDS
GDP
## 1                 83     6              8.16         65      0.1
584.2592
## 17                 0    99              6.00         99      0.1
3954.2278
## 33                24    95                NA         95      0.1
4132.7629
## 49                98     7                NA         64      1.9
3695.7937
## 65                 0    86                NA         99      0.2
13566.9541
## 81                 9    93                NA         94      0.1
13467.1236
##    Population thinness..1.19.years thinness.5.9.years
## 1    33736494                 17.2               17.3
## 17      28873                  1.2                1.3
## 33   39871528                  6.0                5.8
## 49    2785935                  8.3                8.2
## 65         NA                  3.3                3.3
## 81   43417765                  1.0                0.9
##    Income.composition.of.resources Schooling
## 1                            0.479      10.1
## 17                           0.762      14.2
```

```
## 33                            0.743      14.4
## 49                            0.531      11.4
## 65                            0.784      13.9
## 81                            0.826      17.3
```

```
cor(LE_2015[,4], LE_2015[,10], use = "complete.obs") # Calculate the
correlation coefficient between Life Expectancy(col 4) and Measles (col 10)
for the year 2015.
```

```
## [1] -0.07461652
```

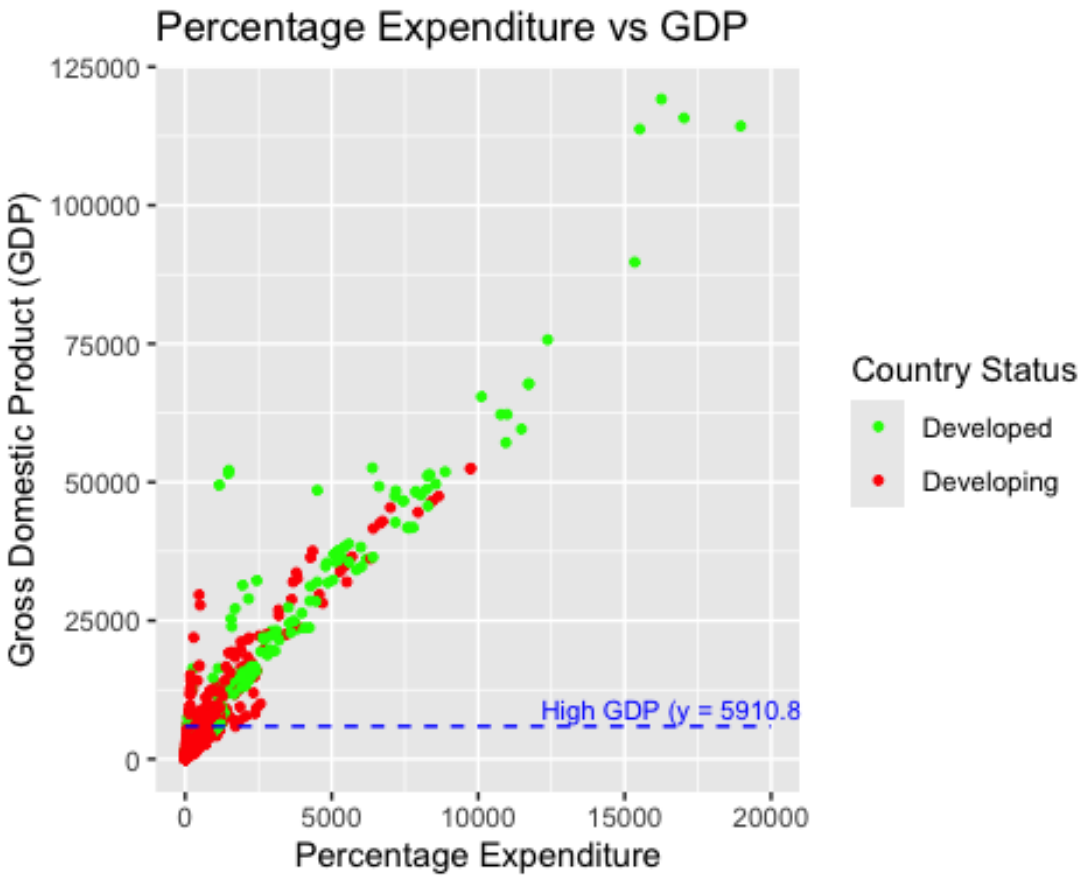
EX4: Return to a dataframe a full covariance of Infant Deaths and Polio in 2015.

```
cov(LE_2015[,6], LE_2015[,13], use = "complete.obs") # Calculate the
covariance between Infant Deaths (column 6) and Polio (column 13) for the
year 2015.
```

```
## [1] -263.2172
```

EX5: Use ggplot2 to draw a plot of Percentage Expenditure and GDP. Use ticks, labels, annotations and legends.

```
library(ggplot2)
LE_clean <- na.omit(Life_Expectancy) # Remove rows with missing values
ggplot(data = LE_clean, mapping = aes(x = percentage.expenditure, y = GDP,
color = Status)) + #plots Percentage Expenditure vs GDP
  geom_point(size = 1) + #adds ticks
  labs(
    title = "Percentage Expenditure vs GDP", #adds title
    x = "Percentage Expenditure", #adds x-axis label
    y = "Gross Domestic Product (GDP)", #adds y-axis label
    color = "Country Status") + #adds legend
  scale_color_manual(values = c("Developing" = "red", "Developed" = "green"))
+ # Custom colors for 'developing' and 'developed'
  annotate("text", x = 17000, y = 9000, label = "High GDP (y = 5910.81)",
color = "blue", size = 3) + # Add text annotation
  annotate("segment", x = 0, xend = 20000, y = 5910.81, yend = 5910.81, color
= "blue", linetype = "dashed")  # Add a horizontal line annotation
```



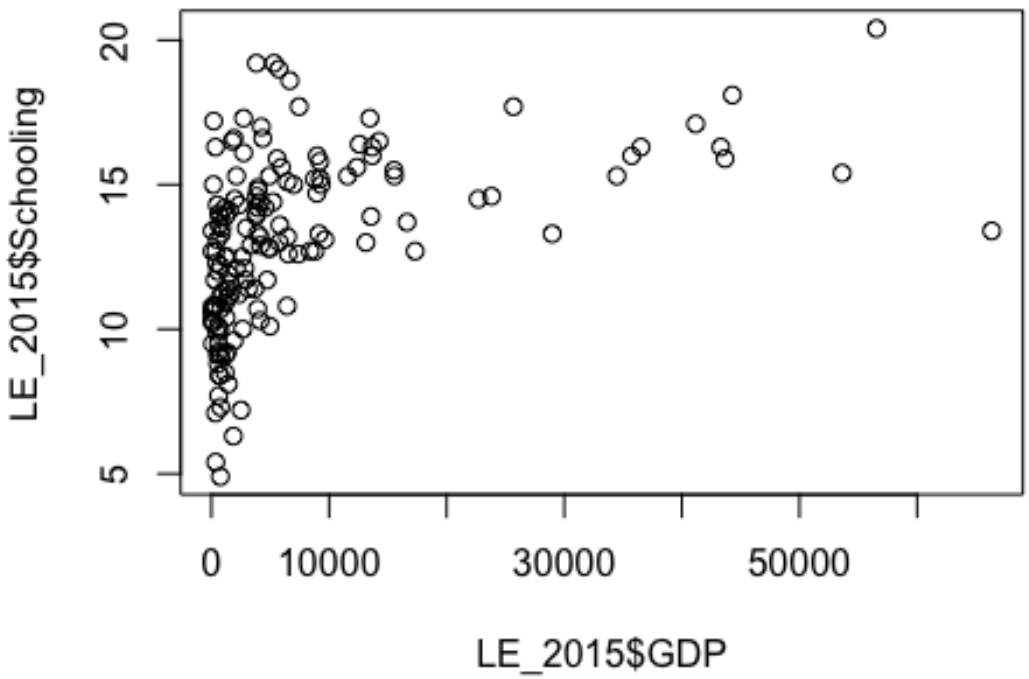

```
#Note: High GDP is defined using the 3rd quartile value for GDP, as
calculated in the EX2 summary stats.
```

EX6: Plot a histogram of Life Expectancy

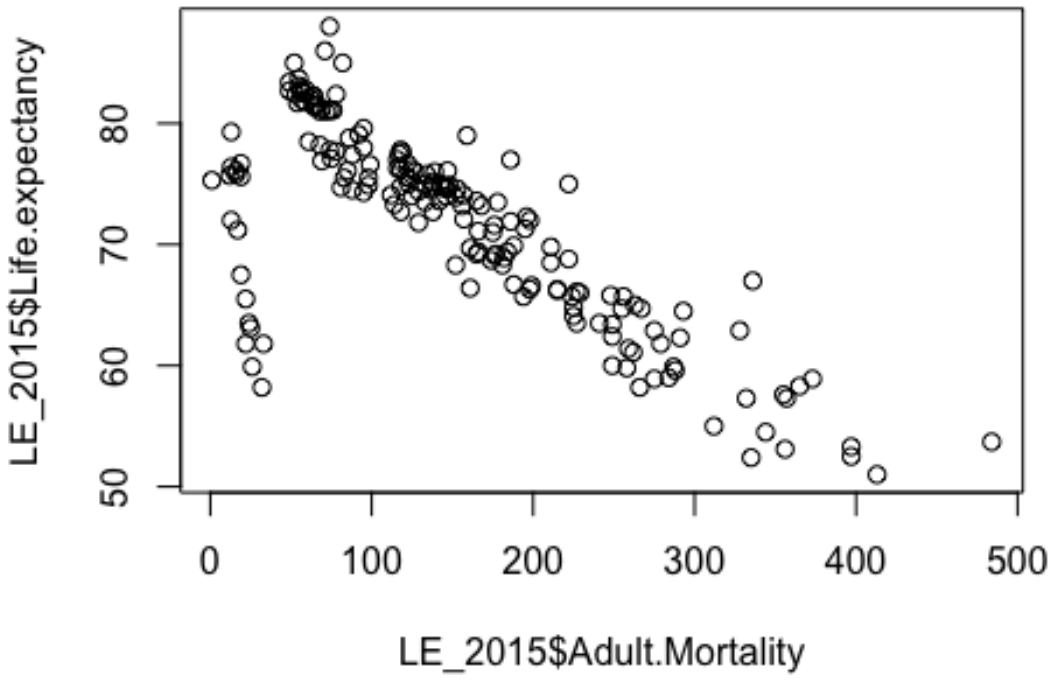
```
hist(Life_Expectancy$Life.expectancy) # Create a histogram of the Life
Expectancy column.
```

EX7: Use R methods to produce the following scatter plots in 2015:

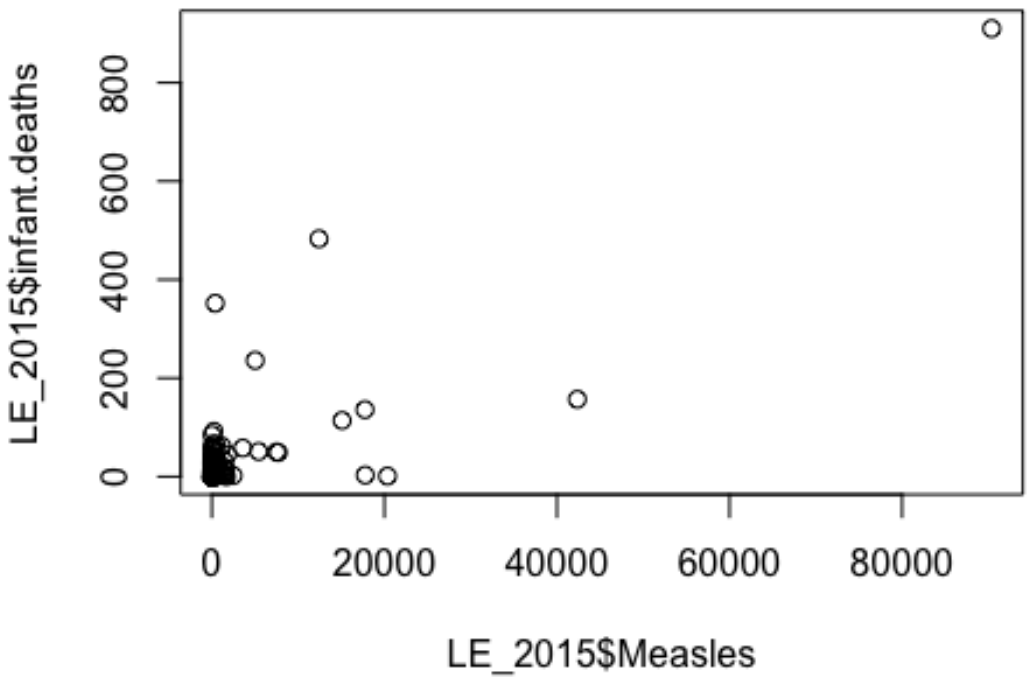
```r
#1) Schooling vs GDP
plot(LE_2015$Schooling~LE_2015$GDP) # Create a scatter plot to examine the relationship between GDP and Schooling.
```



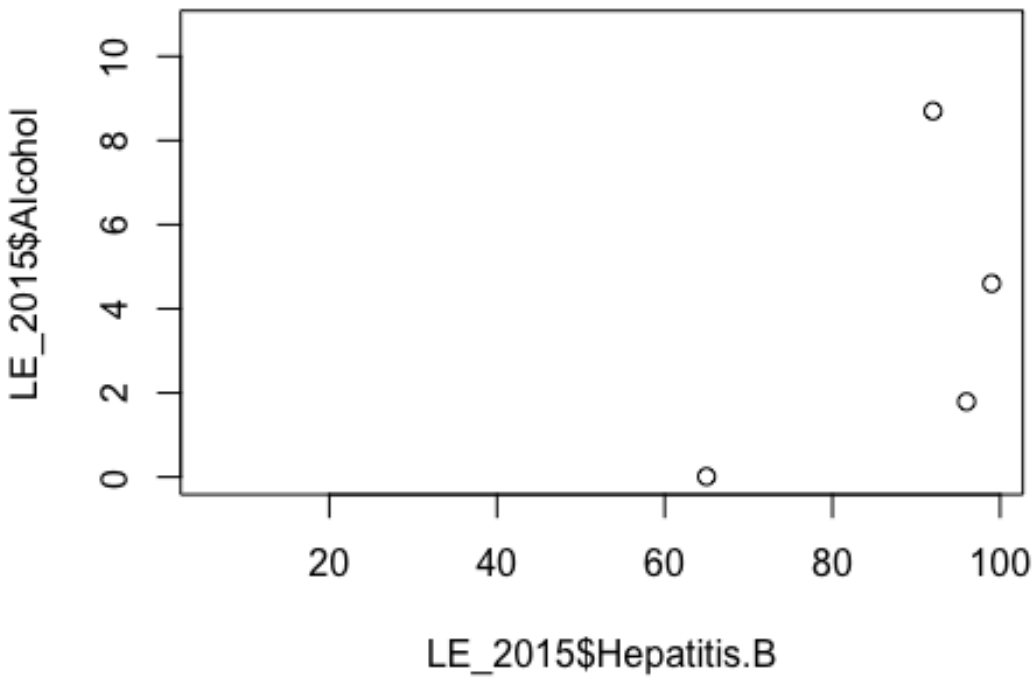

```r
#2) Life Expectancy vs Adult Mortality
plot(LE_2015$Life.expectancy~LE_2015$Adult.Mortality) # Create a scatter plot to explore the relationship between Adult Mortality and Life Expectancy.
```



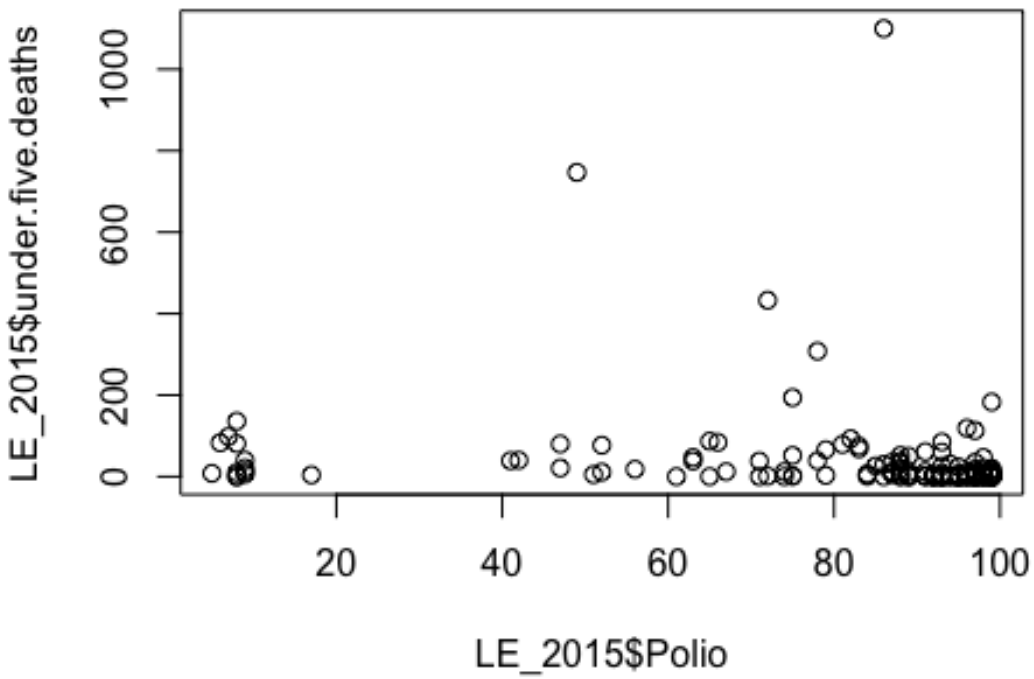

```r
#3) Infant Deaths vs Measles
plot(LE_2015$infant.deaths~LE_2015$Measles) # Create a scatter plot to
investigate the relationship between Infant deaths and Measles.
```




```r
#4) Alcohol vs Hepatitis B
plot(LE_2015$Alcohol~LE_2015$Hepatitis.B) # Create a scatter plot to examine the relationship between Alcohol consumption and Hepatitis B.
```




```r
#5) Under Five Death vs Polio
plot(LE_2015$under.five.deaths~LE_2015$Polio) # Create a scatter plot to
explore the relationship between Under five deaths and Polio.
```



IV. Modelling: You aim to identify factors that contribute to the likelihood of diabetes using classification and clustering methods.

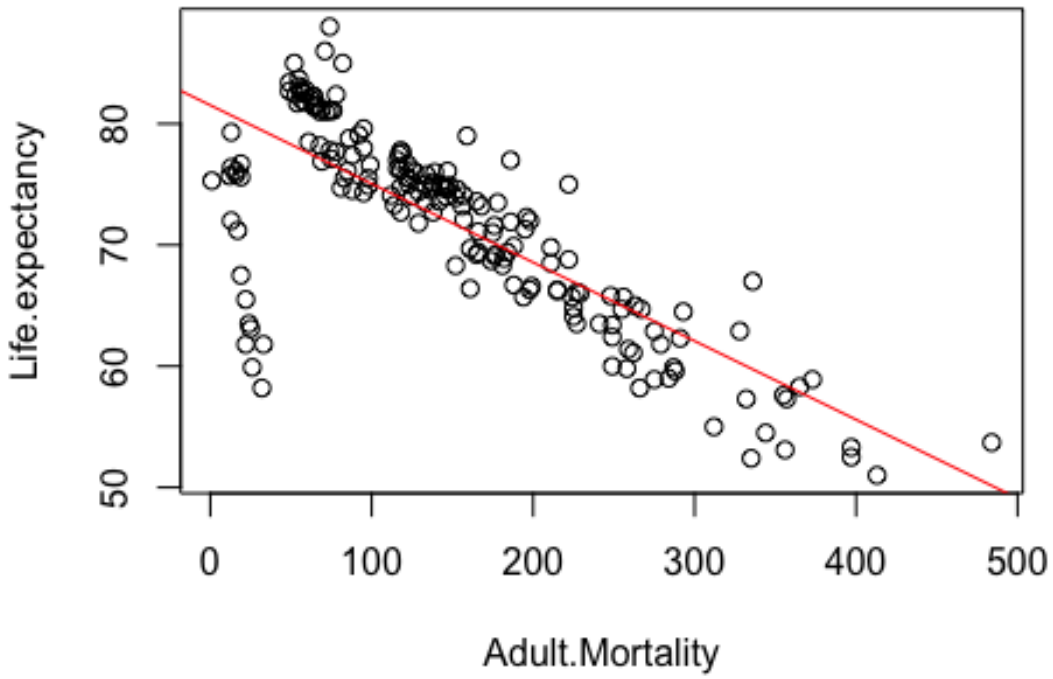
EX8: Modelling: Classification: To build a model that predicts Life Expectancy in 2015 based on several other factors using Linear Regression.

```
linmod1=lm(Life.expectancy ~ Adult.Mortality, data = LE_2015) #create a
linear regression model for Life.expectancy and Adult.Mortality in 2015
summary(linmod1) #prints the results of our first linear regression

##
## Call:
## lm(formula = Life.expectancy ~ Adult.Mortality, data = LE_2015)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.2580  -1.8069   0.7741   2.9917  11.2668
```

```
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     81.53404    0.70303  115.97   <2e-16 ***
## Adult.Mortality -0.06488    0.00388  -16.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.107 on 181 degrees of freedom
## Multiple R-squared:  0.607,  Adjusted R-squared:  0.6049
## F-statistic: 279.6 on 1 and 181 DF,  p-value: < 2.2e-16
```

```
plot(Life.expectancy ~ Adult.Mortality, data = LE_2015) #plots a scatterplot
of our variables
abline(linmod1, col="red") #adds our regression line to the scatterplot in
red
```

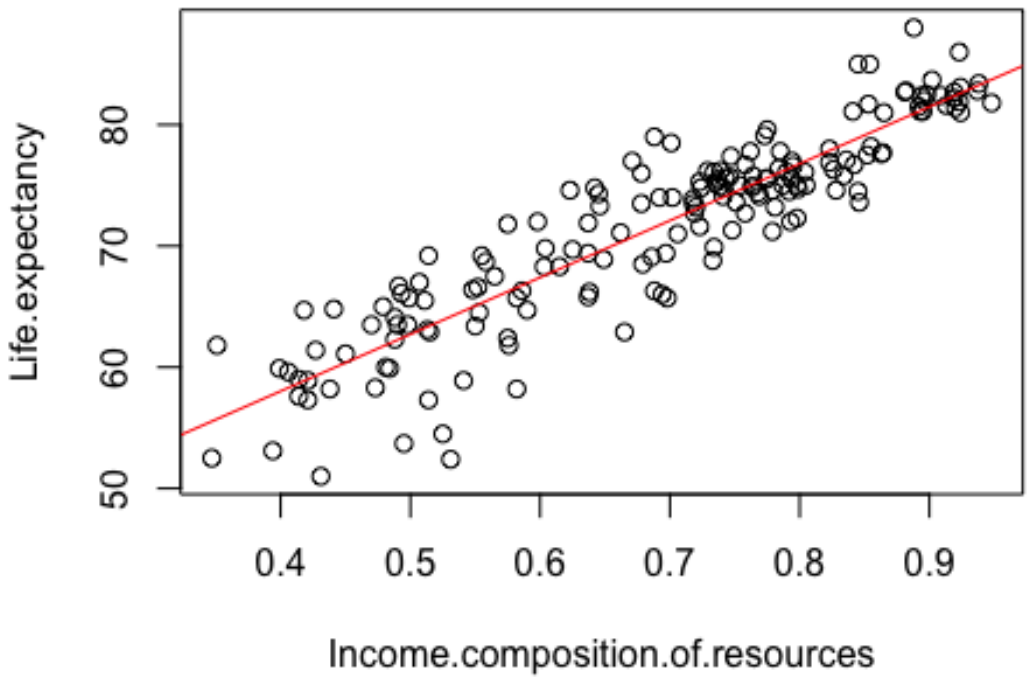



```
linmod2=lm(Life.expectancy ~ Income.composition.of.resources, data = LE_2015)
#create a linear regression model for Life.expectancy and
Income.composition.of.resources in 2015
summary(linmod2) #prints the results of our second linear regression
```

```
##
## Call:
```

```
## lm(formula = Life.expectancy ~ Income.composition.of.resources,
##     data = LE_2015)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.770  -1.733   0.132   2.110   7.463
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       39.254      1.178   33.33   <2e-16 ***
## Income.composition.of.resources   46.923      1.662   28.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.345 on 171 degrees of freedom
##   (10 observations deleted due to missingness)
## Multiple R-squared:  0.8233, Adjusted R-squared:  0.8223
## F-statistic: 796.8 on 1 and 171 DF,  p-value: < 2.2e-16
```

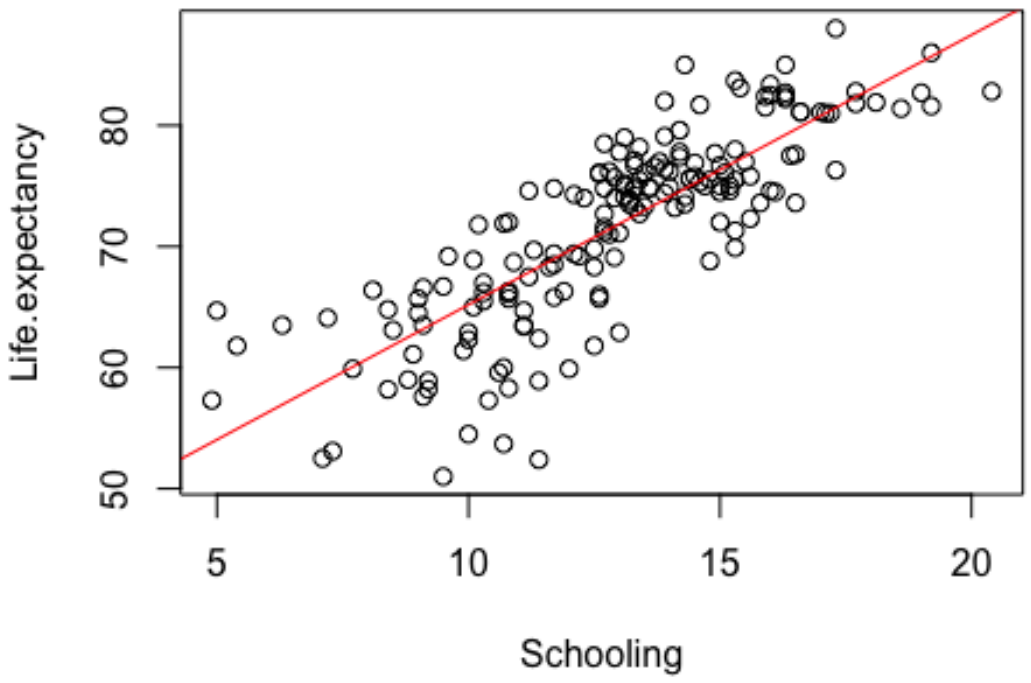
```
plot(Life.expectancy ~ Income.composition.of.resources, data = LE_2015)
#plots a scatterplot of our variables
abline(linmod2, col="red") #adds our regression line to the scatterplot in
red
```



```r
linmod3=lm(Life.expectancy ~ Schooling, data = LE_2015) #create a linear regression model for Life.expectancy and Schooling in 2015
summary(linmod3) #prints the results of our third linear regression
```

```
##
## Call:
## lm(formula = Life.expectancy ~ Schooling, data = LE_2015)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.909  -2.547   0.317   3.170  10.655
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.9016     1.5870   27.03   <2e-16 ***
## Schooling     2.2287     0.1198   18.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.575 on 171 degrees of freedom
##   (10 observations deleted due to missingness)
## Multiple R-squared:  0.6694, Adjusted R-squared:  0.6675
## F-statistic: 346.2 on 1 and 171 DF,  p-value: < 2.2e-16
```

```r
plot(Life.expectancy ~ Schooling, data = LE_2015) #plots a scatterplot of our variables
abline(linmod3, col="red") #adds our regression line to the scatterplot in red
```



EX9: Modelling: Classification: To build a model that predicts Life Expectancy in 2015 based on several other factors using Random Forest (randomForest).

```
#Random forest with the entire Life expectancy dataset
#We first have to split the data into two subsets: training (70%) and test
(30%)
ind3 <- sample(2, nrow(LE_clean), replace=TRUE, prob=c(0.8, 0.2)) #makes two
subsets
train3Data <- LE_clean[ind3==1,] #80% for training subset
val3Data <- LE_clean[ind3==2,] #20% for validation subset
#install.packages('randomForest') #Loads in the package
library(randomForest)

## randomForest 4.7-1.2

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rf3 <- randomForest(Life.expectancy ~., data=train3Data, ntree=100,
proximity=TRUE) #Predict Life.expectancy with all the variables in the data
rfpredtable<-table(predict(rf3), train3Data$Life.expectancy) #Creates a table
for the predictions
head(rfpredtable) #because this table is extensive, we decided to display on
the first six rows of the predictions table
```

```
##
##                   44 44.5 44.6 44.8 45.3 45.4 45.5 45.9 46 46.2 46.4 46.7
47.1
##   46.2265416666667  0    1    0    0    0    0    0    0  0    0    0    0
0
##   46.33425          0    0    0    0    0    0    0    0  1    0    0    0
0
##   46.4763425925926  0    0    1    0    0    0    0    0  0    0    0    0
0
##   46.6543137254902  0    0    0    1    0    0    0    0  0    0    0    0
0
##   47.1455729166667  0    0    0    0    1    0    0    0  0    0    0    0
0
##   47.3101923076923  0    1    0    0    0    0    0    0  0    0    0    0
0
##
##                   47.8 48.1 48.2 48.4 48.5 48.6 48.7 48.8 48.9 49.1 49.2
49.3
##   46.2265416666667    0    0    0    0    0    0    0    0    0    0    0
0
##   46.33425            0    0    0    0    0    0    0    0    0    0    0
0
##   46.4763425925926    0    0    0    0    0    0    0    0    0    0    0
0
##   46.6543137254902    0    0    0    0    0    0    0    0    0    0    0
0
##   47.1455729166667    0    0    0    0    0    0    0    0    0    0    0
0
##   47.3101923076923    0    0    0    0    0    0    0    0    0    0    0
0
##
##                   49.4 49.5 49.6 49.7 49.8 49.9 50 51 51.1 51.2 51.3 51.4
51.6
##   46.2265416666667    0    0    0    0    0    0  0  0    0    0    0    0
0
##   46.33425            0    0    0    0    0    0  0  0    0    0    0    0
0
##   46.4763425925926    0    0    0    0    0    0  0  0    0    0    0    0
0
```

```
##   46.6543137254902    0    0    0    0    0    0  0  0    0    0    0    0
0
##   47.1455729166667    0    0    0    0    0    0  0  0    0    0    0    0
0
##   47.3101923076923    0    0    0    0    0    0  0  0    0    0    0    0
0
##
##                   51.7 51.8 52 52.1 52.2 52.3 52.4 52.6 52.7 52.8 52.9 53
53.2
##   46.2265416666667    0    0  0    0    0    0    0    0    0    0    0  0
0
##   46.33425            0    0  0    0    0    0    0    0    0    0    0  0
0
##   46.4763425925926    0    0  0    0    0    0    0    0    0    0    0  0
0
##   46.6543137254902    0    0  0    0    0    0    0    0    0    0    0  0
0
##   47.1455729166667    0    0  0    0    0    0    0    0    0    0    0  0
0
##   47.3101923076923    0    0  0    0    0    0    0    0    0    0    0  0
0
##
##                   53.3 53.4 53.6 53.7 53.8 54 54.1 54.2 54.3 54.5 54.8
54.9 55
##   46.2265416666667    0    0    0    0    0  0    0    0    0    0    0
0  0
##   46.33425            0    0    0    0    0  0    0    0    0    0    0
0  0
##   46.4763425925926    0    0    0    0    0  0    0    0    0    0    0
0  0
##   46.6543137254902    0    0    0    0    0  0    0    0    0    0    0
0  0
##   47.1455729166667    0    0    0    0    0  0    0    0    0    0    0
0  0
##   47.3101923076923    0    0    0    0    0  0    0    0    0    0    0
0  0
##
##                   55.3 55.5 55.6 55.7 55.9 56 56.1 56.2 56.3 56.4 56.5
56.6
##   46.2265416666667    0    0    0    0    0  0    0    0    0    0    0
0
##   46.33425            0    0    0    0    0  0    0    0    0    0    0
0
##   46.4763425925926    0    0    0    0    0  0    0    0    0    0    0
0
##   46.6543137254902    0    0    0    0    0  0    0    0    0    0    0
0
##   47.1455729166667    0    0    0    0    0  0    0    0    0    0    0
0
##   47.3101923076923    0    0    0    0    0  0    0    0    0    0    0
```

```
##
##                   56.7 56.8 56.9 57 57.1 57.2 57.3 57.4 57.5 57.6 57.8
57.9 58
##   46.2265416666667    0    0    0  0    0    0    0    0    0    0    0
0  0
##   46.33425            0    0    0  0    0    0    0    0    0    0    0
0  0
##   46.4763425925926    0    0    0  0    0    0    0    0    0    0    0
0  0
##   46.6543137254902    0    0    0  0    0    0    0    0    0    0    0
0  0
##   47.1455729166667    0    0    0  0    0    0    0    0    0    0    0
0  0
##   47.3101923076923    0    0    0  0    0    0    0    0    0    0    0
0  0
##
##                   58.1 58.2 58.3 58.4 58.6 58.8 58.9 59 59.1 59.2 59.3
59.4
##   46.2265416666667    0    0    0    0    0    0    0  0    0    0    0
0
##   46.33425            0    0    0    0    0    0    0  0    0    0    0
0
##   46.4763425925926    0    0    0    0    0    0    0  0    0    0    0
0
##   46.6543137254902    0    0    0    0    0    0    0  0    0    0    0
0
##   47.1455729166667    0    0    0    0    0    0    0  0    0    0    0
0
##   47.3101923076923    0    0    0    0    0    0    0  0    0    0    0
0
##
##                   59.5 59.6 59.7 59.8 59.9 60 61 61.1 61.2 61.3 61.4 61.6
61.7
##   46.2265416666667    0    0    0    0    0  0  0    0    0    0    0    0
0
##   46.33425            0    0    0    0    0  0  0    0    0    0    0    0
0
##   46.4763425925926    0    0    0    0    0  0  0    0    0    0    0    0
0
##   46.6543137254902    0    0    0    0    0  0  0    0    0    0    0    0
0
##   47.1455729166667    0    0    0    0    0  0  0    0    0    0    0    0
0
##   47.3101923076923    0    0    0    0    0  0  0    0    0    0    0    0
0
##
##                   61.8 62 62.1 62.2 62.3 62.4 62.5 62.6 62.7 62.8 62.9 63
63.1
##   46.2265416666667    0  0    0    0    0    0    0    0    0    0    0  0
```

```
0
##   46.33425           0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.4763425925926   0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.6543137254902   0  0    0    0    0    0    0    0    0    0    0  0
0
##   47.1455729166667   0  0    0    0    0    0    0    0    0    0    0  0
0
##   47.3101923076923   0  0    0    0    0    0    0    0    0    0    0  0
0
##
##                   63.2 63.3 63.4 63.5 63.7 63.8 63.9 64 64.1 64.2 64.3
64.4
##   46.2265416666667   0    0    0    0    0    0    0  0    0    0    0
0
##   46.33425           0    0    0    0    0    0    0  0    0    0    0
0
##   46.4763425925926   0    0    0    0    0    0    0  0    0    0    0
0
##   46.6543137254902   0    0    0    0    0    0    0  0    0    0    0
0
##   47.1455729166667   0    0    0    0    0    0    0  0    0    0    0
0
##   47.3101923076923   0    0    0    0    0    0    0  0    0    0    0
0
##
##                   64.5 64.6 64.7 64.8 64.9 65 65.1 65.2 65.3 65.4 65.5
65.6
##   46.2265416666667   0    0    0    0    0  0    0    0    0    0    0
0
##   46.33425           0    0    0    0    0  0    0    0    0    0    0
0
##   46.4763425925926   0    0    0    0    0  0    0    0    0    0    0
0
##   46.6543137254902   0    0    0    0    0  0    0    0    0    0    0
0
##   47.1455729166667   0    0    0    0    0  0    0    0    0    0    0
0
##   47.3101923076923   0    0    0    0    0  0    0    0    0    0    0
0
##
##                   65.7 65.8 65.9 66 66.1 66.2 66.3 66.4 66.5 66.6 66.7
66.8
##   46.2265416666667   0    0    0  0    0    0    0    0    0    0    0
0
##   46.33425           0    0    0  0    0    0    0    0    0    0    0
0
##   46.4763425925926   0    0    0  0    0    0    0    0    0    0    0
0
```

```
##   46.6543137254902    0    0    0  0    0    0    0    0    0    0    0
0
##   47.1455729166667    0    0    0  0    0    0    0    0    0    0    0
0
##   47.3101923076923    0    0    0  0    0    0    0    0    0    0    0
0
##
##                   66.9 67 67.1 67.2 67.3 67.4 67.5 67.6 67.7 67.8 67.9 68
68.1
##   46.2265416666667    0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.33425            0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.4763425925926    0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.6543137254902    0  0    0    0    0    0    0    0    0    0    0  0
0
##   47.1455729166667    0  0    0    0    0    0    0    0    0    0    0  0
0
##   47.3101923076923    0  0    0    0    0    0    0    0    0    0    0  0
0
##
##                   68.2 68.3 68.4 68.5 68.6 68.7 68.8 68.9 69 69.1 69.2
69.3
##   46.2265416666667    0    0    0    0    0    0    0    0  0    0    0
0
##   46.33425            0    0    0    0    0    0    0    0  0    0    0
0
##   46.4763425925926    0    0    0    0    0    0    0    0  0    0    0
0
##   46.6543137254902    0    0    0    0    0    0    0    0  0    0    0
0
##   47.1455729166667    0    0    0    0    0    0    0    0  0    0    0
0
##   47.3101923076923    0    0    0    0    0    0    0    0  0    0    0
0
##
##                   69.4 69.5 69.6 69.7 69.8 69.9 70 71 71.1 71.2 71.3 71.4
71.5
##   46.2265416666667    0    0    0    0    0    0  0  0    0    0    0    0
0
##   46.33425            0    0    0    0    0    0  0  0    0    0    0    0
0
##   46.4763425925926    0    0    0    0    0    0  0  0    0    0    0    0
0
##   46.6543137254902    0    0    0    0    0    0  0  0    0    0    0    0
0
##   47.1455729166667    0    0    0    0    0    0  0  0    0    0    0    0
0
##   47.3101923076923    0    0    0    0    0    0  0  0    0    0    0    0
```

```
0
##
##                   71.6 71.7 71.8 71.9 72 72.1 72.2 72.3 72.4 72.5 72.6
72.7
##   46.2265416666667    0    0    0    0  0    0    0    0    0    0    0
0
##   46.33425            0    0    0    0  0    0    0    0    0    0    0
0
##   46.4763425925926    0    0    0    0  0    0    0    0    0    0    0
0
##   46.6543137254902    0    0    0    0  0    0    0    0    0    0    0
0
##   47.1455729166667    0    0    0    0  0    0    0    0    0    0    0
0
##   47.3101923076923    0    0    0    0  0    0    0    0    0    0    0
0
##
##                   72.8 72.9 73 73.1 73.2 73.3 73.4 73.5 73.6 73.7 73.8
73.9 74
##   46.2265416666667    0    0  0    0    0    0    0    0    0    0    0
0  0
##   46.33425            0    0  0    0    0    0    0    0    0    0    0
0  0
##   46.4763425925926    0    0  0    0    0    0    0    0    0    0    0
0  0
##   46.6543137254902    0    0  0    0    0    0    0    0    0    0    0
0  0
##   47.1455729166667    0    0  0    0    0    0    0    0    0    0    0
0  0
##   47.3101923076923    0    0  0    0    0    0    0    0    0    0    0
0  0
##
##                   74.1 74.2 74.3 74.4 74.5 74.6 74.7 74.8 74.9 75 75.1
75.2
##   46.2265416666667    0    0    0    0    0    0    0    0    0  0    0
0
##   46.33425            0    0    0    0    0    0    0    0    0  0    0
0
##   46.4763425925926    0    0    0    0    0    0    0    0    0  0    0
0
##   46.6543137254902    0    0    0    0    0    0    0    0    0  0    0
0
##   47.1455729166667    0    0    0    0    0    0    0    0    0  0    0
0
##   47.3101923076923    0    0    0    0    0    0    0    0    0  0    0
0
##
##                   75.3 75.4 75.5 75.6 75.7 75.8 75.9 76 76.1 76.2 76.3
76.4
##   46.2265416666667    0    0    0    0    0    0    0  0    0    0    0
```

```
##   46.33425           0    0    0    0    0    0    0  0    0    0    0
0
##   46.4763425925926   0    0    0    0    0    0    0  0    0    0    0
0
##   46.6543137254902   0    0    0    0    0    0    0  0    0    0    0
0
##   47.1455729166667   0    0    0    0    0    0    0  0    0    0    0
0
##   47.3101923076923   0    0    0    0    0    0    0  0    0    0    0
0
##
##                    76.5 76.6 76.7 76.8 76.9 77 77.1 77.2 77.3 77.5 77.6
77.7
##   46.2265416666667   0    0    0    0    0  0    0    0    0    0    0
0
##   46.33425           0    0    0    0    0  0    0    0    0    0    0
0
##   46.4763425925926   0    0    0    0    0  0    0    0    0    0    0
0
##   46.6543137254902   0    0    0    0    0  0    0    0    0    0    0
0
##   47.1455729166667   0    0    0    0    0  0    0    0    0    0    0
0
##   47.3101923076923   0    0    0    0    0  0    0    0    0    0    0
0
##
##                    77.8 78 78.1 78.2 78.3 78.4 78.5 78.6 78.7 78.8 78.9 79
79.1
##   46.2265416666667   0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.33425           0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.4763425925926   0  0    0    0    0    0    0    0    0    0    0  0
0
##   46.6543137254902   0  0    0    0    0    0    0    0    0    0    0  0
0
##   47.1455729166667   0  0    0    0    0    0    0    0    0    0    0  0
0
##   47.3101923076923   0  0    0    0    0    0    0    0    0    0    0  0
0
##
##                    79.2 79.3 79.4 79.5 79.6 79.7 79.8 79.9 80 81 81.1 81.2
81.3
##   46.2265416666667   0    0    0    0    0    0    0    0  0  0    0    0
0
##   46.33425           0    0    0    0    0    0    0    0  0  0    0    0
0
##   46.4763425925926   0    0    0    0    0    0    0    0  0  0    0    0
0
```

```
##   46.6543137254902    0    0    0    0    0    0    0    0  0  0    0    0
0
##   47.1455729166667    0    0    0    0    0    0    0    0  0  0    0    0
0
##   47.3101923076923    0    0    0    0    0    0    0    0  0  0    0    0
0
##
##                   81.4 81.5 81.6 81.7 81.8 81.9 82 82.2 82.3 82.4 82.5
82.6
##   46.2265416666667    0    0    0    0    0    0  0    0    0    0    0
0
##   46.33425            0    0    0    0    0    0  0    0    0    0    0
0
##   46.4763425925926    0    0    0    0    0    0  0    0    0    0    0
0
##   46.6543137254902    0    0    0    0    0    0  0    0    0    0    0
0
##   47.1455729166667    0    0    0    0    0    0  0    0    0    0    0
0
##   47.3101923076923    0    0    0    0    0    0  0    0    0    0    0
0
##
##                   82.7 83 84 85 86 87 88 89
##   46.2265416666667    0  0  0  0  0  0  0  0
##   46.33425            0  0  0  0  0  0  0  0
##   46.4763425925926    0  0  0  0  0  0  0  0
##   46.6543137254902    0  0  0  0  0  0  0  0
##   47.1455729166667    0  0  0  0  0  0  0  0
##   47.3101923076923    0  0  0  0  0  0  0  0
```

```
print(rf3) #Prints the Random Forest
```

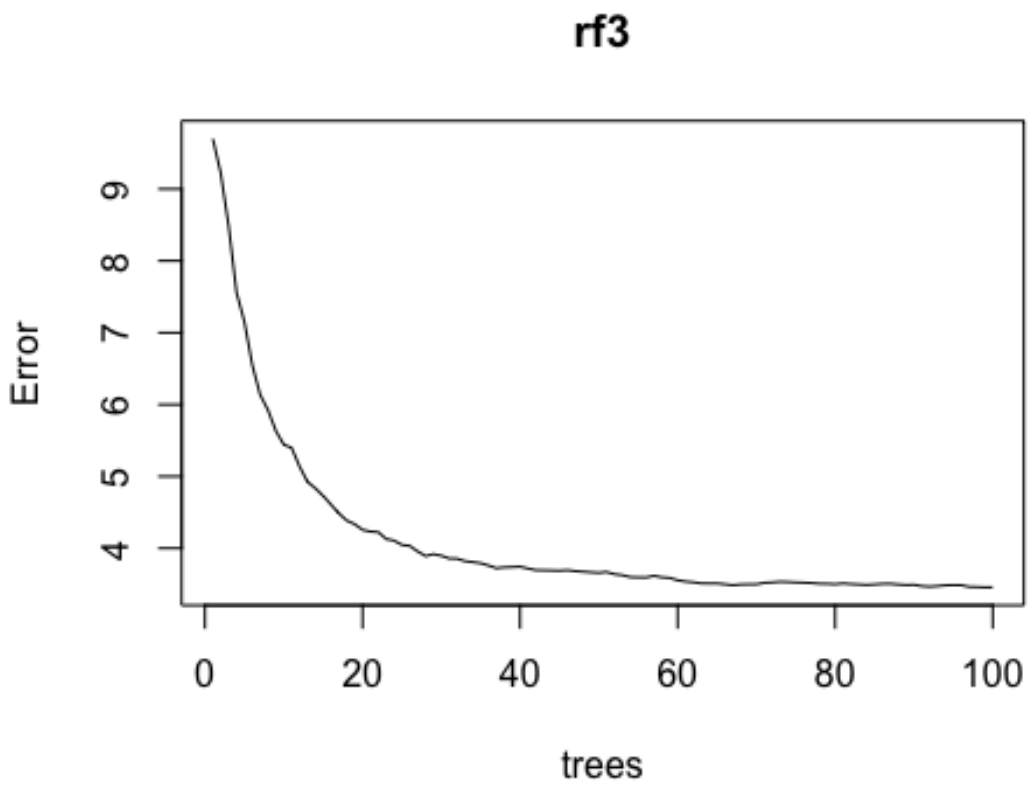
```
##
## Call:
##  randomForest(formula = Life.expectancy ~ ., data = train3Data,      ntree
= 100, proximity = TRUE)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 7
##
##           Mean of squared residuals: 3.458051
##                     % Var explained: 95.61
```

```
attributes(rf3)
```

```
## $names
##  [1] "call"            "type"            "predicted"       "mse"
##  [5] "rsq"             "oob.times"       "importance"      "importanceSD"
##  [9] "localImportance" "proximity"       "ntree"           "mtry"
## [13] "forest"          "coefs"           "y"               "test"
## [17] "inbag"           "terms"
```

```
##
## $class
## [1] "randomForest.formula" "randomForest"
```

```
plot(rf3) #Plots the Error Rate of Random Forest
```

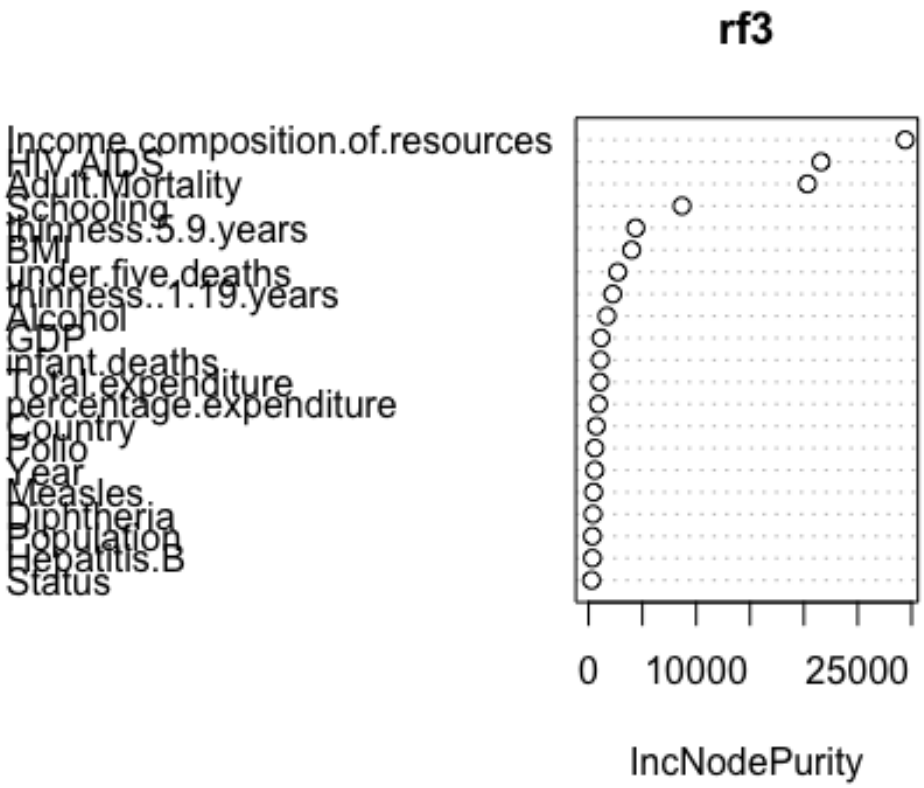



```
importance(rf3) #Prints the importance of variables
```

```
##                                 IncNodePurity
## Country                              717.6284
## Year                                 584.3725
## Status                               284.5567
## Adult.Mortality                    20326.1410
## infant.deaths                       1095.9369
## Alcohol                             1715.0193
## percentage.expenditure               930.0805
## Hepatitis.B                          361.7802
## Measles                              463.2605
## BMI                                 4004.9033
## under.five.deaths                   2708.4963
## Polio                                585.5303
## Total.expenditure                   1047.2764
## Diphtheria                           414.0251
## HIV.AIDS                           21589.1466
```

```
## GDP                                1161.6865
## Population                          362.5060
## thinness..1.19.years               2251.2863
## thinness.5.9.years                 4382.8626
## Income.composition.of.resources   29409.8582
## Schooling                          8679.6956
```

```r
varImpPlot(rf3) #Plots the variable importance
```




```r
LE_Pred3 <- predict(rf3, newdata=val3Data) #tests the random forest with test data
rftable <- table(LE_Pred3, val3Data$Life.expectancy) #checked with a table
head(rftable) #again because this table is extensive, we decided to display on the first six rows of the predictions table
```

```
##
## LE_Pred3           44.3 44.6 45.1 45.3 45.6 46 46.2 46.4 46.6 47.1 48.1 51
51.5
##   45.6343             1    0    0    0    0  0    0    0    0    0    0  0
0
##   46.6528166666667    0    0    0    1    0  0    0    0    0    0    0  0
0
##   47.6483166666666    0    0    0    0    0  0    1    0    0    0    0  0
0
```

```
##   48.0069333333333    0    0    0    0    0  0    0    1    0    0    0  0
0
##   48.0254             0    0    0    0    0  0    0    0    0    1    0  0
0
##   48.4367333333333    0    1    0    0    0  0    0    0    0    0    0  0
0
##
## LE_Pred3           52.1 52.3 52.6 53.6 54.1 54.8 54.9 55.3 55.6 55.8 56.1
56.5
##   45.6343             0    0    0    0    0    0    0    0    0    0    0
0
##   46.6528166666667    0    0    0    0    0    0    0    0    0    0    0
0
##   47.6483166666666    0    0    0    0    0    0    0    0    0    0    0
0
##   48.0069333333333    0    0    0    0    0    0    0    0    0    0    0
0
##   48.0254             0    0    0    0    0    0    0    0    0    0    0
0
##   48.4367333333333    0    0    0    0    0    0    0    0    0    0    0
0
##
## LE_Pred3           56.7 57.1 57.3 57.4 57.5 57.6 58 58.1 58.2 58.4 58.5
58.7
##   45.6343             0    0    0    0    0    0  0    0    0    0    0
0
##   46.6528166666667    0    0    0    0    0    0  0    0    0    0    0
0
##   47.6483166666666    0    0    0    0    0    0  0    0    0    0    0
0
##   48.0069333333333    0    0    0    0    0    0  0    0    0    0    0
0
##   48.0254             0    0    0    0    0    0  0    0    0    0    0
0
##   48.4367333333333    0    0    0    0    0    0  0    0    0    0    0
0
##
## LE_Pred3           58.8 58.9 59.1 59.2 59.5 59.7 59.9 61.2 61.3 61.5 61.7
61.9
##   45.6343             0    0    0    0    0    0    0    0    0    0    0
0
##   46.6528166666667    0    0    0    0    0    0    0    0    0    0    0
0
##   47.6483166666666    0    0    0    0    0    0    0    0    0    0    0
0
##   48.0069333333333    0    0    0    0    0    0    0    0    0    0    0
0
##   48.0254             0    0    0    0    0    0    0    0    0    0    0
0
##   48.4367333333333    0    0    0    0    0    0    0    0    0    0    0
```

```
0
##
## LE_Pred3           62 62.2 62.5 62.6 62.7 62.8 62.9 63 63.3 63.5 63.7 63.8
64
##   45.6343           0    0    0    0    0    0    0  0    0    0    0    0
0
##   46.6528166666667  0    0    0    0    0    0    0  0    0    0    0    0
0
##   47.6483166666666  0    0    0    0    0    0    0  0    0    0    0    0
0
##   48.0069333333333  0    0    0    0    0    0    0  0    0    0    0    0
0
##   48.0254           0    0    0    0    0    0    0  0    0    0    0    0
0
##   48.4367333333333  0    0    0    0    0    0    0  0    0    0    0    0
0
##
## LE_Pred3           64.2 64.3 64.6 64.7 64.8 64.9 65 65.1 65.3 65.5 65.6 66
66.2
##   45.6343             0    0    0    0    0    0  0    0    0    0    0  0
0
##   46.6528166666667    0    0    0    0    0    0  0    0    0    0    0  0
0
##   47.6483166666666    0    0    0    0    0    0  0    0    0    0    0  0
0
##   48.0069333333333    0    0    0    0    0    0  0    0    0    0    0  0
0
##   48.0254             0    0    0    0    0    0  0    0    0    0    0  0
0
##   48.4367333333333    0    0    0    0    0    0  0    0    0    0    0  0
0
##
## LE_Pred3           66.4 66.5 66.6 67.1 67.2 67.3 67.4 67.6 67.7 67.8 67.9
68
##   45.6343             0    0    0    0    0    0    0    0    0    0    0
0
##   46.6528166666667    0    0    0    0    0    0    0    0    0    0    0
0
##   47.6483166666666    0    0    0    0    0    0    0    0    0    0    0
0
##   48.0069333333333    0    0    0    0    0    0    0    0    0    0    0
0
##   48.0254             0    0    0    0    0    0    0    0    0    0    0
0
##   48.4367333333333    0    0    0    0    0    0    0    0    0    0    0
0
##
## LE_Pred3           68.3 68.5 68.7 68.8 68.9 69 69.1 69.3 69.4 69.5 69.6
69.8 70
##   45.6343             0    0    0    0    0  0    0    0    0    0    0
```

```
0  0
##   46.6528166666667    0    0    0    0    0  0    0    0    0    0    0
0  0
##   47.6483166666666    0    0    0    0    0  0    0    0    0    0    0
0  0
##   48.0069333333333    0    0    0    0    0  0    0    0    0    0    0
0  0
##   48.0254             0    0    0    0    0  0    0    0    0    0    0
0  0
##   48.4367333333333    0    0    0    0    0  0    0    0    0    0    0
0  0
##
## LE_Pred3           71 71.1 71.2 71.3 71.4 71.6 71.7 71.8 71.9 72 72.1 72.2
72.3
##   45.6343           0    0    0    0    0    0    0    0    0  0    0    0
0
##   46.6528166666667  0    0    0    0    0    0    0    0    0  0    0    0
0
##   47.6483166666666  0    0    0    0    0    0    0    0    0  0    0    0
0
##   48.0069333333333  0    0    0    0    0    0    0    0    0  0    0    0
0
##   48.0254           0    0    0    0    0    0    0    0    0  0    0    0
0
##   48.4367333333333  0    0    0    0    0    0    0    0    0  0    0    0
0
##
## LE_Pred3           72.4 72.5 72.6 72.7 72.8 72.9 73 73.1 73.2 73.3 73.4
73.5
##   45.6343             0    0    0    0    0    0  0    0    0    0    0
0
##   46.6528166666667    0    0    0    0    0    0  0    0    0    0    0
0
##   47.6483166666666    0    0    0    0    0    0  0    0    0    0    0
0
##   48.0069333333333    0    0    0    0    0    0  0    0    0    0    0
0
##   48.0254             0    0    0    0    0    0  0    0    0    0    0
0
##   48.4367333333333    0    0    0    0    0    0  0    0    0    0    0
0
##
## LE_Pred3           73.6 73.7 73.8 73.9 74 74.1 74.2 74.4 74.5 74.6 74.7
74.8
##   45.6343             0    0    0    0  0    0    0    0    0    0    0
0
##   46.6528166666667    0    0    0    0  0    0    0    0    0    0    0
0
##   47.6483166666666    0    0    0    0  0    0    0    0    0    0    0
0
```

```
##   48.0069333333333    0    0    0    0  0    0    0    0    0    0    0
0
##   48.0254             0    0    0    0  0    0    0    0    0    0    0
0
##   48.4367333333333    0    0    0    0  0    0    0    0    0    0    0
0
##
## LE_Pred3           74.9 75 75.1 75.2 75.3 75.4 75.5 75.6 75.8 75.9 76 76.1
76.3
##   45.6343             0  0    0    0    0    0    0    0    0    0  0    0
0
##   46.6528166666667    0  0    0    0    0    0    0    0    0    0  0    0
0
##   47.6483166666666    0  0    0    0    0    0    0    0    0    0  0    0
0
##   48.0069333333333    0  0    0    0    0    0    0    0    0    0  0    0
0
##   48.0254             0  0    0    0    0    0    0    0    0    0  0    0
0
##   48.4367333333333    0  0    0    0    0    0    0    0    0    0  0    0
0
##
## LE_Pred3           76.6 76.9 77.1 77.2 77.3 77.6 77.7 77.9 78 78.2 78.3
78.4
##   45.6343             0    0    0    0    0    0    0    0  0    0    0
0
##   46.6528166666667    0    0    0    0    0    0    0    0  0    0    0
0
##   47.6483166666666    0    0    0    0    0    0    0    0  0    0    0
0
##   48.0069333333333    0    0    0    0    0    0    0    0  0    0    0
0
##   48.0254             0    0    0    0    0    0    0    0  0    0    0
0
##   48.4367333333333    0    0    0    0    0    0    0    0  0    0    0
0
##
## LE_Pred3           78.5 78.7 79 79.3 79.4 79.6 79.9 80 81 81.1 81.2 81.4
81.5
##   45.6343             0    0  0    0    0    0    0  0  0    0    0    0
0
##   46.6528166666667    0    0  0    0    0    0    0  0  0    0    0    0
0
##   47.6483166666666    0    0  0    0    0    0    0  0  0    0    0    0
0
##   48.0069333333333    0    0  0    0    0    0    0  0  0    0    0    0
0
##   48.0254             0    0  0    0    0    0    0  0  0    0    0    0
0
##   48.4367333333333    0    0  0    0    0    0    0  0  0    0    0    0
```

```
0
##
## LE_Pred3           81.7 81.8 82.1 83 86 88
##   45.6343             0    0    0  0  0  0
##   46.6528166666667    0    0    0  0  0  0
##   47.6483166666666    0    0    0  0  0  0
##   48.0069333333333    0    0    0  0  0  0
##   48.0254             0    0    0  0  0  0
##   48.4367333333333    0    0    0  0  0  0

#Random forest with three independent variables for 2015
LE_2015_clean_3 <- LE_2015[!is.na(LE_2015$Adult.Mortality) &
                           !is.na(LE_2015$Income.composition.of.resources) &
                           !is.na(LE_2015$Schooling), ]  # Remove rows with
missing values for our variables
#We first have to split the data into two subsets: training (70%) and test
(30%)
ind <- sample(2, nrow(LE_2015_clean_3), replace=TRUE, prob=c(0.8, 0.2))
#makes two subsets
trainData <- LE_2015_clean_3[ind==1,] #80% for training subset
valData <- LE_2015_clean_3[ind==2,] #20% for validation subset
#install.packages('randomForest') #Loads in the package
library(randomForest)
rf <- randomForest(Life.expectancy ~
Adult.Mortality+Income.composition.of.resources+Schooling, data=trainData,
ntree=100, proximity=TRUE) #Predict Life.expectancy with the three selected
variables in 2015
rfpredtable2<-table(predict(rf), trainData$Life.expectancy) #Creates a table
for the predictions
head(rfpredtable2) #again because this table is extensive, we decided to
display on the first six rows of the predictions table

##
##                   51 52.4 52.5 53.1 53.7 54.5 57.3 57.6 58.2 58.3 58.9 59
59.6
##   55.9147154471545  0    0    1    0    0    0    0    0    0    0    0  0
0
##   56.1719230769231  0    0    0    0    0    0    0    1    0    0    0  0
0
##   56.2910215053763  0    0    0    0    0    0    0    0    0    1    0  0
0
##   57.2841228070175  0    0    0    1    0    0    0    0    0    0    0  0
0
##   57.6818686868687  0    0    0    0    0    0    0    0    0    0    0  0
0
##   57.9779333333333  1    0    0    0    0    0    0    0    0    0    0  0
0
##
##                   59.9 60 61.1 61.4 62.3 62.4 62.9 63.1 63.4 63.5 64.1
64.7
```

```
##   55.9147154471545    0  0    0    0    0    0    0    0    0    0    0
0
##   56.1719230769231    0  0    0    0    0    0    0    0    0    0    0
0
##   56.2910215053763    0  0    0    0    0    0    0    0    0    0    0
0
##   57.2841228070175    0  0    0    0    0    0    0    0    0    0    0
0
##   57.6818686868687    0  0    0    0    0    0    0    0    0    0    0
0
##   57.9779333333333    0  0    0    0    0    0    0    0    0    0    0
0
##
##                   64.8 65 65.5 65.7 65.8 66.2 66.3 66.4 66.6 66.7 67 67.5
68.3
##   55.9147154471545    0  0    0    0    0    0    0    0    0    0  0    0
0
##   56.1719230769231    0  0    0    0    0    0    0    0    0    0  0    0
0
##   56.2910215053763    0  0    0    0    0    0    0    0    0    0  0    0
0
##   57.2841228070175    0  0    0    0    0    0    0    0    0    0  0    0
0
##   57.6818686868687    0  0    0    0    0    0    0    0    0    0  1    0
0
##   57.9779333333333    0  0    0    0    0    0    0    0    0    0  0    0
0
##
##                   68.9 69.1 69.2 69.4 69.8 69.9 71 71.1 71.2 71.3 71.6
71.8
##   55.9147154471545    0    0    0    0    0    0  0    0    0    0    0
0
##   56.1719230769231    0    0    0    0    0    0  0    0    0    0    0
0
##   56.2910215053763    0    0    0    0    0    0  0    0    0    0    0
0
##   57.2841228070175    0    0    0    0    0    0  0    0    0    0    0
0
##   57.6818686868687    0    0    0    0    0    0  0    0    0    0    0
0
##   57.9779333333333    0    0    0    0    0    0  0    0    0    0    0
0
##
##                   71.9 72 72.3 72.7 73.2 73.3 73.5 73.6 73.9 74 74.1 74.3
74.5
##   55.9147154471545    0  0    0    0    0    0    0    0    0  0    0    0
0
##   56.1719230769231    0  0    0    0    0    0    0    0    0  0    0    0
0
##   56.2910215053763    0  0    0    0    0    0    0    0    0  0    0    0
```

```
0
##   57.2841228070175    0  0    0    0    0    0    0    0    0  0    0    0
0
##   57.6818686868687    0  0    0    0    0    0    0    0    0  0    0    0
0
##   57.9779333333333    0  0    0    0    0    0    0    0    0  0    0    0
0
##
##                   74.6 74.7 74.8 74.9 75 75.2 75.3 75.5 75.6 75.7 75.8 76
76.1
##   55.9147154471545    0    0    0    0  0    0    0    0    0    0    0  0
0
##   56.1719230769231    0    0    0    0  0    0    0    0    0    0    0  0
0
##   56.2910215053763    0    0    0    0  0    0    0    0    0    0    0  0
0
##   57.2841228070175    0    0    0    0  0    0    0    0    0    0    0  0
0
##   57.6818686868687    0    0    0    0  0    0    0    0    0    0    0  0
0
##   57.9779333333333    0    0    0    0  0    0    0    0    0    0    0  0
0
##
##                   76.2 76.4 76.6 76.7 76.9 77 77.4 77.5 77.6 77.7 77.8 78
79
##   55.9147154471545    0    0    0    0    0  0    0    0    0    0    0  0
0
##   56.1719230769231    0    0    0    0    0  0    0    0    0    0    0  0
0
##   56.2910215053763    0    0    0    0    0  0    0    0    0    0    0  0
0
##   57.2841228070175    0    0    0    0    0  0    0    0    0    0    0  0
0
##   57.6818686868687    0    0    0    0    0  0    0    0    0    0    0  0
0
##   57.9779333333333    0    0    0    0    0  0    0    0    0    0    0  0
0
##
##                   79.1 79.6 81 81.1 81.4 81.5 81.6 81.7 81.8 82 82.4 82.5
82.7
##   55.9147154471545    0    0  0    0    0    0    0    0    0  0    0    0
0
##   56.1719230769231    0    0  0    0    0    0    0    0    0  0    0    0
0
##   56.2910215053763    0    0  0    0    0    0    0    0    0  0    0    0
0
##   57.2841228070175    0    0  0    0    0    0    0    0    0  0    0    0
0
##   57.6818686868687    0    0  0    0    0    0    0    0    0  0    0    0
0
```

```
##   57.9779333333333    0    0  0    0    0    0    0    0    0  0    0    0
0
##
##                    82.8 83.1 83.4 83.7 85 86 88
##   55.9147154471545    0    0    0    0  0  0  0
##   56.1719230769231    0    0    0    0  0  0  0
##   56.2910215053763    0    0    0    0  0  0  0
##   57.2841228070175    0    0    0    0  0  0  0
##   57.6818686868687    0    0    0    0  0  0  0
##   57.9779333333333    0    0    0    0  0  0  0
```

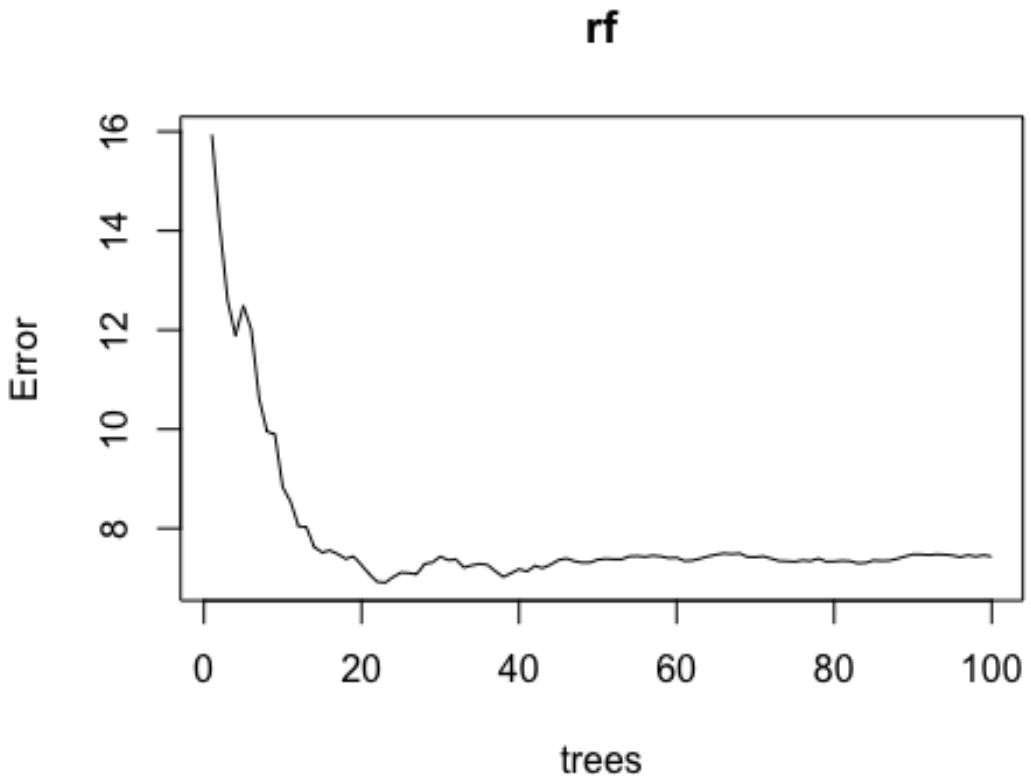
```
print(rf) #Prints the Random Forest
```

```
##
## Call:
##  randomForest(formula = Life.expectancy ~ Adult.Mortality +
Income.composition.of.resources +      Schooling, data = trainData, ntree =
100, proximity = TRUE)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 7.432409
##                     % Var explained: 88.79
```

```
attributes(rf)
```

```
## $names
##  [1] "call"            "type"            "predicted"       "mse"
##  [5] "rsq"             "oob.times"       "importance"      "importanceSD"
##  [9] "localImportance" "proximity"       "ntree"           "mtry"
## [13] "forest"          "coefs"           "y"               "test"
## [17] "inbag"           "terms"
##
## $class
## [1] "randomForest.formula" "randomForest"
```

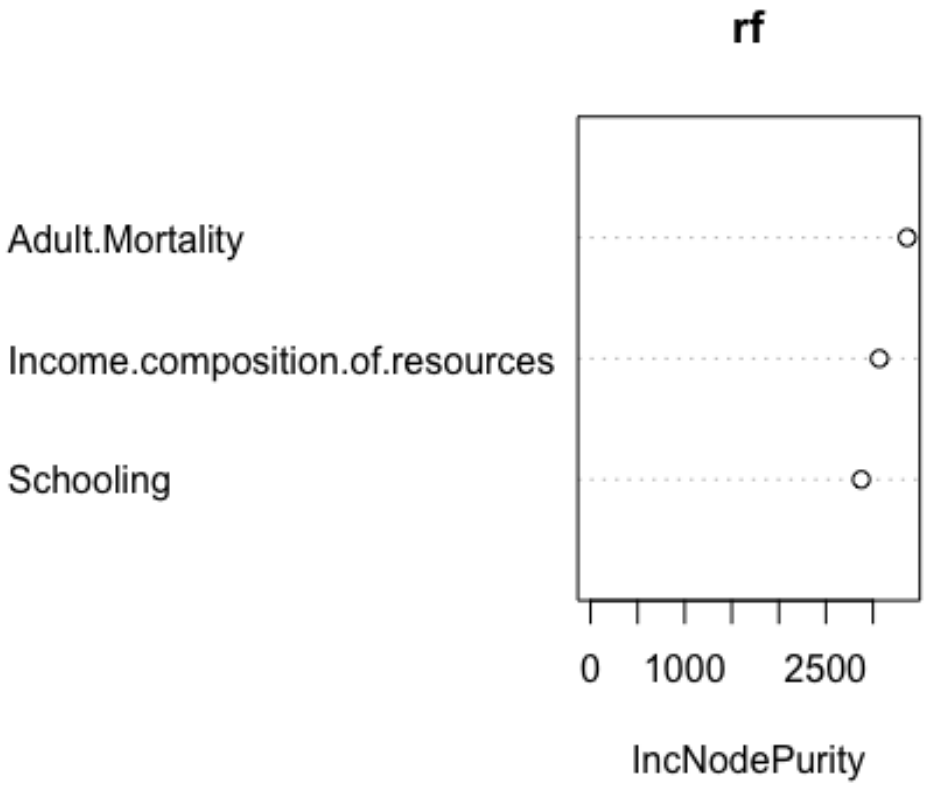
```
plot(rf) #Plots the Error Rate of Random Forest
```




```r
importance(rf) #Prints the importance of variables
```

```
##                                   IncNodePurity
## Adult.Mortality                        3363.888
## Income.composition.of.resources        3068.566
## Schooling                              2875.681
```

```r
varImpPlot(rf) #Plots the variable importance
```




```
LE_Pred <- predict(rf, newdata=valData) #tests the random forest with test
data
rftable2 <- table(LE_Pred, valData$Life.expectancy) #checked with a table
head(rftable2) #again because this table is extensive, we decided to display
on the first six rows of the predictions table

##
## LE_Pred            59.9 61.8 62.9 64.5 66 66.1 68.3 68.5 68.7 68.8 69.7 72
73.6
##   61.69955            1    0    0    0  0    0    0    0    0    0    0  0
0
##   61.7354833333333    0    0    0    1  0    0    0    0    0    0    0  0
0
##   62.0354166666667    0    1    0    0  0    0    0    0    0    0    0  0
0
##   62.4408             0    0    0    0  0    1    0    0    0    0    0  0
0
##   67.4782             0    0    1    0  0    0    0    0    0    0    0  0
0
##   67.7804166666667    0    0    0    0  1    0    0    0    0    0    0  0
0
##
## LE_Pred            74.4 74.5 74.6 74.8 74.9 76.3 77.1 78.2 78.5 81 81.1
```

```
81.9
##   61.69955            0    0    0    0    0    0    0    0    0  0    0
0
##   61.7354833333333    0    0    0    0    0    0    0    0    0  0    0
0
##   62.0354166666667    0    0    0    0    0    0    0    0    0  0    0
0
##   62.4408             0    0    0    0    0    0    0    0    0  0    0
0
##   67.4782             0    0    0    0    0    0    0    0    0  0    0
0
##   67.7804166666667    0    0    0    0    0    0    0    0    0  0    0
0
##
## LE_Pred             82.2 82.8
##   61.69955             0    0
##   61.7354833333333     0    0
##   62.0354166666667     0    0
##   62.4408              0    0
##   67.4782              0    0
##   67.7804166666667     0    0
```

#plot(margin(rf, valData$Life.expectancy)) would plot the margin of predictions, however, given our dependent variable and use of random forest, it does not produce anything. This is further examined in EX11.

EX10: Modelling: Classification: To group countries into developing and developed using Logistic Regression

```
LE_clean <- na.omit(Life_Expectancy) # Remove rows with missing values
LE_clean$develop <- ifelse(LE_clean$Status == "Developed", 1, 0) #sets status
to one if developed, zero otherwise
set.seed(1234) #ensure reproducibility
#We first have to split the data into two subsets: training (70%) and test
(30%)
ind2 <- sample(2, nrow(LE_clean), replace=TRUE, prob=c(0.8, 0.2)) #makes two
subsets
train2Data <- LE_clean[ind2==1,] #80% for training subset
val2Data <- LE_clean[ind2==2,] #20% for validation subset
mylogit <- glm(develop ~ Life.expectancy+GDP+Schooling, data=train2Data,
family="binomial") #creates Logit model
summary(mylogit) #produces summary of model created

##
## Call:
## glm(formula = develop ~ Life.expectancy + GDP + Schooling, family =
"binomial",
##     data = train2Data)
##
## Coefficients:
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.951e+01  1.896e+00 -10.290  < 2e-16 ***
## Life.expectancy  1.222e-01  2.853e-02   4.282 1.85e-05 ***
## GDP              2.315e-05  7.932e-06   2.919  0.00352 **
## Schooling        6.153e-01  7.480e-02   8.226  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1095.24  on 1320  degrees of freedom
## Residual deviance:  606.46  on 1317  degrees of freedom
## AIC: 614.46
##
## Number of Fisher Scoring iterations: 7
```

```
pred <- predict(mylogit, val2Data, type="response") #Makes a prediction for
the validation data with the logit model
model_pred_admit <- rep("0", 328) #Creates a set of 328 values, the number of
observations in valdata, all equal to "0"
model_pred_admit[pred>0.5] <- "1" #tells the model that when the prediction
is greater than 0.5, the country is developed, meaning equal to one
tab<-table(model_pred_admit, val2Data$develop) #creates a confusion matrix
comparing the predicted developed country and actual developed countries
print(tab) #Prints the confusion matrix
```

```
##
## model_pred_admit   0   1
##                0 270  25
##                1   8  25
```

```
1-sum(diag(tab))/sum(tab) #calculates the misclassification error of this
model
```

```
## [1] 0.1006098
```

V. Analysis:

EX11: Compare the outcomes of the Linear Regression classification techniques and those with the Random Forest classification techniques in EX8 and EX9.

With the linear regression technique, we explore the linear relationship between life expectancy in 2015 and three independent variables- adult mortality, income composition of resources, and schooling. The three models had statistically significant variables and relatively high R-squared values. The plots provided show a moderately positive correlation between life expectancy and income composition of resources as well as life expectancy and schooling. However, adult mortality seems to have a strong negative correlation.

Linear regression is useful in providing clear and interpretable relationships with one independent variable, but its inability to compare multiple independent variables at once, makes it not ideal for this dataset. With the random forest, we are still analyzing the same three independent variables. Because our dependent variable is numerical and continuous, this technique made many predictions that were hard to understand. However, the model itself had a 88.78 percent of variance explained, which makes it a good model with room for improvement. The error rate plot shows that the error rate decreases as we increase the number of trees up to 100. All of our independent variables were also identified as important, with adult mortality being highlighted as the most important one. We could not plot the margin of predictions because the model is used for regression, which involves predicting a numerical and continuous variable, and margin plots are typically used for classification tasks. While the linear regression provided better visuals and clarity, overall, the random forest was better accurately as it offered robust modeling for non-linear interactions and offered insights of multiple independent variables.

EX12: Interpretation and Conclusions: Analyze which features are contributing the most to Life Expectancy in EX9-EX10.

To determine the independent variables that contributed the most to Life Expectancy, the random forest's abilities to identify variables of importance was especially useful. We first ran the model with all of the variables in the dataset. Using the importance function and the variable importance plot, we assessed which variables were both logically and statistically valuable. We found adult mortality, income composition of resources, schooling to be the best. We reran the model with these select variables to help compare the results of the different techniques including the results of the linear regression. For the logistic model, we were analyzing the country status (develop/developing). We used the variables life expectancy, GDP, and schooling to make our model. All three variables were statistically significant and the model had a low misclassification error of approximately 10%. While this model was not used to specifically assess the life expectancy variable, we can see that it has much influence, as it is strongly associated, in the development of a country.

EX13: Conclusions: Draw any conclusions from the outcomes in EX8-EX10.

All of our models have consistently identified adult mortality, income composition of resources, and schooling as critical contributors to life expectancy. However, these may not be all the significant independent variables, more testing would be needed to determine whether there are more. As discussed in EX11, our linear regression models highlighted the correlations between each independent variable tested and life expectancy. As countries receive more resources and schooling services, life expectancy increases. It follows that if adults are increasingly dying, it strongly impacts the decrease of life expectancy. Given the complexity of the life expectancy variable, using a single

independent variable for predictions is not advised as seen by the decent but not significantly high R-squared values ranging from 61% to 82%. The random forest models then emphasised the importance of these variables in 2015 to life expectancy. It should be noted that EX8 and EX9 use the 2015 subset of the life expectancy dataset, which offers limited data that might not reflect patterns from previous years. Finally, with the logistic model, we analyzed variables that contribute to a country's development status. Developed countries were seen to be related to higher life expectancy, GDP, and schooling, as seen by the positive coefficients. While this model has a relatively high overall accuracy rate (90%), it also has a relatively high false positive rate (24%), meaning it mistakenly identifies countries that are developing as developed. Unlike EX8 and EX9, the logit model uses the entire life expectancy dataset. The additional data provided could help explain why the accuracy is the highest with this model.

Reflections

1. List three things you learned in completing your Project

(1) Data cleaning and preparation - learned how to handle missing values effectively using functions like na.omit and understand the importance of clean data for analysis. (2) Visualizing insights - developed skills in creating meaningful plots with ggplot2, such as scatter plots and histograms, and learned to annotate key points in the data for better understanding. (3) Gained experience in using Linear Regression, Random Forest and logistic regression to predict outcomes, and understood how to interpret model outputs, such as feature importance.

2. List three problems you encountered in completing your Project, and how did you resolve them

(1) The dataset had missing values that caused errors in calculations and model training. Used na.omit to remove rows with missing values and verified data completeness before proceeding with analysis. Then for tasks that required data from 2015, we had limited information, especially after removing rows with empty values. We fixed this by consulting with you first and either removing only empty values from select variables or restoring the larger dataset. (2) We had difficulty using ggplot, especially when it came to annotation and legends, as this is something we had not practiced before. Fortunately, the textbook provided a starting point and we were able to research details the function includes. (3) We also had issues with the models themselves. When we tried using all variables to run the model, we got NA's. We were able to whittle it down to a handful of variables by plotting each individual variable against the outcome variable and also doing individual models for various combinations of variables against the outcome variable. This allowed us to understand which variables were statistically significant enough to retain in their respective models and give us the desired output.

3. List one suggestion to improve the Project Providing a clear explanation (maybe a data dictionary) of each variable in the dataset would help avoid misinterpretation and ensure accurate calculations and visualizations.