DASC 5420 - Assignment 1

Pham Thi Thai - T00727094

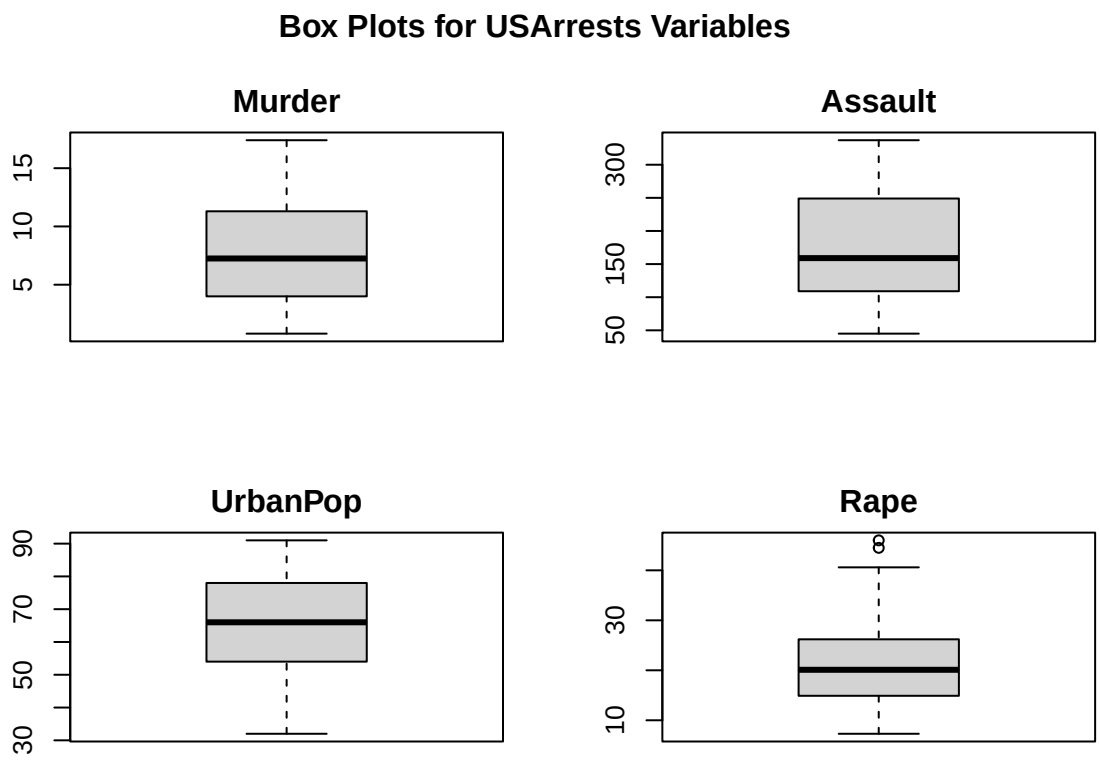
2024-02-12

```
library(tidyverse)
library(ggplot2)
library(tinytex)
```

1. USArrests

(a) Construct box plots for each of the four variables and comment on the univariate characteristics of each briefly.

```
data("USArrests")

# Create boxplots
par(mfrow = c(2, 2), oma = c(0, 2, 2, 0), mar = c(4, 4, 2, 1))

boxplot(USArrests$Murder, main = "Murder")
boxplot(USArrests$Assault, main = "Assault")
boxplot(USArrests$UrbanPop, main = "UrbanPop")
boxplot(USArrests$Rape, main = "Rape")

title("Box Plots for USArrests Variables", outer = TRUE, line = 1)
```




```
# Summary of univariate characteristics for each variable
data <- data.frame(
  Murder = USArrests$Murder,
  Assault = USArrests$Assault,
  UrbanPop = USArrests$UrbanPop,
  Rape = USArrests$Rape
)
summary_df <- summary(data)
print(summary_df)
```

```
##      Murder          Assault         UrbanPop          Rape
##  Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07
##  Median : 7.250   Median :159.0   Median :66.00   Median :20.10
##  Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18
##  Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00
```

The "Murder" variable represents the rate of murder arrests per 100,000 individuals, providing insights into crime prevalence. Murder demonstrates a symmetric distribution centered around 7.250, with a range from 0.800 to 17.400 and no outliers. "Assault" indicates the rate of assault arrests per 100,000, shedding light on violent crime patterns. Assault exhibits a right-skewed distribution, centered at 159.0, with a range from 45.0 to 337.0 and no outliers. "UrbanPop" is the percentage of urban population, reflecting the urbanization level across states. UrbanPop displays near symmetry around 66.00, with a range of 32.00 to 91.00 and no outliers. Lastly, the "Rape" variable represents the rate of rape arrests per 100,000, offering an understanding of sexual assault occurrences. Rape is symmetric, centered at 20.10, with a range from 7.30 to 46.00, and there are some outliers.

(b) Obtain the sample mean vector X; the sample covariance matrix S and the sample correlation matrix R: What can you say about the relationship between the four variables?

```
# Calculating the sample mean vector
mean_vector <- colMeans(USArrests)
cat("Sample Mean Vector (X):\n", mean_vector, "\n\n")
```

```
## Sample Mean Vector (X):
##  7.788 170.76 65.54 21.232
```

```
# Calculating the sample covariance matrix
cov_matrix <- cov(USArrests)
cat("Sample Covariance Matrix (S):\n")
```

```
## Sample Covariance Matrix (S):
```

```
print(cov_matrix)
```

```
##              Murder   Assault   UrbanPop      Rape
## Murder    18.970465  291.0624   4.386204  22.99141
## Assault  291.062367 6945.1657 312.275102 519.26906
## UrbanPop   4.386204  312.2751 209.518776  55.76808
## Rape      22.991412  519.2691  55.768082  87.72916
```

```
cat("\n")
```

```
# Calculating the sample correlation matrix
cor_matrix <- cor(USArrests)
cat("Sample Correlation Matrix (R):\n")
```

```
## Sample Correlation Matrix (R):
```

```
print(cor_matrix)
```

```
##              Murder   Assault   UrbanPop      Rape
## Murder   1.00000000 0.8018733 0.06957262 0.5635788
## Assault  0.80187331 1.0000000 0.25887170 0.6652412
## UrbanPop 0.06957262 0.2588717 1.00000000 0.4113412
## Rape     0.56357883 0.6652412 0.41134124 1.0000000
```

The analysis of covariance and correlation sheds light on the relationships between the variables in the dataset.

Covariance is a measure of how two variables change together. In our dataset, we observe positive covariances between Murder and Assault (291.06), Murder and Rape (22.99), and Assault and Rape (519.27), suggesting positive relationships between these variables.

Correlation, on the other hand, standardizes the measure of association between variables by scaling it to a range of -1 to 1. The correlation matrix reveals that Murder and Assault have a strong positive correlation (0.80), as do Assault and Rape (0.67), while Murder and Rape exhibit a moderate positive correlation (0.56). UrbanPop shows weak correlations with the crime variables, indicating a less pronounced linear relationship.

(c) Let $d_S(x, q) = \sqrt{(x-q)'S^{-1}(x-q)}$ where S is the sample covariance matrix, be the statistical distance between an observed point x and a fixed point q. Find the state with the highest murder rate and compute the statistical distance of each observation from that state.

```
# Define the stat.dist function
stat.dist <- function(Y, p) {
  S.inv <- solve(var(Y))
  dist <- function(y) sqrt(t(y - p) %*% S.inv %*% (y - p))
  return(apply(Y, 1, dist))
}

# Convert USArrests to a matrix
X <- as.matrix(USArrests)

# Find the state with the highest murder rate
state_with_highest_murder_rate <- rownames(X)[which.max(X[, "Murder"])]


# Extract the data for the state with the highest murder rate
state_data <- X[state_with_highest_murder_rate, ]

# Calculate the statistical distance from the state with the highest murder rate
statistical_distances <- stat.dist(X, state_data)

# Create a data frame for better presentation
result_df <- data.frame(State = rownames(X), Distance = statistical_distances)

# Determine the maximum length of State names
max_state_length <- max(nchar(result_df$State))

# Display the result
cat("State with the highest murder rate:", state_with_highest_murder_rate, "\n")
```

```
## State with the highest murder rate: Georgia
```

```
# Display the results of Statistical distance

for (i in 1:nrow(result_df)) {
  cat(result_df$State[i], strrep(" ", max_state_length - nchar(result_df$State[i]) + 1),
      result_df$Distance[i], "\n")
}
```

```
## Alabama          2.237537
## Alaska           5.353208
## Arizona          5.014492
## Arkansas         3.289001
## California       4.690648
## Colorado         4.16002
## Connecticut      4.174242
## Delaware         5.061498
## Florida          3.192869
## Georgia          0
## Hawaii           3.347875
```

```
## Idaho            4.517524
## Illinois         3.501859
## Indiana          2.672336
## Iowa             3.863734
## Kansas           3.061579
## Kentucky         1.887496
## Louisiana        1.699453
## Maine            4.317894
## Maryland         3.992429
## Massachusetts    4.290383
## Michigan         2.990537
## Minnesota        3.799544
## Mississippi      2.453937
## Missouri         2.867506
## Montana          3.12813
## Nebraska         3.546803
## Nevada           3.781271
## New Hampshire    3.926579
## New Jersey       3.553152
## New Mexico       3.663327
## New York         3.367715
## North Carolina   5.005631
## North Dakota     4.481025
## Ohio             2.782045
## Oklahoma         3.264305
## Oregon           4.307186
## Pennsylvania     2.972147
## Rhode Island     5.396792
## South Carolina   2.952706
## South Dakota     3.850774
## Tennessee        1.457513
## Texas            1.997121
## Utah             4.253167
## Vermont          4.482377
## Virginia         2.63066
## Washington       4.296123
## West Virginia    3.341547
## Wisconsin        3.690465
## Wyoming          3.421648
```

(d) Using these computed distances determine which six states are "closest" to the state with the highest murder rate. Can you provide an explanation why these seven states are statistically similar in terms of their violent crime rates and percentage of urban population in 1973?

```r
# Sort the data frame by statistical distances
sorted_result_df <- result_df[order(result_df$Distance), ]

# Select the top six closest states
closest_states <- sorted_result_df[2:7, ]

# Display the closest states
closest_states
```

```
##                   State Distance
## Tennessee     Tennessee 1.457513
## Louisiana     Louisiana 1.699453
## Kentucky       Kentucky 1.887496
## Texas             Texas 1.997121
## Alabama         Alabama 2.237537
## Mississippi Mississippi 2.453937
```

Using statistic summary to explain

```
# Subset the data for the seven states
selected_states <- c("Georgia", "Tennessee", "Louisiana", "Kentucky", "Texas",
                     "Alabama", "Mississippi")
subset_data <- USArrests[selected_states, ]

# Display the subset of the data
subset_data
```

```
##             Murder Assault UrbanPop Rape
## Georgia       17.4     211       60 25.8
## Tennessee     13.2     188       59 26.9
## Louisiana     15.4     249       66 22.2
## Kentucky       9.7     109       52 16.3
## Texas         12.7     201       80 25.5
## Alabama       13.2     236       58 21.2
## Mississippi   16.1     259       44 17.1
```

```
# Summary statistics for each variable
summary_stats <- apply(subset_data, 2, summary)
print(summary_stats)
```

```
##           Murder  Assault UrbanPop     Rape
## Min.     9.70000 109.0000 44.00000 16.30000
## 1st Qu. 12.95000 194.5000 55.00000 19.15000
## Median  13.20000 211.0000 59.00000 22.20000
## Mean    13.95714 207.5714 59.85714 22.14286
## 3rd Qu. 15.75000 242.5000 63.00000 25.65000
## Max.    17.40000 259.0000 80.00000 26.90000
```

```
# Calculate the Correlation matrix for subset data
correlation_matrix <- cor(subset_data)
cat("Correlation Matrix for subset data of 7 states:\n")
```

```
## Correlation Matrix for subset data of 7 states:
```

```
print(correlation_matrix)
```

```
##               Murder    Assault    UrbanPop      Rape
## Murder    1.00000000 0.77623391 -0.08344099 0.3014246
## Assault   0.77623391 1.00000000  0.01334624 0.1475061
## UrbanPop -0.08344099 0.01334624  1.00000000 0.6707549
## Rape      0.30142459 0.14750609  0.67075493 1.0000000
```

The seven states examined – Georgia, Tennessee, Louisiana, Kentucky, Texas, Alabama, and Mississippi – demonstrate statistical similarities in their crime rates and urban population percentages for the year 1973. Notably, the mean murder rate across these states is approximately 13.96, with individual states exhibiting rates ranging from 9.7 to 17.4. The mean assault rate is around 207.57, varying between 109 and 259 across states. Similarly, the mean rape rate hovers at approximately 22.14, with state-specific rates ranging from

16.3 to 26.9.

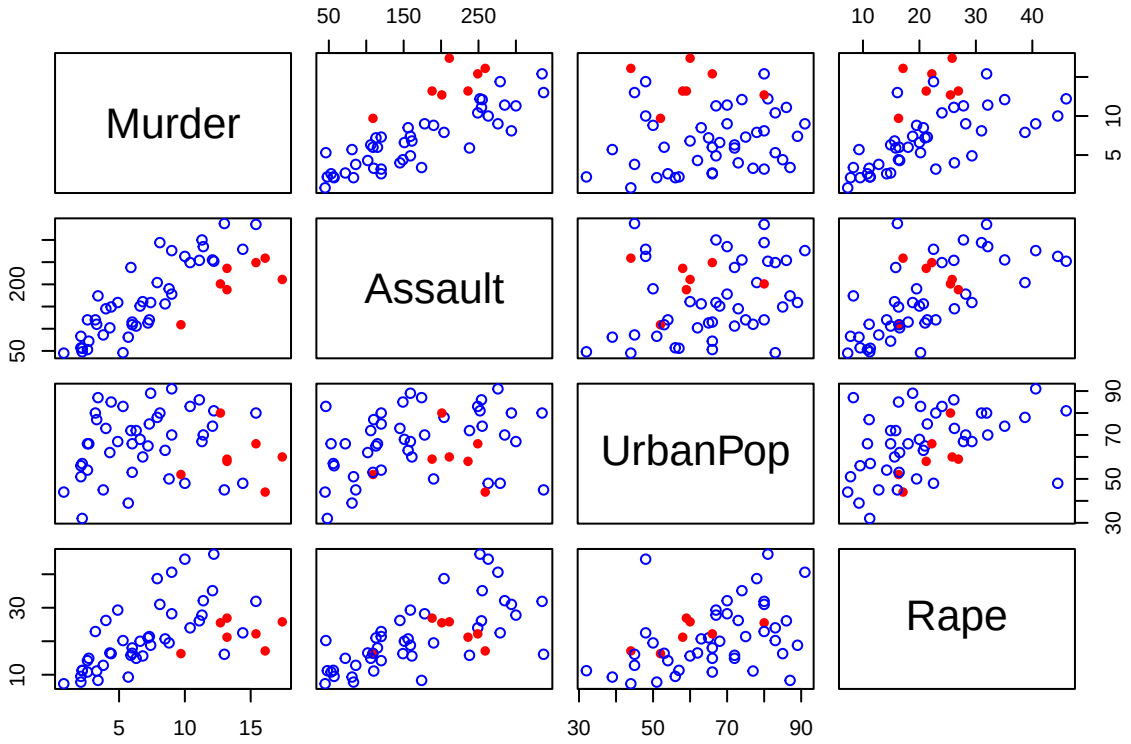
In terms of urban population, these states share a commonality, with an average percentage of around 59.86%. Individual states range from 44% to 80% in terms of urbanization, reflecting a relatively consistent urban population distribution.

Based on the statistical summary and correlation matrix provided, it is evident that there are significant relationships among crime rates and urbanization across the seven selected states. Specifically, a strong positive correlation exists between Murder and Assault rates, indicating that states with higher murder rates tend to experience elevated assault rates as well. Additionally, there is a moderate positive correlation between Murder and Rape rates, implying a potential overlap in the prevalence of these violent crimes. Interestingly, urbanization appears to have minimal influence on murder rates, as indicated by a weak negative correlation between Murder rate and Urban Population percentage. Conversely, there is a strong positive correlation between Rape rate and Urban Population percentage, suggesting that states with higher urban populations tend to report higher rape rates.

The six states have relatively small statistical distances from the state with the highest murder rate, Georgia. This suggests that they share similarities in murder rates, socio-economic factors, geographical proximity, common challenges, law enforcement strategies, and historical context with the state with the highest murder rate. These factors collectively contribute to a closer resemblance in crime rates among these states compared to others.

(e) Obtain pairwise scatterplots (also called matrixplot) for the four variables in R and identify (using a different color or symbol) the points for the seven states you found in part (d). Do what you see on the scatterplots and what you concluded in part (d) agree with each other?

```r
# Create a color vector for the points
point_colors <- ifelse(rownames(USArrests) %in% selected_states, "red", "blue")

# Create a matrix plot with customized points
pairs(USArrests, col = point_colors, pch = ifelse(rownames(USArrests) %in% selected_states, 16, 1))
```



Based on the scatterplot matrix analysis, it is evident that strong correlations exist between certain variables among the seven selected states. Specifically, there is a pronounced positive association between Murder and Assault rates, as indicated by the clustering of points towards one side of the scatterplot. This suggests that states with higher murder rates also tend to experience elevated assault rates. Additionally, while a positive correlation is observed between Murder and Rape rates, it is not as pronounced as Murder and Assault. Points on the scatterplot between Murder and Rape are more dispersed, yet still exhibit some concentration in certain areas, indicating a moderate positive association between these variables.

Interestingly, no clear correlation is evident between Murder and Urban Population, suggesting that urbanization may not have a significant impact on murder rates among the states. Similarly, there is no strong correlation between Assault and Urban Population, with points widely scattered on the scatterplot between these variables, lacking a distinct trend.

Conversely, a strong positive correlation is identified between Rape and Urban Population, with points concentrated at high values for both variables. This underscores the significant influence of urbanization on rape rates among the states, highlighting the complex interplay between urbanization dynamics and crime patterns.

Overall, these observations from the scatterplot matrix analysis align with previous conclusions regarding the correlations between variables and the impact of urbanization on crime rates across the states.

2. WHO Life Expectancy

(a) Design a hypothesis for something you can predict from this data set using Linear Regression. Write your hypothesis and provide justification. Include an explanation of why you think you will be able to predict something and list the inputs and output.

```
# Load the dataset
data <- read.csv("Life Expectancy Data.csv")
head(data)
```

```
##       Country Year     Status Life.expectancy Adult.Mortality infant.deaths
## 1 Afghanistan 2015 Developing            65.0             263            62
## 2 Afghanistan 2014 Developing            59.9             271            64
## 3 Afghanistan 2013 Developing            59.9             268            66
## 4 Afghanistan 2012 Developing            59.5             272            69
## 5 Afghanistan 2011 Developing            59.2             275            71
## 6 Afghanistan 2010 Developing            58.8             279            74
##   Alcohol percentage.expenditure Hepatitis.B Measles  BMI under.five.deaths
## 1    0.01              71.279624          65    1154 19.1                83
## 2    0.01              73.523582          62     492 18.6                86
## 3    0.01              73.219243          64     430 18.1                89
## 4    0.01              78.184215          67    2787 17.6                93
## 5    0.01               7.097109          68    3013 17.2                97
## 6    0.01              79.679367          66    1989 16.7               102
##   Polio Total.expenditure Diphtheria HIV.AIDS       GDP Population
## 1     6              8.16         65      0.1 584.25921   33736494
## 2    58              8.18         62      0.1 612.69651     327582
## 3    62              8.13         64      0.1 631.74498   31731688
## 4    67              8.52         67      0.1 669.95900    3696958
## 5    68              7.87         68      0.1  63.53723    2978599
## 6    66              9.20         66      0.1 553.32894    2883167
##   thinness..1.19.years thinness.5.9.years Income.composition.of.resources
## 1                 17.2               17.3                           0.479
## 2                 17.5               17.5                           0.476
## 3                 17.7               17.7                           0.470
## 4                 17.9               18.0                           0.463
## 5                 18.2               18.2                           0.454
## 6                 18.4               18.4                           0.448
##   Schooling
## 1      10.1
## 2      10.0
## 3       9.9
## 4       9.8
## 5       9.5
## 6       9.2
```

Recommend a hypothesis and justification

We hypothesize that life expectancy can be predicted based on a combination of health factors and economic indicators using linear regression. Specifically, we expect that factors such as adult mortality rate, BMI (Body Mass Index), GDP (Gross Domestic Product), percentage expenditure, schooling, and alcohol consumption, in addition to the status of a country (developed or developing), can collectively influence life expectancy.

- Health Factors: Variables such as adult mortality rate and BMI are widely recognized as significant

determinants of life expectancy. Higher adult mortality rates and unhealthy BMI levels are generally associated with lower life expectancy.

- Economic Indicators: Economic indicators like GDP, income composition of resources, and schooling are also influential factors. Higher GDP typically correlates with better healthcare infrastructure and access to resources, which can positively impact life expectancy. Similarly, a higher level of schooling may lead to greater awareness and adoption of healthier lifestyle choices, contributing to higher life expectancy.
- Alcohol Consumption: Alcohol consumption is a known risk factor for various health issues, including liver disease, cardiovascular disease, and certain cancers. Thus, higher levels of alcohol consumption may negatively impact life expectancy.
- Country Status: The status of a country (developed or developing) may also play a role in life expectancy, as developed countries generally have better healthcare systems, higher standards of living, and access to resources compared to developing countries.
- Percentage expenditure: represents the proportion of a country's total expenditure allocated to healthcare. It provides insights into the level of investment in healthcare infrastructure and services within a country. By incorporating this variable, we can assess the impact of healthcare spending on life expectancy, alongside other health-related and socioeconomic factors.

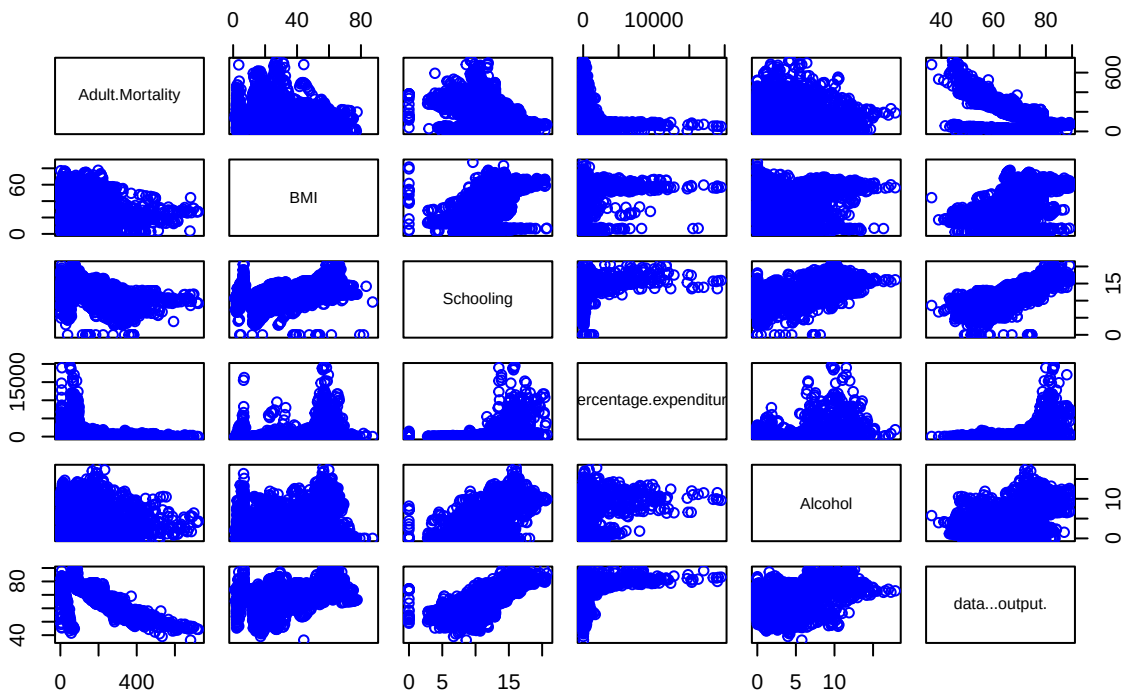
Inputs: Status (Developed or Developing), Adult mortality rate, BMI, GDP, Schooling, Percentage expenditure, Alcohol consumption

Output: Life expectancy

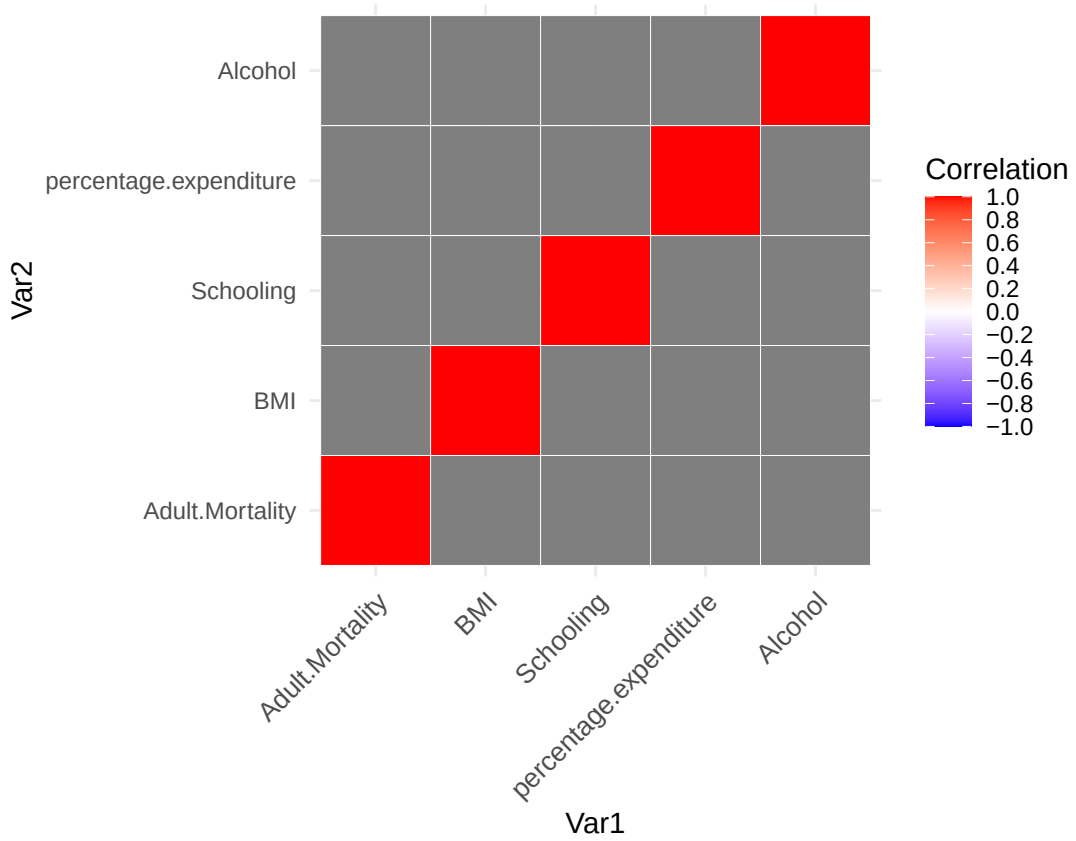
Explanation: Given the significant influence of health factors and economic indicators on life expectancy, it is reasonable to assume that a linear regression model can effectively predict life expectancy based on these variables. By analyzing data from 193 countries, we can identify patterns and relationships between the input variables and life expectancy, allowing us to develop a predictive model that accurately estimates life expectancy for different countries.

Visualization

```
#Visualization
inputs <- c("Adult.Mortality", "BMI", "Schooling", "percentage.expenditure", "Alcohol")
output <- "Life.expectancy"

# Create a matrix scatter plot
pairs(data.frame(data[, inputs], data[, output]), main = "Scatter Plot Matrix", col="blue")
```




```r
library(reshape2)

# Compute the correlation matrix
correlation_matrix <- cor(data[, inputs])

# Reshape the correlation matrix for visualization
melted_correlation <- melt(correlation_matrix)

# Find the absolute maximum correlation coefficient
max_correlation <- max(abs(correlation_matrix), na.rm = TRUE)

# Create a heatmap
ggplot(data = melted_correlation, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name="Correlation", n.breaks = 10) +  # Adjust the number of breaks
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 10, hjust = 1)) +
  coord_fixed()
```




```
# Create violin plots for each input variable with color by output variable
plots <- lapply(inputs, function(var) {
  ggplot(data, aes_string(x = output, y = var, fill = output)) +
    geom_violin() +
    labs(x = output, y = var) +
    theme_minimal() +
    scale_fill_gradient(low = "blue", high = "red") +
    theme(legend.position = "none")
})
```

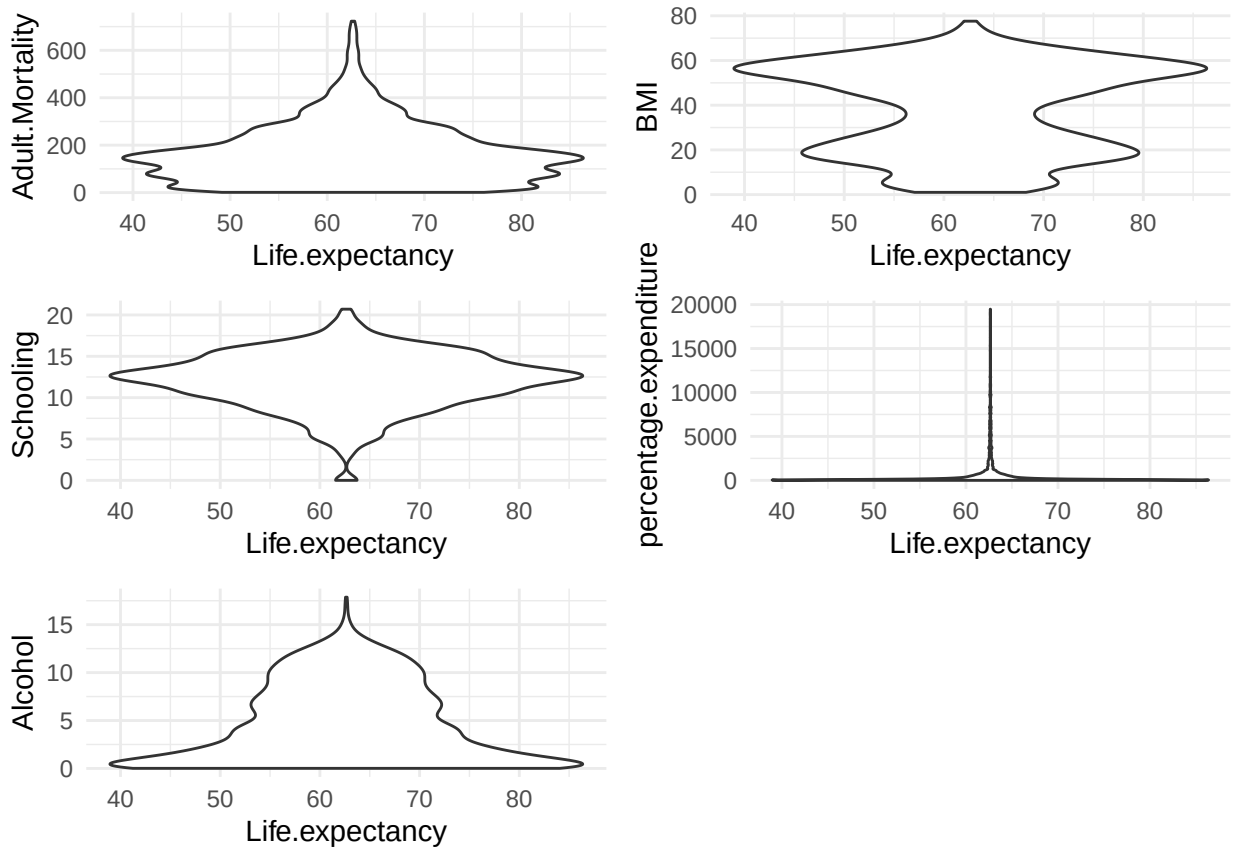
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# Arrange the violin plots in a grid
gridExtra::grid.arrange(grobs = plots, ncol = 2)
```

```
## Warning: Removed 10 rows containing non-finite values (`stat_ydensity()`).
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
## Warning: Removed 42 rows containing non-finite values (`stat_ydensity()`).

## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 170 rows containing non-finite values (`stat_ydensity()`).

## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 10 rows containing non-finite values (`stat_ydensity()`).

## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 203 rows containing non-finite values (`stat_ydensity()`).

## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```r
# Create histograms for each input variable with color by output variable
plots <- lapply(inputs, function(var) {
  ggplot(data, aes_string(x = var, fill = output)) +
    geom_histogram(binwidth = 1, position = "identity") +
    labs(x = var, y = "Frequency") +
    theme_minimal() +
    scale_fill_gradient(low = "blue", high = "red") +
    theme(legend.position = "none")
})
# Arrange the histograms in a grid
gridExtra::grid.arrange(grobs = plots, ncol = 2)
```

```
## Warning: Removed 10 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
## Warning: Removed 34 rows containing non-finite values (`stat_bin()`).
```

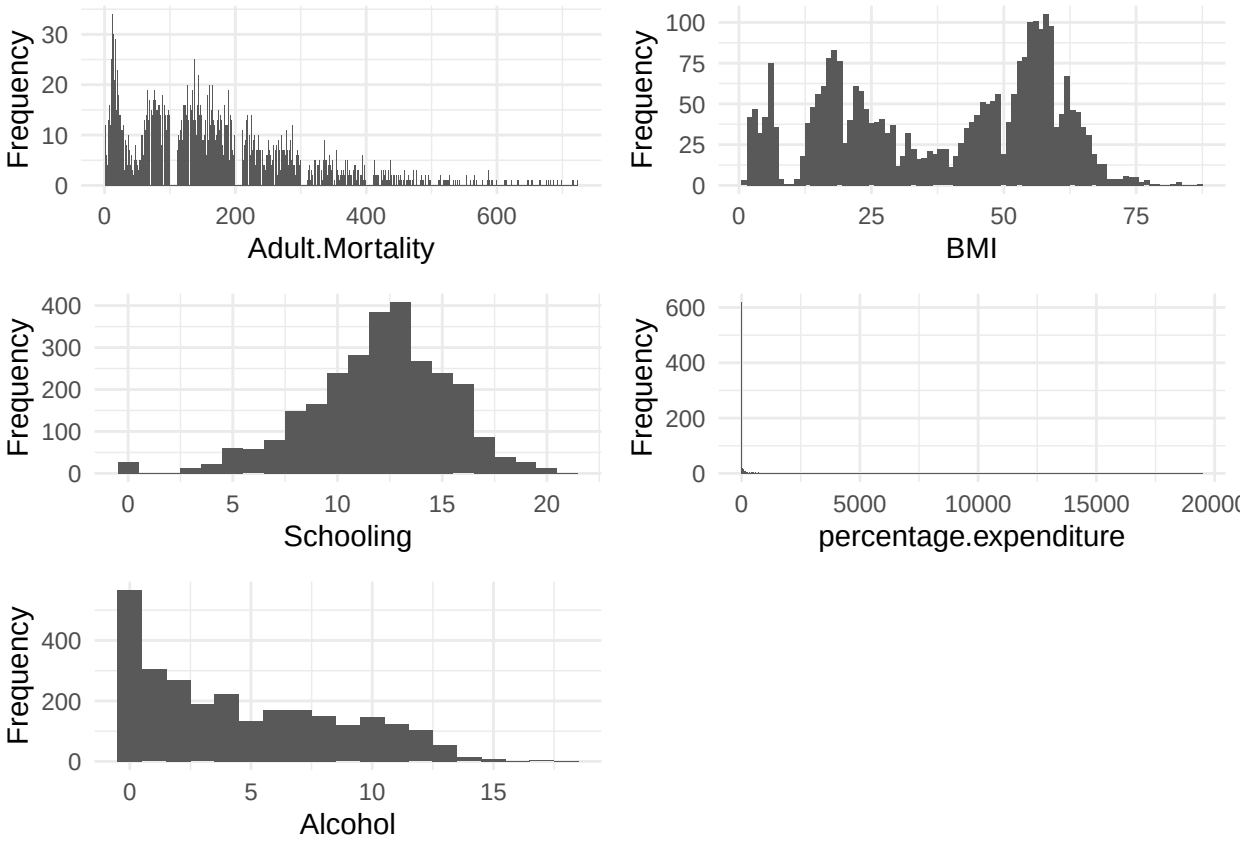
```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
## Warning: Removed 163 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
## Warning: Removed 194 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



(b) Process any necessary data transformation. Explain why you are using that transformation

Check missing values

```
# Subset data
subset_data <- data[, c(inputs, output)]
```

```r
# Check for missing values
missing_values <- colSums(is.na(subset_data))
print("Missing values:")
```

```
## [1] "Missing values:"
```

```r
print(missing_values)
```

```
##        Adult.Mortality                    BMI              Schooling
##                     10                     34                    163
## percentage.expenditure                Alcohol        Life.expectancy
##                      0                    194                     10
```

```r
# Check for missing values in the "Status" variable
missing_status <- sum(is.na(subset_data$Status))
print("Missing_status:")
```

```
## [1] "Missing_status:"
```

```r
print(missing_status)
```

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

```r
# Print summary statistics
print("Summary statistics:")
```

```
## [1] "Summary statistics:"
```

```r
print(summary(subset_data))
```

```
##  Adult.Mortality      BMI          Schooling     percentage.expenditure
##  Min.   :  1.0   Min.   : 1.00   Min.   : 0.00   Min.   :    0.000
##  1st Qu.: 74.0   1st Qu.:19.30   1st Qu.:10.10   1st Qu.:    4.685
##  Median :144.0   Median :43.50   Median :12.30   Median :   64.913
##  Mean   :164.8   Mean   :38.32   Mean   :11.99   Mean   :  738.251
##  3rd Qu.:228.0   3rd Qu.:56.20   3rd Qu.:14.30   3rd Qu.:  441.534
##  Max.   :723.0   Max.   :87.30   Max.   :20.70   Max.   :19479.912
##  NA's   :10      NA's   :34      NA's   :163
##     Alcohol        Life.expectancy
##  Min.   : 0.0100   Min.   :36.30
##  1st Qu.: 0.8775   1st Qu.:63.10
##  Median : 3.7550   Median :72.10
##  Mean   : 4.6029   Mean   :69.22
##  3rd Qu.: 7.7025   3rd Qu.:75.70
##  Max.   :17.8700   Max.   :89.00
##  NA's   :194       NA's   :10
```

Handle missing value by imputation method and feature scaling

```r
library(mice)
# Select relevant columns
selected_cols <- c("Status", "Adult.Mortality", "BMI", "GDP", "Schooling",
                   "percentage.expenditure", "Alcohol", "Life.expectancy")

# Subset the data
selected_data <- data[, selected_cols]

# Perform mice imputation
```

```r
imputed_data <- mice(selected_data, method = "pmm", m = 5)
```

```
##
##  iter imp variable
##   1   1  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   1   2  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   1   3  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   1   4  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   1   5  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   2   1  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   2   2  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   2   3  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   2   4  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   2   5  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   3   1  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   3   2  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   3   3  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   3   4  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   3   5  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   4   1  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   4   2  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   4   3  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   4   4  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   4   5  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   5   1  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   5   2  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   5   3  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   5   4  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
##   5   5  Adult.Mortality  BMI  GDP  Schooling  Alcohol  Life.expectancy
```

```
## Warning: Number of logged events: 1
```

```r
# Complete the imputation
imputed_data <- complete(imputed_data)

# Feature scaling function
feature_scaling <- function(x) {
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}

# Apply feature scaling to selected columns
scaled_data <- as.data.frame(lapply
(imputed_data[, c("Adult.Mortality", "BMI", "GDP", "Schooling", "Alcohol",
                  "percentage.expenditure")], feature_scaling))


# Create dummy variables for 'Status'
dummy_status <- model.matrix(~ Status - 1, data = imputed_data)

# Combine scaled data with dummy variables
scaled_data <- cbind(dummy_status, scaled_data)

# Add 'Life.expectancy' column
scaled_data$Life.expectancy <- imputed_data$Life.expectancy
```

The transformations applied serve two main purposes: feature scaling and dummy coding.

- **Feature Scaling**:

Feature scaling standardizes the range of the features so that they have a mean of zero and a standard deviation of one. This is important because features with different scales may lead to biased or inefficient models. By scaling the features, we ensure that each feature contributes proportionately to the model's learning process, preventing certain features from dominating due to their larger scales. In the provided code, the feature_scaling function is applied to each numeric feature column (Adult.Mortality, BMI, GDP, Schooling, percentage.expenditure, Alcohol) to standardize their values.

- **Dummy Coding for Status**:

The variable "Status" is categorical, with two levels: "Developed" and "Developing." To include this categorical variable in the regression model, we need to convert it into a numeric format. Dummy coding achieves this by creating binary indicator variables for each level of the categorical variable. In the provided code, the model.matrix function generates dummy variables for the "Status" column, resulting in two new binary columns: "Developed" and "Developing." This allows us to incorporate the categorical information into the regression model without assuming any ordinal relationship between the categories.

(c) Perform a Linear Regression and comment on the output.

```
# Check for missing values in scaled_data
missing_values <- colSums(is.na(scaled_data))
print("Missing values in scaled_data:")
```

```
## [1] "Missing values in scaled_data:"
```

```
print(missing_values)
```

```
##  StatusDeveloped StatusDeveloping  Adult.Mortality              BMI
##                0                0                0                0
##              GDP        Schooling          Alcohol  Life.expectancy
##                0                0                0                0
```

```
# Perform Linear Regression without one of the dummy variables for 'Status'
model <- lm(Life.expectancy ~ . - StatusDeveloping, data = scaled_data)

# Summary of the Linear Regression Model
summary(model)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ . - StatusDeveloping, data = scaled_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.3830  -2.3058   0.2831   2.9847  22.3567
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             68.9454     0.1052 655.474  < 2e-16 ***
## StatusDeveloped          1.5132     0.3190   4.744  2.2e-06 ***
## Adult.Mortality         -3.8106     0.1045 -36.472  < 2e-16 ***
## BMI                      1.2190     0.1097  11.107  < 2e-16 ***
## GDP                      0.6956     0.2019   3.446 0.000578 ***
## Schooling                4.2567     0.1336  31.857  < 2e-16 ***
```

```
## Alcohol                 -0.1890     0.1193  -1.585 0.113125
## percentage.expenditure   0.0172     0.1984   0.087 0.930924
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.84 on 2930 degrees of freedom
## Multiple R-squared:  0.7429, Adjusted R-squared:  0.7422
## F-statistic:  1209 on 7 and 2930 DF,  p-value: < 2.2e-16
```

```
# Generate predicted values using the model
predicted <- predict(model, scaled_data)

# Check for missing or invalid values in actual and predicted values
missing_actual <- sum(is.na(scaled_data$Life.expectancy))
missing_predicted <- sum(is.na(predicted))

# Print the number of missing or invalid values
print(paste("Missing or invalid values in actual:", missing_actual))
```

```
## [1] "Missing or invalid values in actual: 0"
```

```
print(paste("Missing or invalid values in predicted:", missing_predicted))
```

```
## [1] "Missing or invalid values in predicted: 0"
```

Interpretation

The linear regression model aimed to predict life expectancy based on various predictors including adult mortality rate, BMI, GDP, schooling, percentage expenditure, and alcohol consumption. Here's a brief interpretation of the results:

Intercept: The intercept term is statistically significant ($p < 0.001$), indicating that when all independent variables are zero, the estimated life expectancy is approximately 68.93 years.
Status (Developed): The coefficient estimate for the developed status variable is 1.7161 ($p < 0.001$), indicating that developed countries tend to have higher life expectancy compared to developing countries.
Adult Mortality: The coefficient estimate for adult mortality is -3.7790 ($p < 0.001$), suggesting that an increase in adult mortality is associated with a decrease in life expectancy.
BMI: The coefficient estimate for BMI is 1.2856 ($p < 0.001$), indicating that higher BMI is positively associated with life expectancy.
GDP: The coefficient estimate for GDP is 0.6965 ($p = 0.00102$), suggesting that higher GDP is associated with higher life expectancy.
Schooling: The coefficient estimate for schooling is 4.2393 ($p < 0.001$), indicating that higher levels of schooling are associated with higher life expectancy.
Alcohol: The coefficient estimate for alcohol consumption is -0.2502 ($p = 0.03454$), suggesting a negative association between alcohol consumption and life expectancy, although the effect is relatively weak.
Percentage Expenditure: The coefficient estimate for percentage expenditure is -0.0139 ($p = 0.94696$), indicating that percentage expenditure is not significantly associated with life expectancy.

Model Evaluation: The overall model fit is significant ($p < 0.001$), indicating that the predictors collectively explain a significant proportion of the variance in life expectancy.

(d) Print out your algorithm performance. Choose the right metric(s) for judging the effectiveness of your prediction

```
# 1. Cross-Validation
library(caret)
trainControl <- trainControl(method = "cv", number = 5)
```

```r
model <- train(Life.expectancy ~ ., data = scaled_data, method = "lm", trControl = trainControl)
cv_results <- model$resample
print(cv_results)
```

```
##       RMSE  Rsquared      MAE Resample
## 1 4.790177 0.7498351 3.364416    Fold1
## 2 4.873263 0.7368385 3.608100    Fold2
## 3 5.056381 0.7092715 3.633531    Fold3
## 4 4.499897 0.7824132 3.292247    Fold4
## 5 4.968509 0.7313806 3.432290    Fold5
```

```r
# 2. Mean Absolute Error (MAE)
mae <- mean(abs(cv_results$MAE))

# 3. Mean Squared Error (MSE)
mse <- mean(cv_results$RMSE^2)

# 4. Root Mean Squared Error (RMSE)
rmse <- sqrt(mse)

# 5. R-squared (R^2)
r_squared <- summary(model)$r.squared

# 6. Adjusted R-squared
adjusted_r_squared <- summary(model)$adj.r.squared

# 7. F-statistic
f_statistic <- summary(model)$fstatistic[1]

# Print out the results
print(paste("Mean Absolute Error (MAE):", mae))
```

```
## [1] "Mean Absolute Error (MAE): 3.46611703353867"
```

```r
print(paste("Mean Squared Error (MSE):", mse))
```

```
## [1] "Mean Squared Error (MSE): 23.4393275093793"
```

```r
print(paste("Root Mean Squared Error (RMSE):", rmse))
```

```
## [1] "Root Mean Squared Error (RMSE): 4.84141792343723"
```

```r
print(paste("R-squared (R^2):", r_squared))
```

```
## [1] "R-squared (R^2): 0.742787646045595"
```

```r
print(paste("Adjusted R-squared:", adjusted_r_squared))
```

```
## [1] "Adjusted R-squared: 0.742261111032382"
```

```r
print(paste("F-statistic:", f_statistic))
```

```
## [1] "F-statistic: 1410.70893180191"
```

Model Performance Summary:
The linear regression model yields a moderate-to-high predictive performance for estimating life expectancy. The average absolute difference between predicted and actual life expectancies (MAE) is 3.45 years, while the root mean squared error (RMSE) stands at 4.84 years. The model explains approximately 74.34% of

the variance in life expectancy (R-squared). This indicates a substantial level of predictive accuracy, with a statistically significant overall model (F-statistic = 1212.82).

Cross-Validation Insights: Cross-validation further validates the model's consistency, with RMSE ranging from approximately 4.49 to 5.13 years across different folds. R-squared values consistently range from approximately 0.71 to 0.78, affirming stable predictive performance.

Overall, the linear regression model demonstrates robust predictive capabilities, capturing essential factors influencing life expectancy with a notable level of accuracy and consistency.

(e) Iterate and improve your algorithm performance. Write down the options you tried and the methods you used to increase the performance of your learning algorithm. Key things to mention are how you determined algorithm parameters, which variables and feature scaling options you explored. Visualize the process if possible (e.g., performance for different parameter values).

Method 1. Remove useless variables

```
# Perform Linear Regression without the variable 'percentage.expenditure' and 'Alcohol'
model_without_percentage <- lm(Life.expectancy ~ Adult.Mortality + BMI + Schooling + StatusDeveloped + G

# Summary
summary(model_without_percentage)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ Adult.Mortality + BMI + Schooling +
##     StatusDeveloped + GDP, data = scaled_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.4043  -2.2910   0.2608   2.9568  22.0153
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      68.9902     0.1021 675.708  < 2e-16 ***
## Adult.Mortality  -3.8105     0.1031 -36.943  < 2e-16 ***
## BMI               1.2654     0.1075  11.768  < 2e-16 ***
## Schooling         4.1755     0.1248  33.470  < 2e-16 ***
## StatusDeveloped   1.3238     0.2879   4.597 4.46e-06 ***
## GDP               0.6990     0.1035   6.756 1.70e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.82 on 2932 degrees of freedom
## Multiple R-squared:  0.7442, Adjusted R-squared:  0.7437
## F-statistic:  1706 on 5 and 2932 DF,  p-value: < 2.2e-16
```

Method 2. Add an interaction term between Status and GDP

```
# Perform LS with interaction between 'Status' and 'GDP', without 'percentage.expenditure' and 'Alcohol
model_with_interaction1 <- lm(Life.expectancy ~ Adult.Mortality + BMI + Schooling
                     + StatusDeveloped + GDP + StatusDeveloped:GDP, data = scaled_data)

# Summary
summary(model_with_interaction1)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ Adult.Mortality + BMI + Schooling +
##     StatusDeveloped + GDP + StatusDeveloped:GDP, data = scaled_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.3233  -2.3205   0.2944   2.9808  21.8506
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          69.0517     0.1042 662.936  < 2e-16 ***
## Adult.Mortality      -3.7954     0.1031 -36.797  < 2e-16 ***
## BMI                   1.2180     0.1086  11.213  < 2e-16 ***
## Schooling             4.1233     0.1259  32.751  < 2e-16 ***
## StatusDeveloped       1.5775     0.3007   5.247 1.66e-07 ***
## GDP                   1.0994     0.1727   6.365 2.27e-10 ***
## StatusDeveloped:GDP  -0.6181     0.2137  -2.892  0.00385 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.814 on 2931 degrees of freedom
## Multiple R-squared:  0.7449, Adjusted R-squared:  0.7444
## F-statistic:  1426 on 6 and 2931 DF,  p-value: < 2.2e-16
```

Method 3. Add an interaction term between Adult.Mortality and Alcohol

```r
# Perform LS with interaction between 'Adult.Mortality' and 'Alcohol'
model_with_interaction2 <- lm(Life.expectancy ~ Adult.Mortality + Alcohol + BMI + Schooling + StatusDeve

# Summary
summary(model_with_interaction2)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ Adult.Mortality + Alcohol + BMI +
##     Schooling + StatusDeveloped + GDP + Adult.Mortality:Alcohol,
##     data = scaled_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.2895  -2.2878   0.2442   2.9126  22.7577
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              68.9208     0.1044 660.112  < 2e-16 ***
## Adult.Mortality          -3.8545     0.1066 -36.152  < 2e-16 ***
## Alcohol                  -0.3007     0.1181  -2.547 0.010922 *
## BMI                       1.2866     0.1074  11.976  < 2e-16 ***
## Schooling                 4.3123     0.1324  32.576  < 2e-16 ***
## StatusDeveloped           1.2832     0.3336   3.847 0.000122 ***
## GDP                       0.6790     0.1033   6.570 5.92e-11 ***
## Adult.Mortality:Alcohol  -0.3751     0.1169  -3.210 0.001343 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.808 on 2930 degrees of freedom
## Multiple R-squared:  0.7456, Adjusted R-squared:  0.745
## F-statistic:  1227 on 7 and 2930 DF,  p-value: < 2.2e-16
```

The regression analysis was performed using three different methods to examine the relationship between Life Expectancy and various predictor variables. Method 1 included Adult Mortality, BMI, Schooling, Status of Development, and GDP as predictors. Method 2 extended Method 1 by adding an interaction term between Status of Development and GDP. Method 3 further expanded the model by including Alcohol consumption and an interaction term between Adult Mortality and Alcohol.

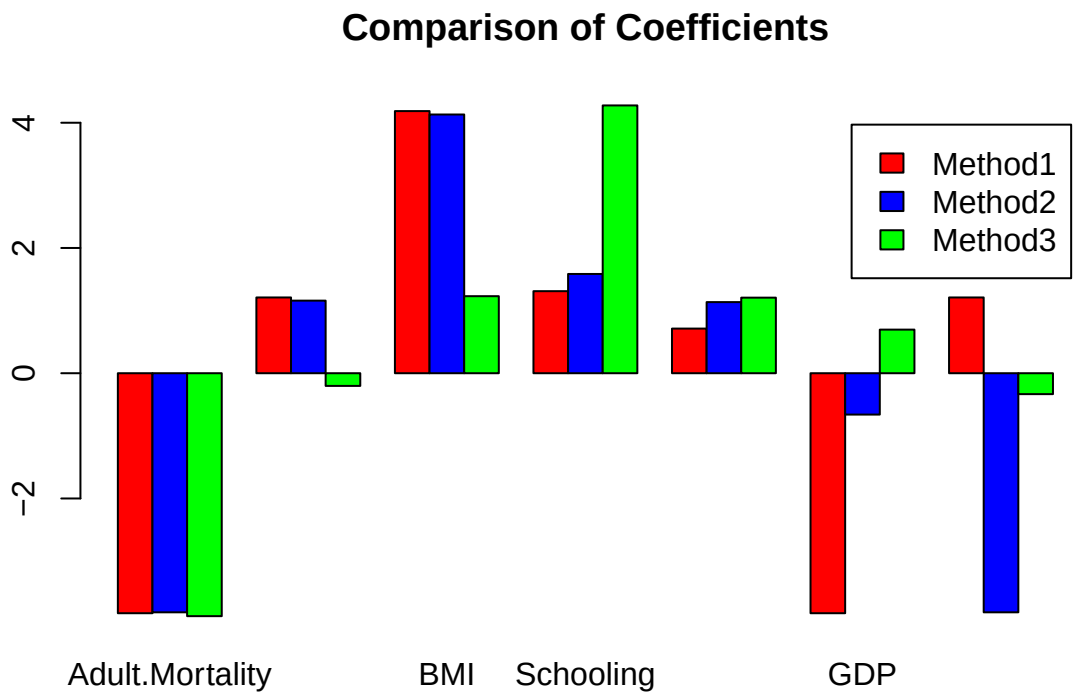
Comparing the results, all three methods yielded regression models with similar goodness-of-fit statistics, indicated by Adjusted R^2 values ranging from 0.7426 to 0.7436. Across the methods, Adult Mortality, BMI, Schooling, Status of Development, and GDP consistently showed significant associations with Life Expectancy, with Method 3 additionally highlighting a significant negative association between Alcohol consumption and Life Expectancy. Furthermore, the interaction terms in Methods 2 and 3 revealed nuanced relationships, with Method 3 also indicating a significant interaction between Adult Mortality and Alcohol consumption.

These findings underscore the multifaceted nature of factors influencing Life Expectancy, incorporating aspects such as healthcare, education, economic development, and lifestyle choices. While the inclusion of additional variables and interaction terms slightly refined the models, the core predictors remained consistent across the methods, emphasizing their robust associations with Life Expectancy.

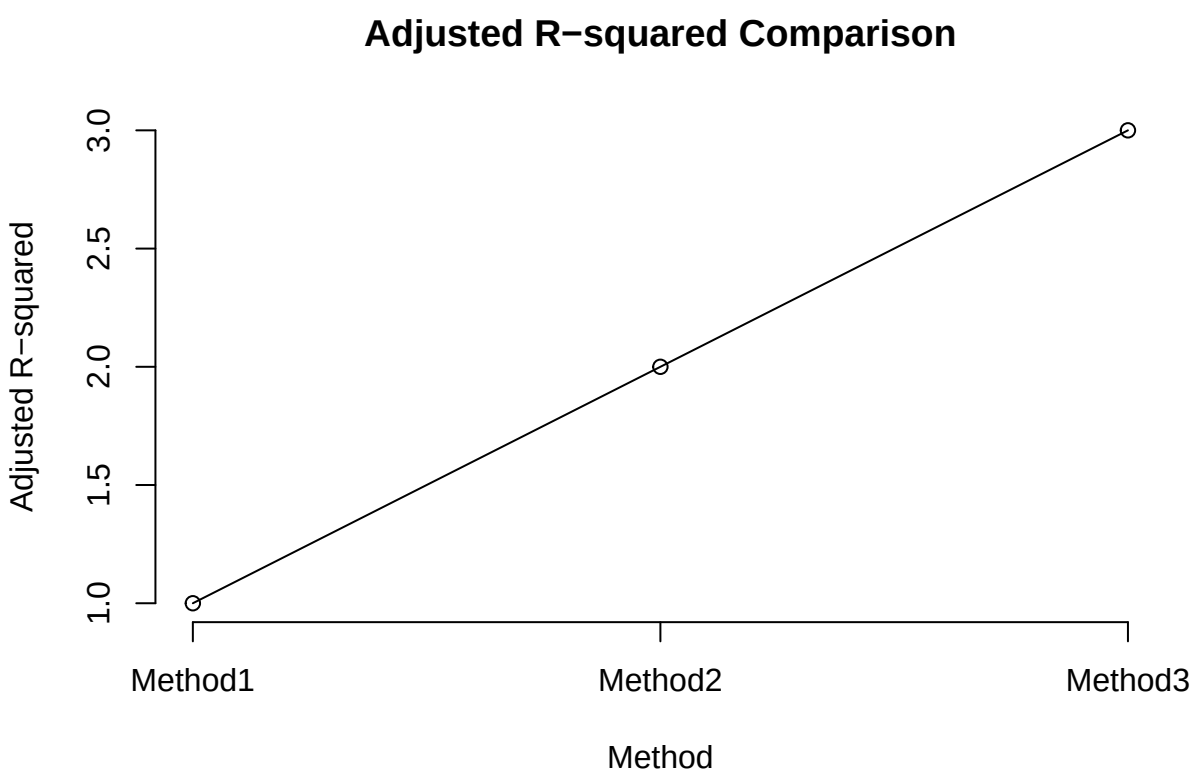
Some visualizations

```r
# Barplot
barplot(rbind(model_without_percentage$coefficients[-1], model_with_interaction1$coefficients[-1], model
        beside = TRUE,
        col = c("red", "blue", "green"),
        names.arg = colnames(model_without_percentage$coefficients)[-1],
        legend.text = c("Method1", "Method2", "Method3"),
        main = "Comparison of Coefficients")
```

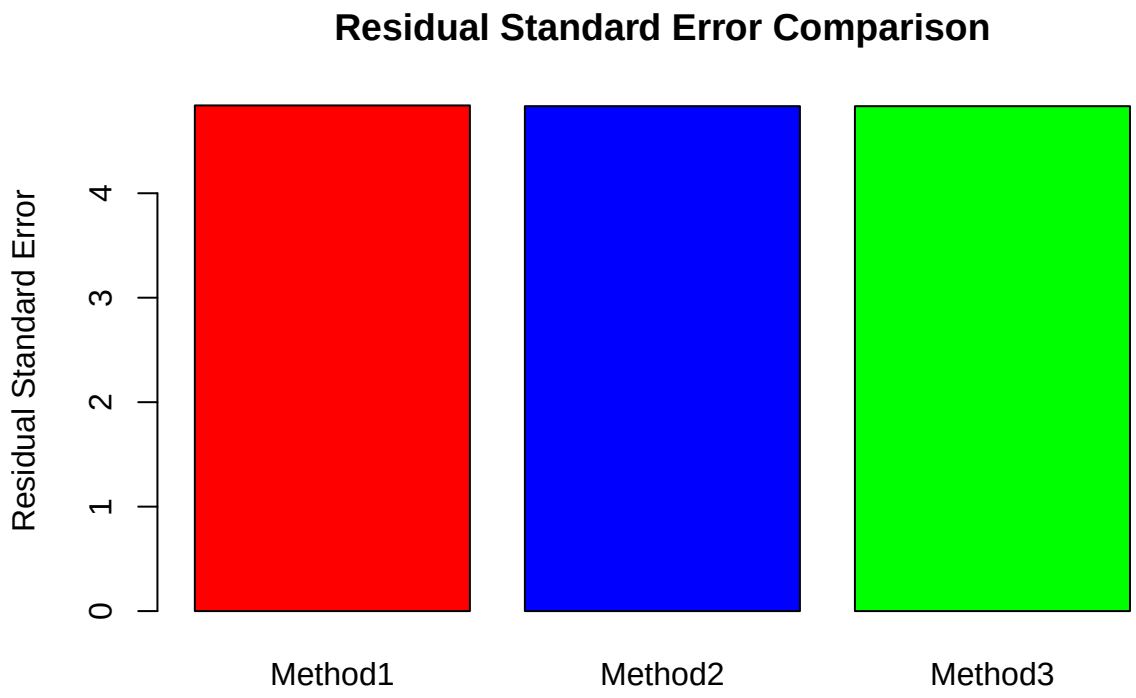
```
## Warning in base::rbind(...): number of columns of result is not a multiple of
## vector length (arg 1)
```



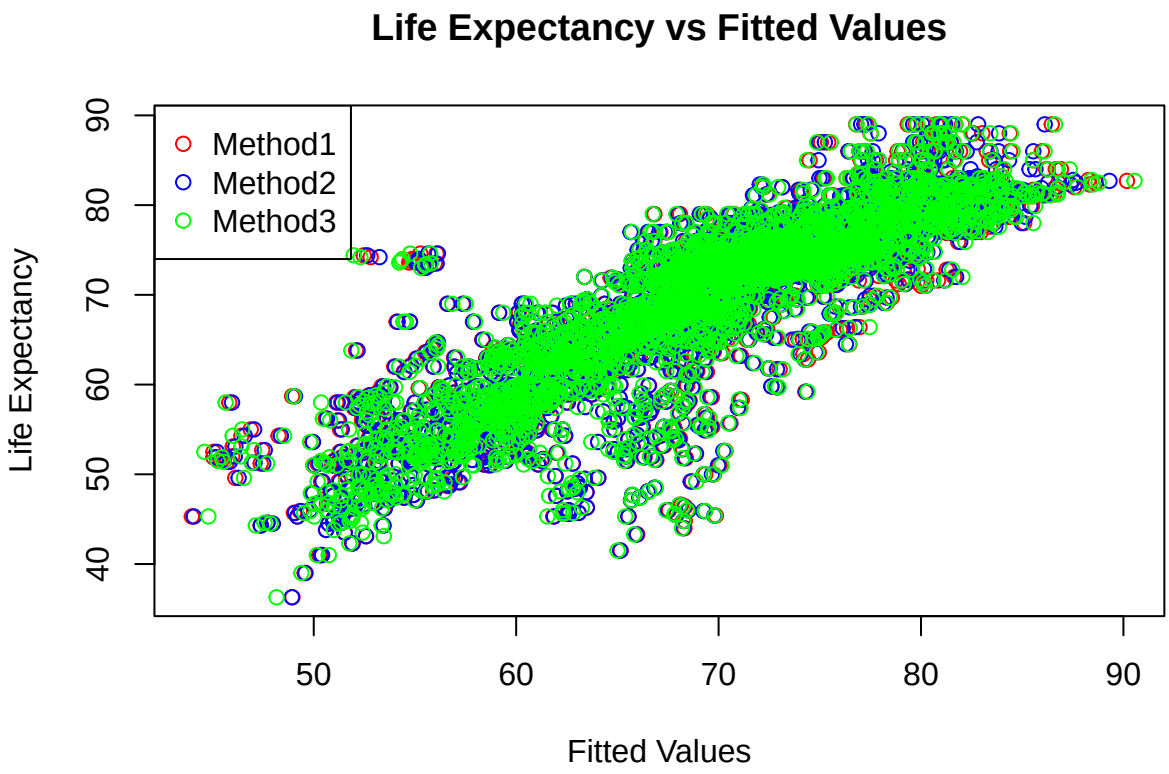

```r
# Line plot
plot(1:3, c(model_without_percentage$adj.r.squared, model_with_interaction1$adj.r.squared, model_with_i
     type = "o",
     xlab = "Method",
     ylab = "Adjusted R-squared",
     main = "Adjusted R-squared Comparison",
     axes = FALSE)
axis(1, at = 1:3, labels = c("Method1", "Method2", "Method3"))
axis(2)
```




```r
# Extract residual standard error from model_without_percentage
summary_model1 <- summary(model_without_percentage)
sigma_model1 <- summary_model1$sigma

# Extract residual standard error from model_with_interaction1
summary_model2 <- summary(model_with_interaction1)
sigma_model2 <- summary_model2$sigma

# Extract residual standard error from model_with_interaction2
summary_model3 <- summary(model_with_interaction2)
sigma_model3 <- summary_model3$sigma

# Barplot
barplot(c(sigma_model1, sigma_model2, sigma_model3),
        col = c("red", "blue", "green"),
        names.arg = c("Method1", "Method2", "Method3"),
        main = "Residual Standard Error Comparison",
        ylab = "Residual Standard Error")
```




```
# Scatterplot
plot(scaled_data$Life.expectancy ~ fitted(model_without_percentage), col = "red", xlab = "Fitted Values
points(scaled_data$Life.expectancy ~ fitted(model_with_interaction1), col = "blue")
points(scaled_data$Life.expectancy ~ fitted(model_with_interaction2), col = "green")
legend("topleft", legend = c("Method1", "Method2", "Method3"), col = c("red", "blue", "green"), pch = 1
```



3. Multiple linear regression

Let Y be a n $\times$ 1 vector for the response variable, X be a n $\times$ (p + 1) design matrix where the first column contains 1's, β be the (p + 1) $\times$ 1 vector of regression coefficients, and ϵ is a n $\times$ 1 vector of error terms. Then the multiple linear regression model can be written as $Y = X\beta + \epsilon$ where $\epsilon \sim$ MVN(0; $\sigma^2 I$), and I is an identity matrix.

(a) Find the least square (LS) estimate of β: Show that LS estimator $\hat{\beta}_{LS}$ is an unbiased estimate of β [i.e., $E(\hat{\beta}_{LS}) = \beta$] and Cov$(\hat{\beta}_{LS}) = \sigma^2(X^TX)^{-1}$

- **Find the LS estimate of** β:

We establish the objective function $Q(\beta) = (Y - X\beta)^T(Y - X\beta)$, and then calculate the gradient of Q with respect to β to obtain $\frac{\partial Q}{\partial \beta} = -2X^T(Y - X\beta)$

Set $\frac{\partial Q}{\partial \beta} = 0$ for optima, resulting in:

$$X^T(Y - X\beta) = 0$$
$$\Rightarrow X^TY = X^TX\beta$$
$$\Rightarrow \hat{\beta}_{LS} = (X^TX)^{-1}X^TY$$

is the LS estimate of β.

- **Show that** $\hat{\beta}_{LS}$ is an unbiased estimate of β:

$$E(\hat{\beta}_{LS}) = E((X^TX)^{-1}X^TY)$$

$$= (X^T X)^{-1} X^T E(Y)$$
$$= (X^T X)^{-1} X^T (X\beta)$$
$$= (X^T X)^{-1} (X^T X)\beta$$
$$= I\beta$$

- **Show that** $\text{Cov}(\hat{\beta}_{LS}) = \sigma^2 (X^T X)^{-1}$

$$\text{Cov}(\hat{\beta}_{LS}) = \text{Var}(\hat{\beta}_{LS})$$
$$= \text{Var}((X^T X)^{-1} X^T Y)$$
$$= (X^T X)^{-1} X^T \cdot \text{Var}(Y) \cdot ((X^T X)^{-1} X^T)^T$$
$$= ((X^T X)^{-1})^T \cdot \sigma^2 I \cdot (X^T X)^{-1} \cdot (X^T X)$$
$$= \sigma^2 \cdot (X^T X)^{-1}$$

(b) Write down the likelihood function (LF) for β and σ^2. Find the maximum likelihood estimate (MLE) of β and σ^2. Compare the ML estimates with LS estimates from (a)

- **The likelihood function (LF) for** β and σ^2

$$L(\beta, \sigma^2 | X, y) = (2\pi\sigma^2)^{-\frac{n}{2}} . e^{-\frac{1}{2\sigma^2}(Y - X\beta)^T (Y - X\beta)}$$

- **Find the maximum likelihood estimate (MLE) of** β and σ^2

Because the maximum likelihood estimate (MLE) is also the maximum log-likelihood estimate (MLLE), we establish the log-likelihood function for β and σ^2:

$$l(\beta, \sigma^2 | X, y) = \log L(\beta, \sigma^2 | X, y) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(Y - X\beta)^T (Y - X\beta)$$

Then, we calculate the derivative of log-likelihood function with respect to β to obtain the MLLE of β.

$$\frac{\partial l(\beta, \sigma^2 | X, y)}{\partial \beta} = \frac{-1}{2\sigma^2}[-2X^T (Y - X\beta)]$$

The normal equation is $X^T Y = X^T X\beta$. Hence, the MLLE for β is $\hat{\beta}_{ML} = (X^T X)^{-1} X^T Y$, which is the same as $\hat{\beta}_{LS}$.

Next, we we calculate the derivative of log-likelihood function with respect to σ^2 to obtain the MLLE of σ^2.

$$\frac{\partial l(\beta, \sigma^2 | X, y)}{\partial \sigma^2} = \frac{-n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}(Y - X\beta)^T (Y - X\beta)$$

Hence, the MLLE of σ^2 is $\hat{\sigma^2}_{\text{ML}} = \frac{1}{n}(Y - X\beta)^T (Y - X\beta)$.

(c) Derive $\text{Var}(\hat{\beta}_{\text{ML}})$, where $\hat{\beta}_{\text{ML}}$ is the maximum likelihood estimator (MLE) of β

Since the MLE of β is the same as $\hat{\beta}_{LS}$, and by part (a), $\text{Var}(\hat{\beta}_{LS}) = \sigma^2 \cdot (X^T X)^{-1}$,

$$\text{Var}(\hat{\beta}_{\text{ML}}) = \sigma^2 \cdot (X^T X)^{-1}$$

(d) Given x_k a data value from the training data, derive $\hat{y_k}$ and find Var($\hat{y}_k$)

The predicted value $\hat{y_k} = x_k\hat{\beta}$, where $\hat{\beta}$ is LSE or MLE of β. Hence, $\hat{y_k} = x_k(X^TX)^{-1}X^TY$.

The matrix form for the fitted value is $\hat{Y} = X\hat{\beta}$. Define the Hat matrix H as $\mathrm{H} = X(X^TX)^{-1}X^T$, then $\hat{Y} = \mathrm{H}Y$. Since $\mathrm{Var}(Y) = \mathrm{Var}(\epsilon) = \sigma^2\mathrm{I}$,

$$\mathrm{Var}(\hat{Y}) = \mathrm{Var}(\mathrm{H}Y) = \mathrm{H}^T \cdot \mathrm{Var}(Y) \cdot \mathrm{H} = \mathrm{H}^T \cdot \sigma^2\mathrm{I} \cdot \mathrm{H} = \sigma^2\mathrm{H}^T\mathrm{H} = \sigma^2\mathrm{H}$$

Therefore, we obtain the $\mathrm{Var}(\hat{y}_k) = \sigma^2\mathrm{h}_k$, where $\mathrm{h}_k = x_k(X^TX)^{-1}x_k^T$ is the k-th vector of matrix H.

The aforementioned outcome stems from two special properties of the Hat matrix:

- *Symmetric*: $\mathrm{H} = \mathrm{H}^T$
- *Idempotent*: $\mathrm{H}^2 = \mathrm{H}$

We omitted the proof as it lies beyond the scope of the assignment.

(e) Given x_{new}, a new vector of feature variables from the test data, derive $\hat{Y}_{new}$ and find Var($\hat{Y}_{new}$). Also, find Var($Y_{new} - \hat{Y}_{new}$).

- **Derive** $\hat{Y}_{new}$

$\hat{Y}_{new}$ can be calculated using the linear regression equation:

$$\hat{Y}_{new} = X_{new}\hat{\beta}$$

where X_{new} is the design matrix for the new data, and $\hat{\beta}$ is the estimate of β.

- **Find** $\mathrm{Var}(\hat{Y}_{new})$

$$\mathrm{Var}(\hat{Y}_{new}) = \mathrm{Var}(X_{new}\hat{\beta}) = X_{new}^T \cdot \mathrm{Var}(\hat{\beta}) \cdot X_{new}$$

By part (c), $\mathrm{Var}(\hat{\beta}) = \sigma^2 \cdot (X^TX)^{-1}$. Hence,

$$\mathrm{Var}(\hat{Y}_{new}) = X_{new}^T \cdot \sigma^2 \cdot (X^TX)^{-1} \cdot X_{new} = \sigma^2 \cdot \mathrm{H}$$

where $\mathrm{H} = X_{new} \cdot (X^TX)^{-1} \cdot X_{new}^T$

- **Find** $\mathrm{Var}(Y_{new} - \hat{Y}_{new})$

$$\mathrm{Var}(Y_{new} - \hat{Y}_{new}) = \mathrm{Var}(Y_{new}) + \mathrm{Var}(\hat{Y}_{new}) - 2\mathrm{Cov}(Y_{new}, \hat{Y}_{new}) = \sigma^2 I + \sigma^2\mathrm{H} - 2\sigma^2\mathrm{H} = \sigma^2(I - \mathrm{H})$$

4. Linear regression

Consider the regression model:

$$E(Y_j) = \beta_0 + \beta_1 x_1 + \beta_2(3x_j^2 - 2), \quad j = 1, 2, 3$$

Where $x_1 = -1, \quad x_2 = 0$, and $x_3 = 1$

(a) Find the least squares estimate (LSE) of β_0, β_1, and β_2.

$$\hat{y}(-1) = E(y|x_1 = -1) = \beta_0 - \beta_1 + \beta_2$$
$$\hat{y}(0) = E(y|x_2 = 0) = \beta_0 - 2\beta_2$$
$$\hat{y}(1) = E(y|x_3 = 1) = \beta_0 + \beta_1 + \beta_2$$

Setting the loss function:

$$Q(\beta) = \sum (y_i - \hat{y_i})^2 = [y(-1) - \beta_0 + \beta_1 - \beta_2]^2 + [y(0) - \beta_0 + 2\beta_2]^2 + [y(1) - \beta_0 - \beta_1 - \beta_2]^2$$

where $y(-1), y(0), y(1)$ are observed values of the response given $x = -1, x = 0, x = 1$, respectively. Calculating the gradient of the loss function with respect to β:

$$\frac{\partial Q}{\partial \beta_0} = -2[y(-1) - \beta_0 + \beta_1 - \beta_2] - 2[y(0) - \beta_0 + 2\beta_2] - 2[y(1) - \beta_0 - \beta_1 - \beta_2] \quad (1)$$
$$\frac{\partial Q}{\partial \beta_1} = 2[y(-1) - \beta_0 + \beta_1 - \beta_2] - 2[y(1) - \beta_0 - \beta_1 - \beta_2] \quad (2)$$
$$\frac{\partial Q}{\partial \beta_2} = -2[y(-1) - \beta_0 + \beta_1 - \beta_2] + 4[y(0) - \beta_0 + 2\beta_2] - 2[y(1) - \beta_0 - \beta_1 - \beta_2] \quad (3)$$

Set $\frac{\partial Q}{\partial \beta_i} = 0 \quad (i = 0, 1, 2)$ for optima

$$(1)(3) \Rightarrow y(0) - \beta_0 + 2\beta_2 = 0 \Rightarrow \beta_0 - 2\beta_2 = y(0)$$
$$(1)(2) \Rightarrow y(1) - \beta_0 + \beta_1 - \beta_2 = 0 \Rightarrow \beta_0 - \beta_1 + \beta_2 = y(1)$$
$$(1) \Rightarrow y(1) - \beta_0 - \beta_1 - \beta_2 = 0 \Rightarrow \beta_0 + \beta_1 + \beta_2 = y(1)$$

Solve for β_i obtains the LSE of β:

$$\hat{\beta_0} = \frac{1}{3}y(0) + \frac{1}{3}y(-1) + \frac{1}{3}y(1)$$
$$\hat{\beta_2} = -\frac{1}{3}y(0) + \frac{1}{6}y(-1) + \frac{1}{6}y(1)$$
$$\hat{\beta_1} = -\frac{1}{2}y(-1) + \frac{1}{2}y(1)$$

(b) Show that the LSE of β_0 and β_1 are unchanged if $\beta_2 = 0$

If $\beta_2 = 0$, then 3 equations for optima in part (a) become

$$\begin{cases} \beta_0 = y(0) \\ \beta_0 - \beta_1 = y(-1) \\ \beta_0 + \beta_1 = y(1) \end{cases} \Rightarrow \begin{cases} \beta_0 = y(0) \\ \beta_1 = \frac{1}{2}(y(1) - y(-1)) \\ y(1) - y(-1) = 2y(0) \end{cases}$$

Hence,

$$\beta_0 = \frac{1}{3}(y(0) + y(-1) + y(1))$$
$$\beta_1 = \frac{1}{2}(y(1) - y(-1))$$

which are unchanged.

5. Logistic regression

Let $Y_i \in 0, 1$, and x_i be the output and input variables in a logistic regression problem where the probability of success is defined as

$$\pi_i = Pr(Y_i = 1) = \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}}$$

where β_0 and β_1 are the intercept and slope parameters, respectively. To estimate the parameters β_0 and β_1, we define the likelihood function as

$$L(\beta) = \prod_{i=1}^{n} \pi_i^{y_i}(1 - \pi_i)^{1-y_i}$$

where n is the sample size. Write down the explicit form of the likelihood function to get the maximum likelihood estimate (MLE) of β_0 and β_1. How can we obtain the MLE of β_0 and β_1? (Here, you don't need to find out the MLE of β_0 and β_1. You just need to write the equations that are used to get the MLE of β_0 and β_1)

- **The explicit form of the likelihood function**:

$$\begin{aligned} L(\beta) &= \prod_{i=1}^{n} \left(\frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} \right)^{y_i} \cdot \left(1 - \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} \right)^{1-y_i} \\ &= \prod_{i=1}^{n} \left(\frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} \right)^{y_i} \cdot \left(\frac{1}{1 + e^{\beta_0+\beta_1 x_i}} \right)^{1-y_i} \\ &= \prod_{i=1}^{n} \frac{e^{\beta_0 y_i+\beta_1 x_i y_i}}{1 + e^{\beta_0+\beta_1 x_i}} \end{aligned}$$

- **Obtaining the MLE of** β_0 and β_1

We establish the log-likelihood function l, take the gradient of l, then set these quantities equal to zero to obtain the MLE of β_0 and β_1

The log-likelihood function:

$$\mathrm{l}(\beta) = \sum_{i=1}^{n} (\beta_0 y_i + \beta_1 x_i y_i - \log(1 + e^{\beta_0+\beta_1 x_i}))$$

The equations to obtain the MLE of β_0 and β_1:

$$\frac{\partial l(\beta)}{\partial \beta_0} = \sum_{i=1}^{n} \left(y_i - \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} \right)$$

$$\frac{\partial l(\beta)}{\partial \beta_1} = \sum_{i=1}^{n} \left(x_i y_i - \frac{x_i e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} \right)$$

6. TWINS Data

Use R for this question. The file twins.Rdata (available on Moodle) contains a subset of data on the National Merit Twin Study. Report (i) the number of twin pairs, (ii) the number of variables, (iii) the number of identical twins and (iv) the number of fraternal twins.

```r
load(file="twins.Rdata")
head(twins)
```

```
##    pairnum zygosity english.1 math.1 socsci.1 natsci.1 vocab.1 english.2 math.2
## 1        1        1        14     13       17       18      14        11     14
## 3        4        1        20     20       16       16      13        17     19
## 5        5        1        11      8       15       16      12        16     13
## 7        7        2         9     19        7       10       6         8     16
## 9       10        2        15     23       23       21      21        15     13
## 11      11        2        20     17       16       12      12        19     18
##    socsci.2 natsci.2 vocab.2
## 1        15       10      12
## 3        13       13      14
## 5        13        8      15
## 7        15       17      11
## 9        13       20      19
## 11       13       18      15
```

(i) The number of twin pairs

```r
# Calculate the number of twin pairs
cat("The number of twin pairs:\n")
```

```
## The number of twin pairs:
```

```r
print(nrow(twins))
```

```
## [1] 839
```

(ii) The number of variables

```r
cat("The number of variables:\n")
```

```
## The number of variables:
```

```r
print(ncol(twins))
```

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

(iii) The number of identical twins

```r
# Count the number of rows where Zygosity is equal to 1
num_identical <- sum(twins$zygosity == 1)
# Print the result
cat("The number of identical twins:\n")
```

```
## The number of identical twins:
```

```r
print(num_identical)
```

```
## [1] 509
```

(iv) The number of fraternal twins

```
# Count the number of rows where Zygosity is not equal to 1
num_fraternal <- sum(twins$zygosity !=1)
# Print the result
cat("The number of fraternal twins:\n")
```

```
## The number of fraternal twins:
```

```
print(num_fraternal)
```

```
## [1] 330
```

Report the quantities

```
# Create a data frame with the information
info_df <- data.frame(
  "Statistic" = c("Number of twin pairs", "Number of variables","Number of identical twins", "Number of
  "Value" = c(nrow(twins), ncol(twins),num_identical, num_fraternal)
)

# Print the table
print(info_df)
```

```
##                   Statistic Value
## 1      Number of twin pairs   839
## 2       Number of variables    12
## 3 Number of identical twins   509
## 4 Number of fraternal twins   330
```