Life Expectancy Analysis

Roswita Hede

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## plotting options
set_plot_dimensions <- function(width_choice , height_choice) {
options(repr.plot.width=width_choice, repr.plot.height=height_choice)
}
```

1. Data Preparation

```
df<- read.csv("Life Expectancy Data.csv")
head(df)
```

	Country <chr>	Y… <int>	Status <chr>	Life.expectancy <dbl>	Adult.Mortality <int>	infant.deaths <int>	Alcohol <dbl>
1	Afghanistan	2015	Developing	65.0	263	62	0.01
2	Afghanistan	2014	Developing	59.9	271	64	0.01
3	Afghanistan	2013	Developing	59.9	268	66	0.01
4	Afghanistan	2012	Developing	59.5	272	69	0.01
5	Afghanistan	2011	Developing	59.2	275	71	0.01
6	Afghanistan	2010	Developing	58.8	279	74	0.01

6 rows | 1-8 of 23 columns

```
#remove unnecassary column
df<-subset(df, select = -c(Country, Year))
```

```
#identify missing values on dataset
missing.rows = dim(df)[1] -  dim(na.omit(df))[1]
sprintf("Dataset size: [%s]", toString(dim(df)))
```

```
## [1] "Dataset size: [2938, 20]"
```

```
sprintf("Missing rows: %s (%s%%)", missing.rows, round((missing.rows*100)/dim(df)[1], 2))
```

```
## [1] "Missing rows: 1289 (43.87%)"
```

```
missings_df <- data.frame(type=c("missing", "non-missing") ,count = c(missing.rows,  dim(na.omit
(df))[1]))

missings_df
```

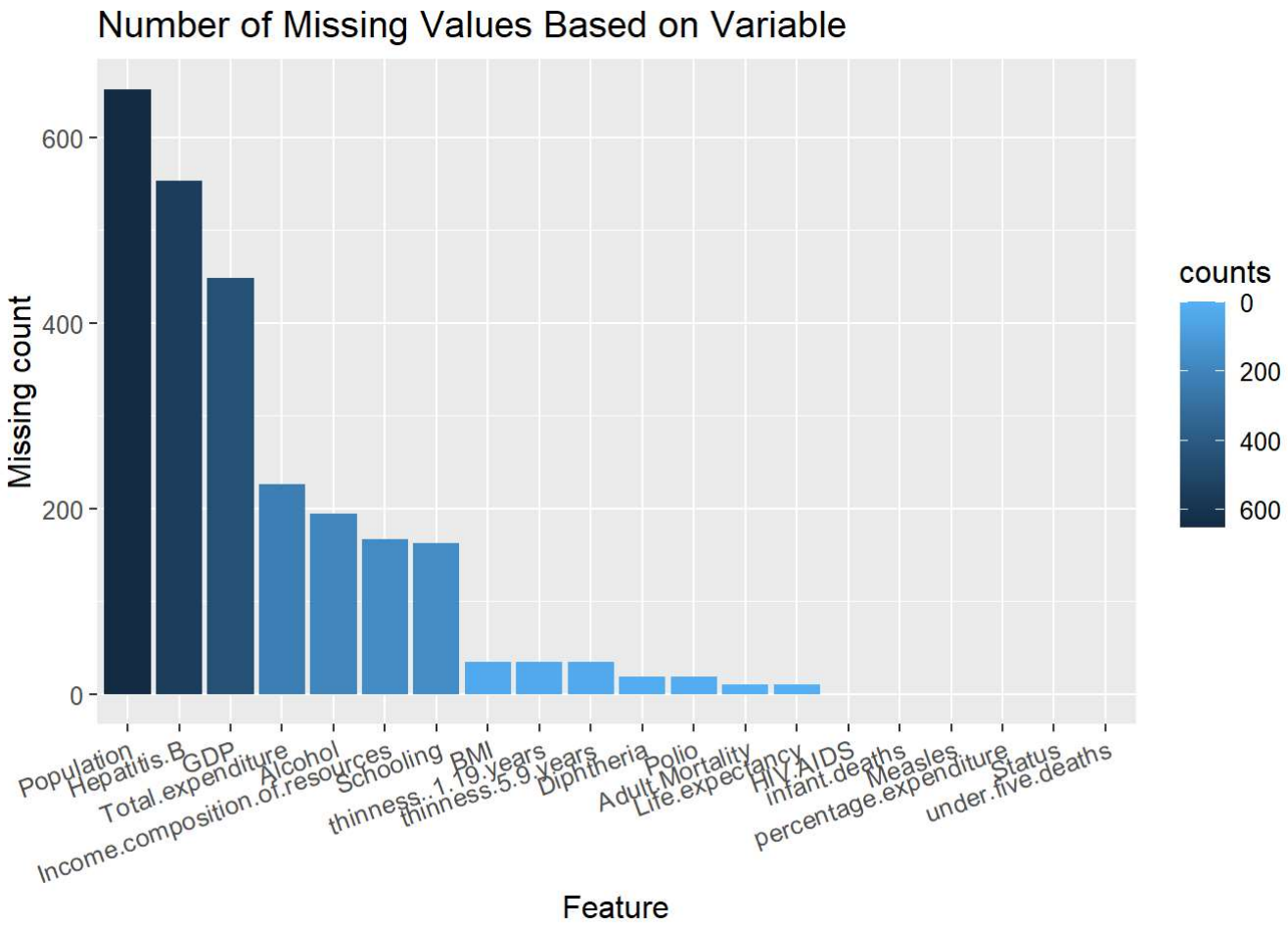
type <chr>	count <int>
missing	1289
non-missing	1649

2 rows

```
set_plot_dimensions(6,4)
ggplot(missings_df, aes(fill=type, y="", x=count)) +
    geom_bar(position="stack", stat="identity")+
    ggtitle("Missing vs Non-missing row counts") +
    xlab("Missing count") + ylab("") +
    theme(text = element_text(size = 12))+
    scale_fill_brewer(palette="Set2")
```

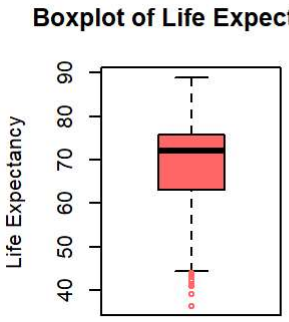
Based on the plot above, it can be seen that 43.87% data contains missing values.

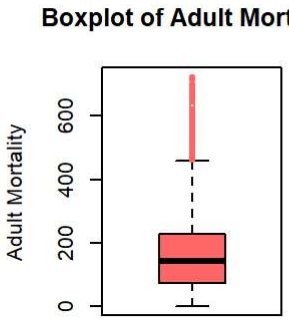
```
missing_counts <- data.frame(feature = factor(names(df)),
                    counts=sapply(df, function(x) sum(is.na(x))))
set_plot_dimensions(16,8)
ggplot(missing_counts,
       aes(x=reorder(feature, -counts), y=counts, fill=counts)) +
                                geom_bar(stat="identity") +
                                ggtitle("Number of Missing Values Based on Variable") +
                                xlab("Feature") + ylab("Missing count") +
                                theme(axis.text.x=element_text(angle=20, hjust=1))+
                                theme(text = element_text(size = 12))+
                                scale_fill_continuous(trans = 'reverse')
```

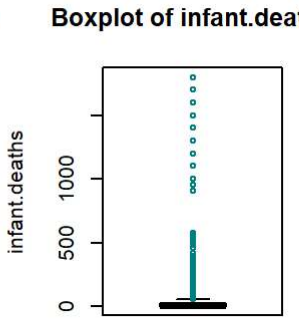


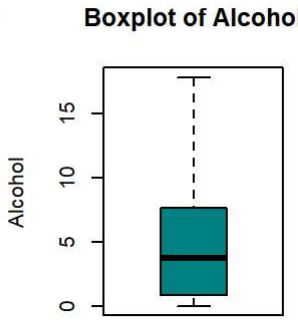
Based on the graph above, the variables with the most missing values are population, hepatitis B, and GDP. Since these variables have more than 40% missing values, we will use imputation to handle the missing values.

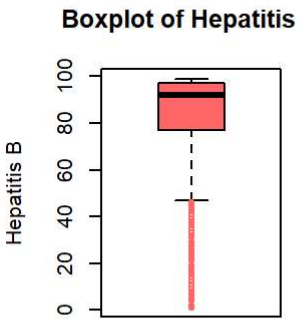
```
#Checking outliers
set_plot_dimensions(20,10)
par(mfrow=c(2,4))
boxplot(df$Life.expectancy,
        ylab = "Life Expectancy",
        main = "Boxplot of Life Expectancy",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$Adult.Mortality,
        ylab = "Adult Mortality",
        main = "Boxplot of Adult Mortality",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$infant.deaths,
        ylab = "infant.deaths",
        main = "Boxplot of infant.deaths",
        col= "#008080",
        outcol="#008080")
boxplot(df$Alcohol,
        ylab = "Alcohol",
        main = "Boxplot of Alcohol",
        col= "#008080",
        outcol="#008080")
boxplot(df$Hepatitis.B,
        ylab = "Hepatitis B",
        main = "Boxplot of Hepatitis B",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$BMI,
        ylab = "BMI",
        main = "Boxplot of BMI",
        col= "#008080",
        outcol="#008080")
boxplot(df$Measles,
        ylab = "Measles",
        main = "Boxplot of Measles",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$under.five.deaths,
        ylab = "under.five.deaths",
        main = "Boxplot of under.five.deaths",
        col= "#008080",
        outcol="#008080")
```

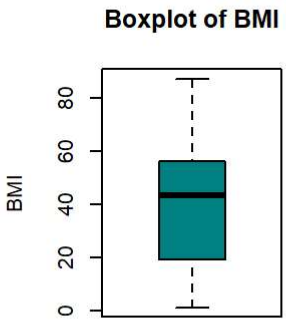

Boxplot of Life Expectanc
Life Expectancy
40
50
60
70
80
90

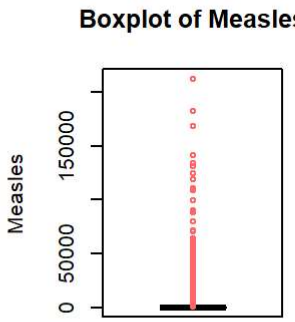

Boxplot of Adult Mortality
Adult Mortality
0
200
400
600

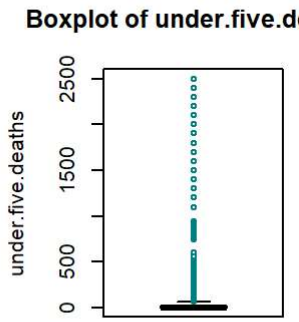

Boxplot of infant.deaths
infant.deaths
0
500
1000


Boxplot of Alcohol
Alcohol
0
5
10
15

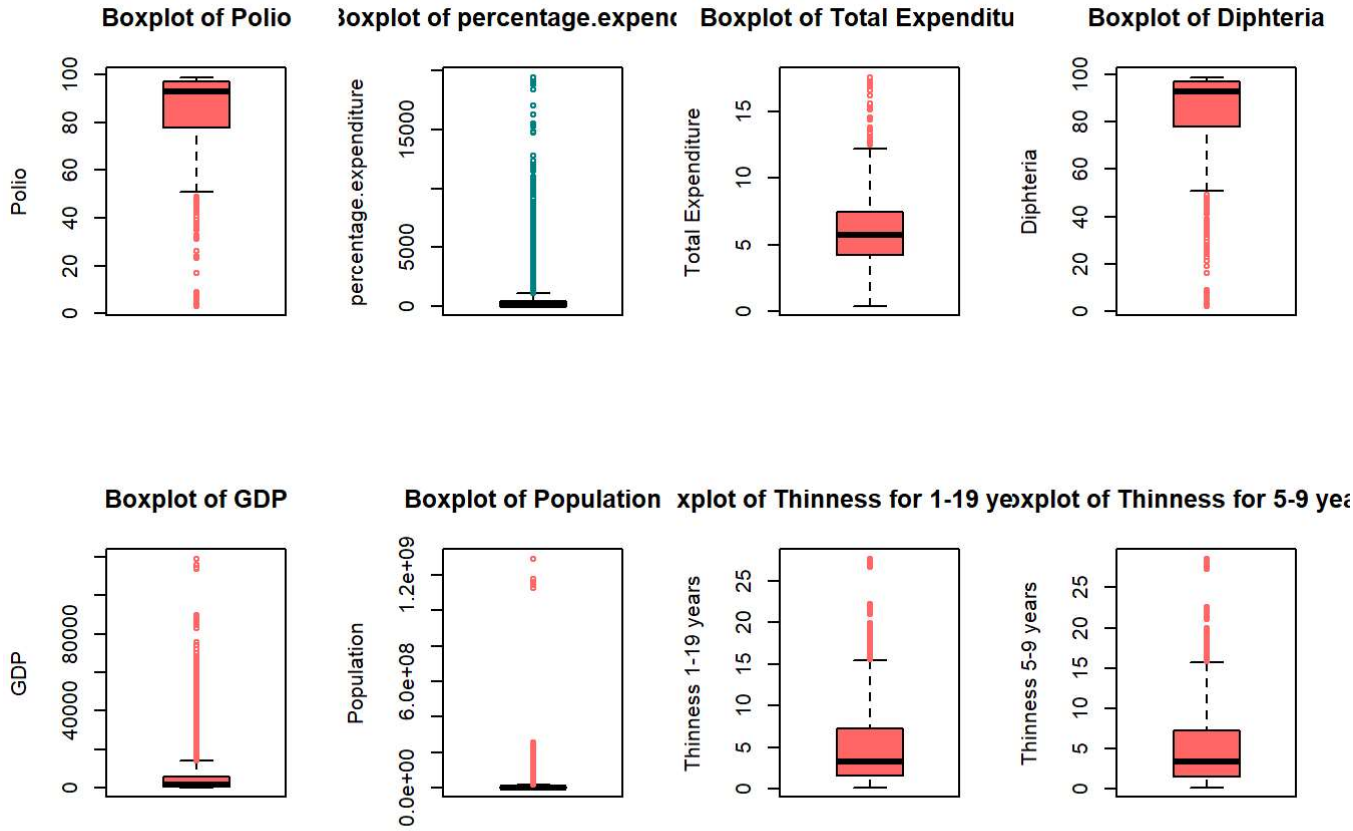

Boxplot of Hepatitis B
Hepatitis B
0
20
40
60
80
100


Boxplot of BMI
BMI
0
20
40
60
80

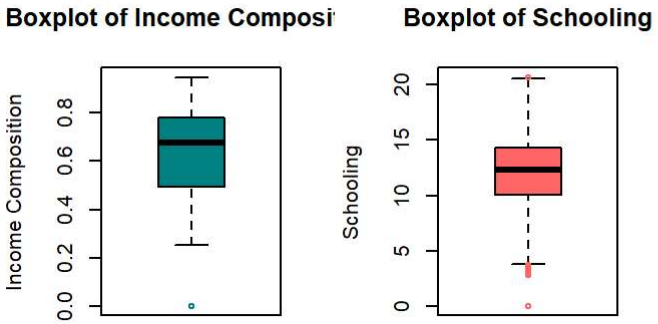

Boxplot of Measles
Measles
0
50000
150000


Boxplot of under.five.deatl
under.five.deaths
0
500
1500
2500

```
boxplot(df$Polio,
        ylab = "Polio",
        main = "Boxplot of Polio",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$percentage.expenditure,
        ylab = "percentage.expenditure",
        main = "Boxplot of percentage.expenditure",
        col= "#008080",
        outcol="#008080")
boxplot(df$Total.expenditure,
        ylab = "Total Expenditure",
        main = "Boxplot of Total Expenditure",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$Diphtheria,
        ylab = "Diphteria",
        main = "Boxplot of Diphteria",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$GDP,
        ylab = "GDP",
        main = "Boxplot of GDP",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$Population,
        ylab = "Population",
        main = "Boxplot of Population",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$thinness..1.19.years,
        ylab = "Thinness 1-19 years",
        main = "Boxplot of Thinness for 1-19 years old",
        col= "#FF6666",
        outcol="#FF6666")
boxplot(df$thinness.5.9.years,
        ylab = "Thinness 5-9 years",
        main = "Boxplot of Thinness for 5-9 years old",
        col= "#FF6666",
        outcol="#FF6666")
```




```
boxplot(df$Income.composition.of.resources,
        ylab = "Income Composition",
        main = "Boxplot of Income Composition",
        col= "#008080",
        outcol="#008080")
boxplot(df$Schooling,
        ylab = "Schooling",
        main = "Boxplot of Schooling",
        col= "#FF6666",
        outcol="#FF6666")
```




```
library(ggpubr)
```

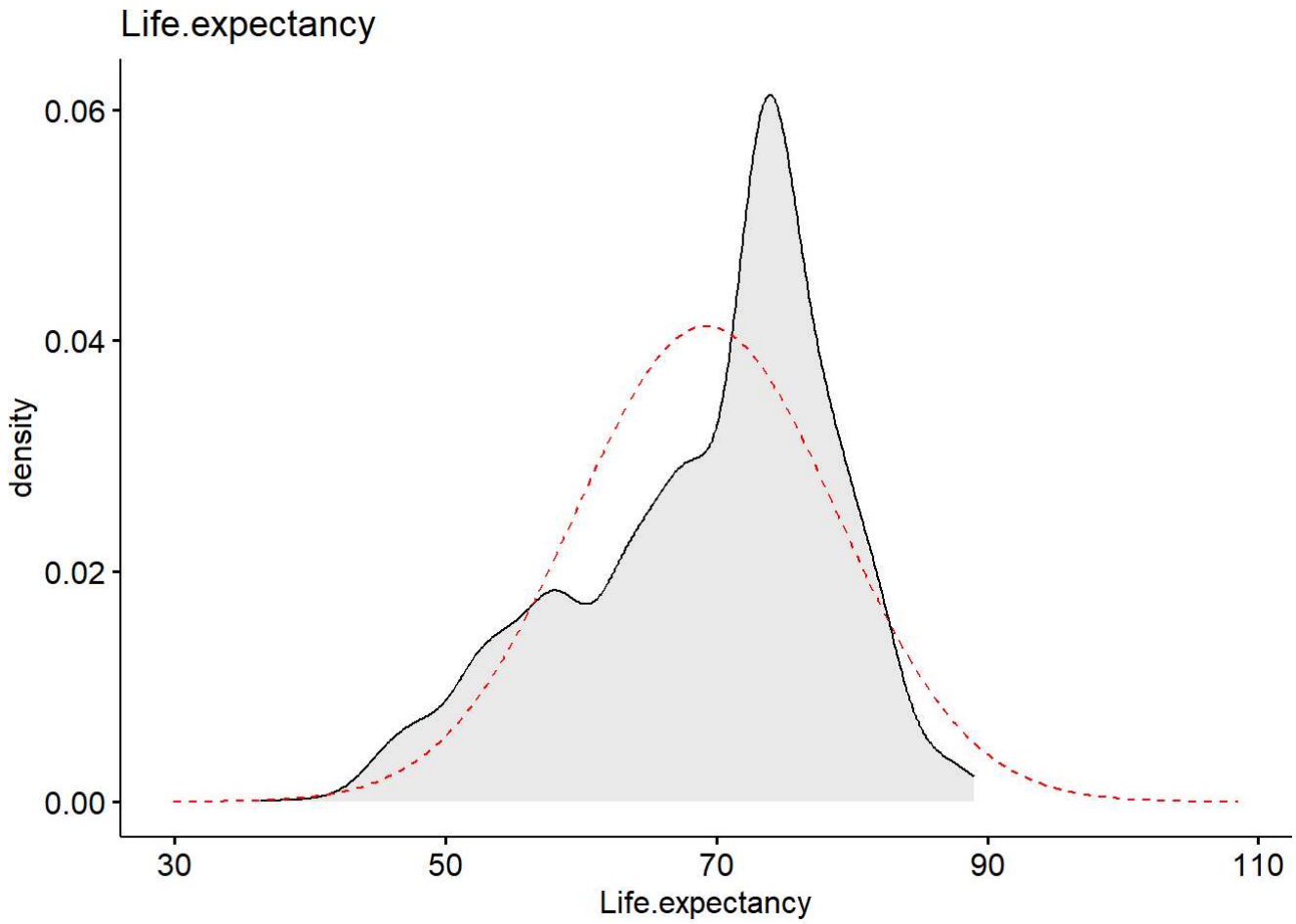
```
## Warning: package 'ggpubr' was built under R version 4.0.5
```

```
par(mfrow=c(2,4))
#distribution of life expectancy
ggdensity(df, x = "Life.expectancy", fill = "lightgray", title = "Life.expectancy") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.

## Warning: Please use `after_stat(density)` instead.
```

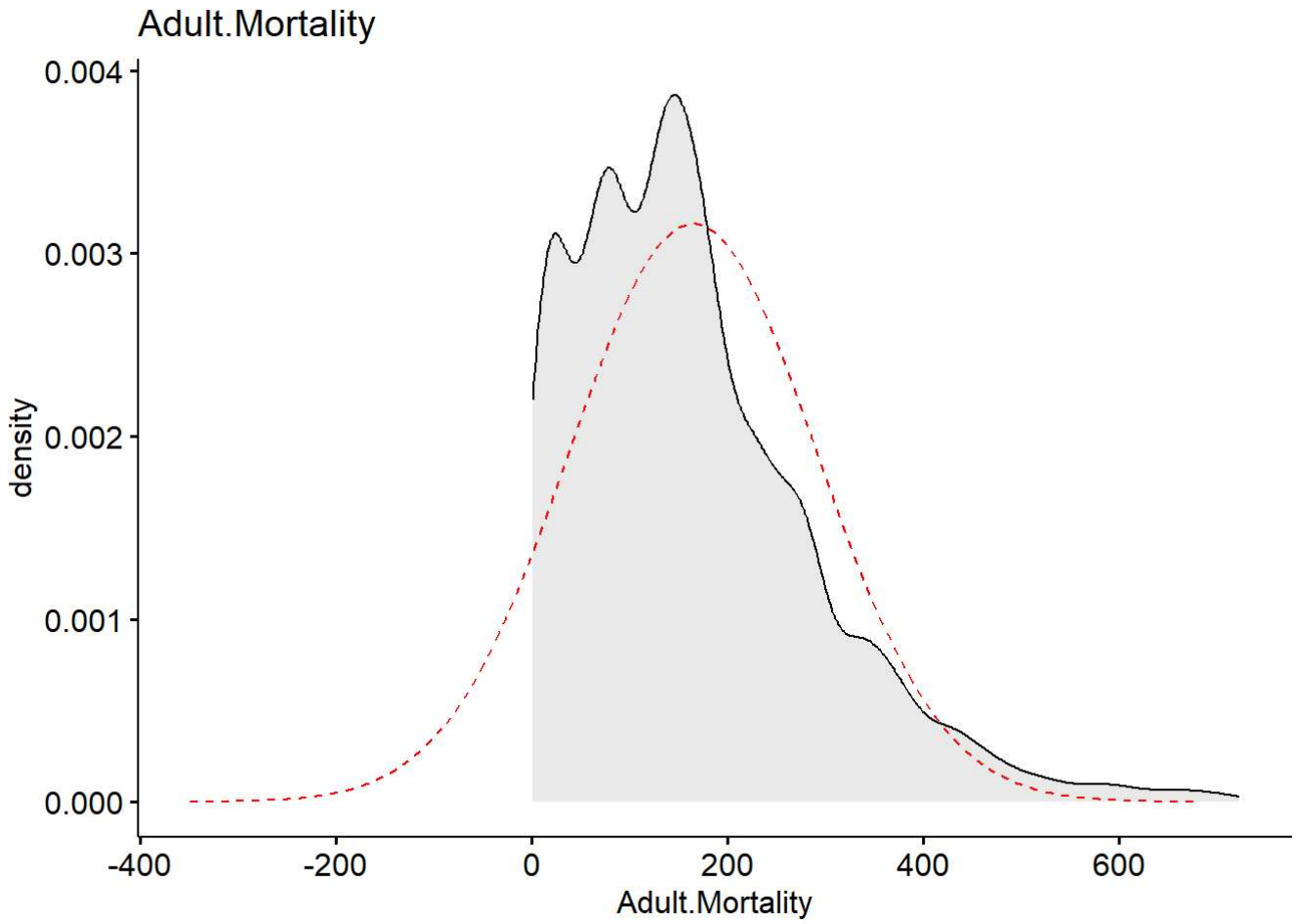
```
## Warning: Removed 10 rows containing non-finite values (`stat_density()`).
```

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

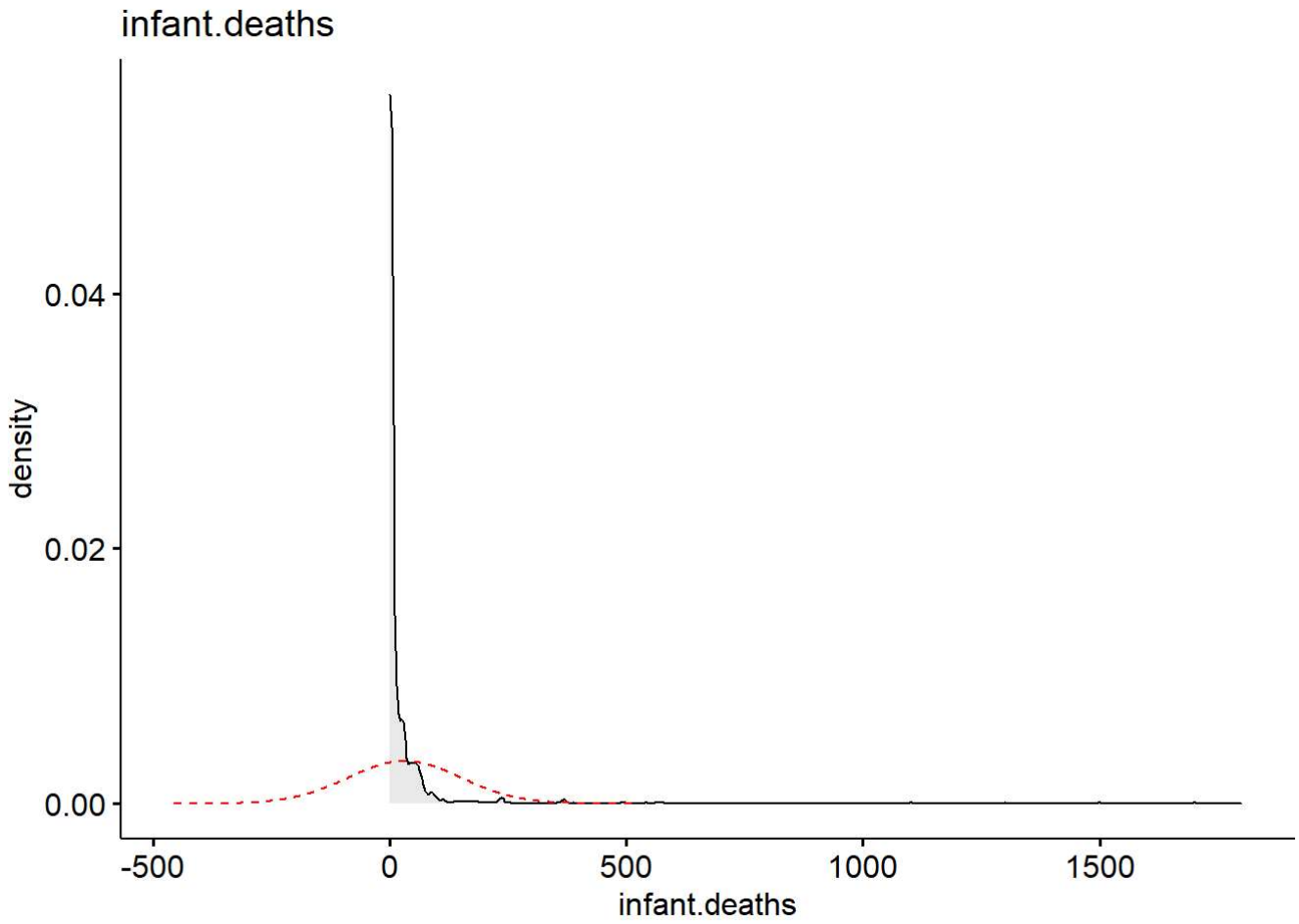



```
ggdensity(df, x = "Adult.Mortality", fill = "lightgray", title = "Adult.Mortality") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

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

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



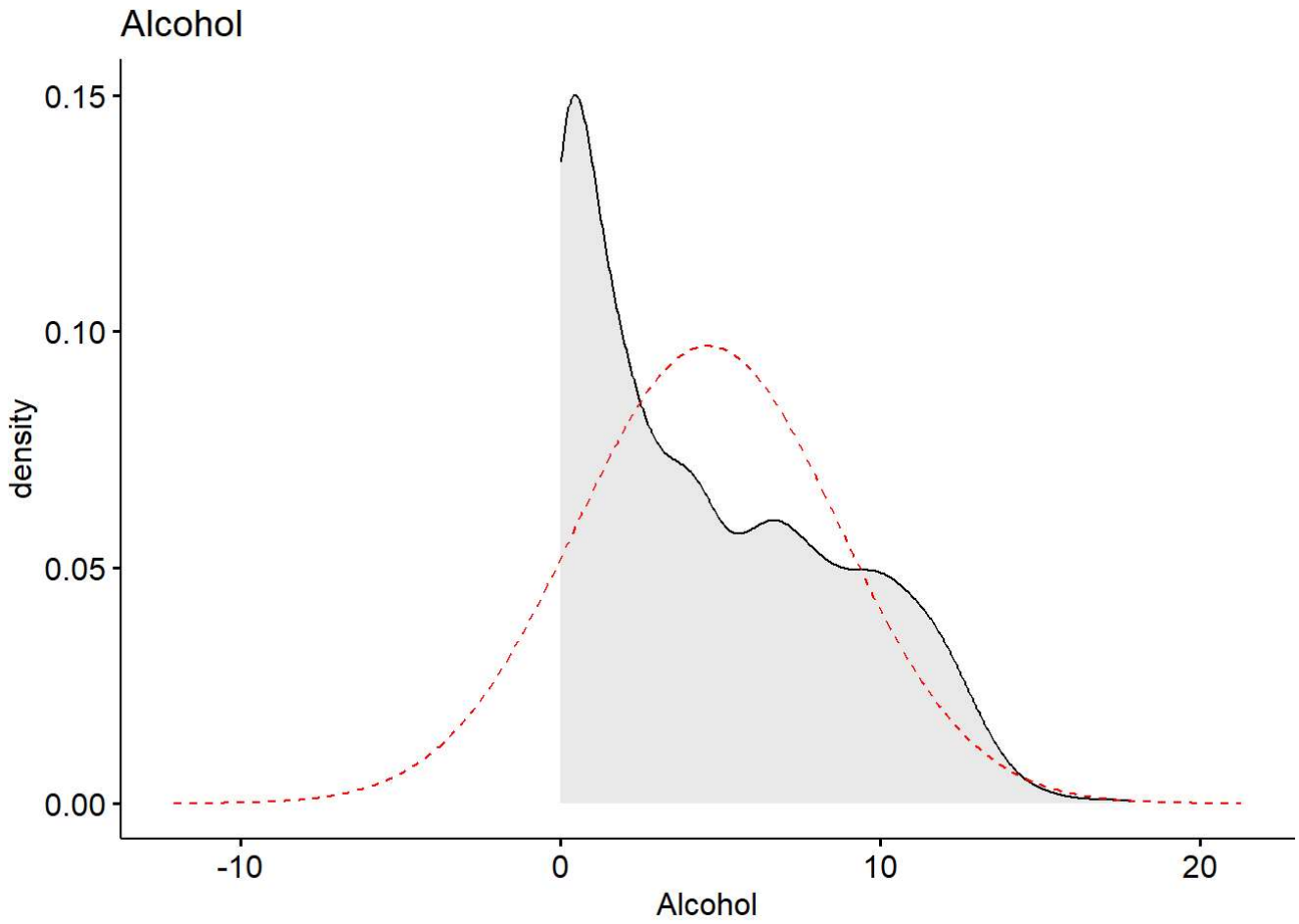

```
ggdensity(df, x = "infant.deaths", fill = "lightgray", title = "infant.deaths") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```



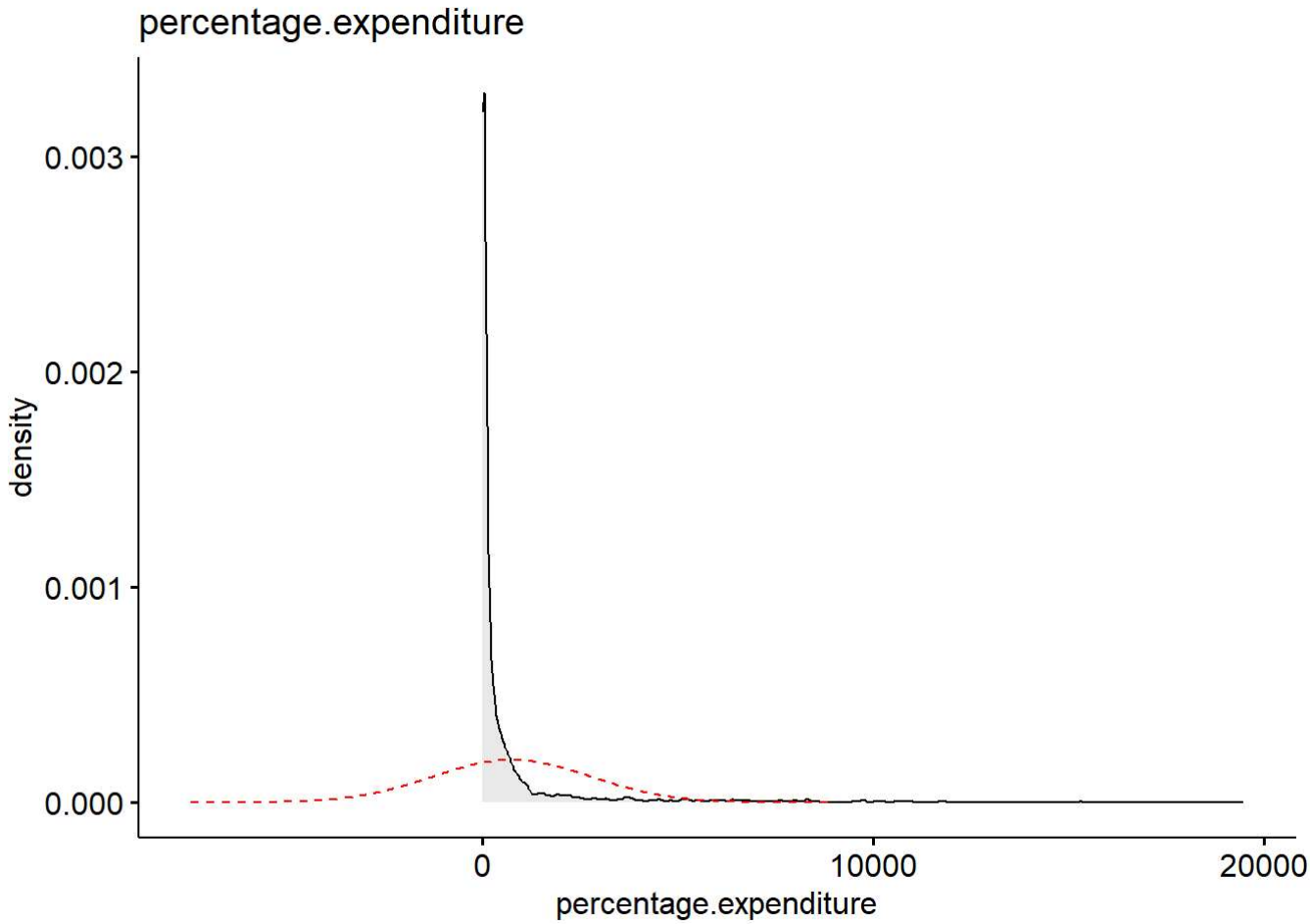

```
ggdensity(df, x = "Alcohol", fill = "lightgray", title = "Alcohol") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

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

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



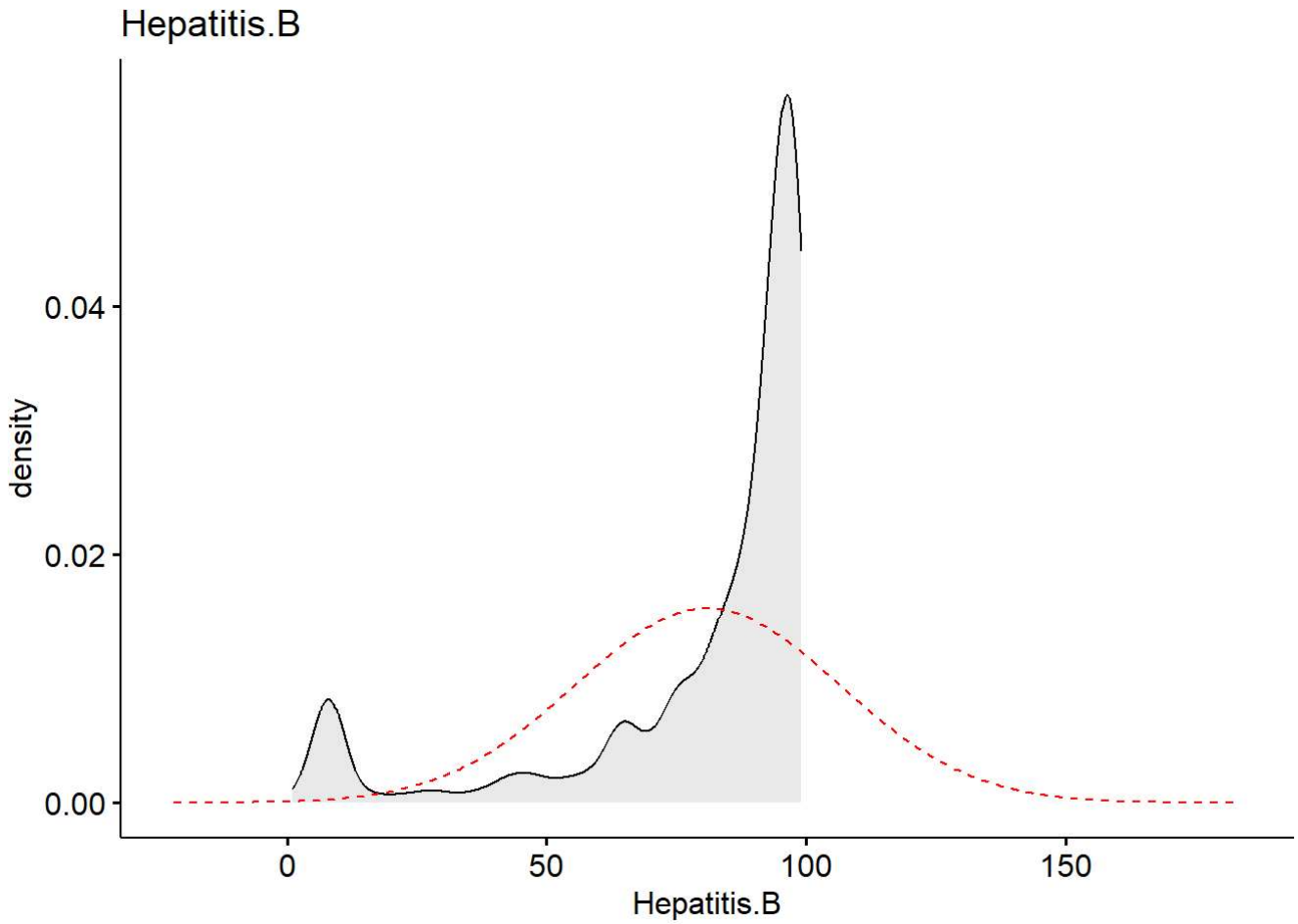

```
ggdensity(df, x = "percentage.expenditure", fill = "lightgray", title = "percentage.expenditur
e") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```



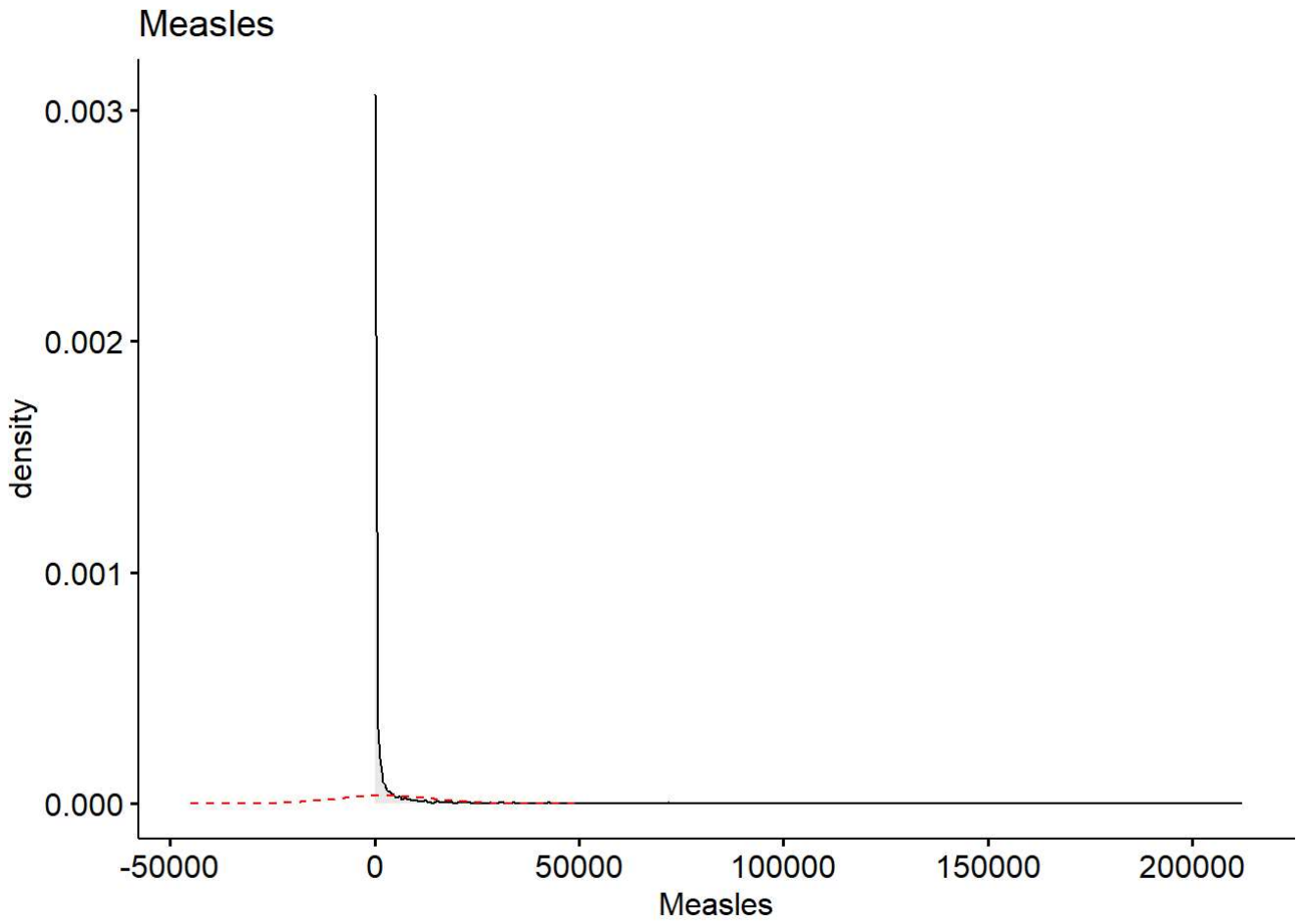

```
ggdensity(df, x = "Hepatitis.B", fill = "lightgray", title = "Hepatitis.B") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

```
## Warning: Removed 553 rows containing non-finite values (`stat_density()`).
```

```
## Warning: Removed 553 rows containing non-finite values
## (`stat_overlay_normal_density()`).
```



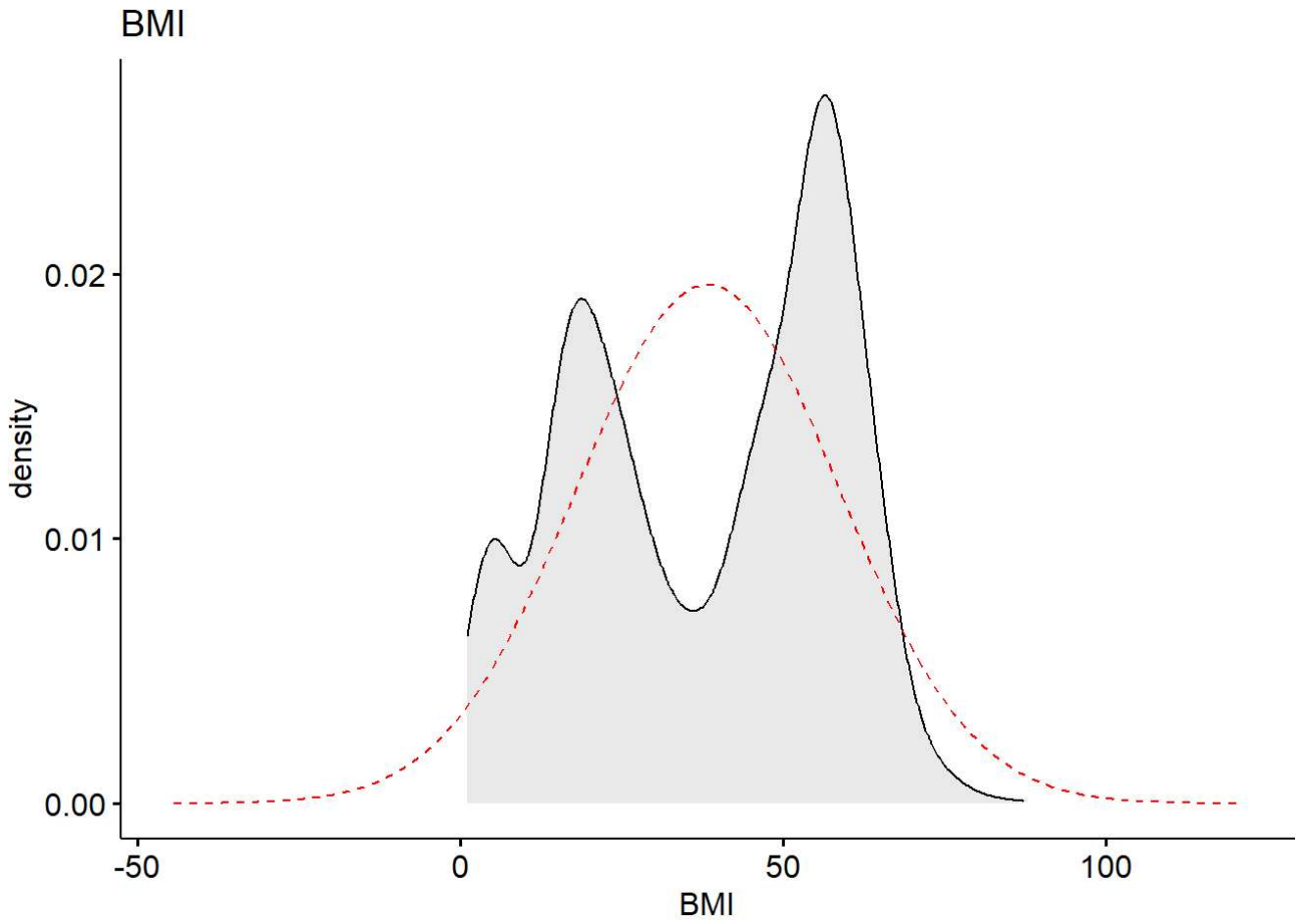

```
ggdensity(df, x = "Measles", fill = "lightgray", title = "Measles") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```



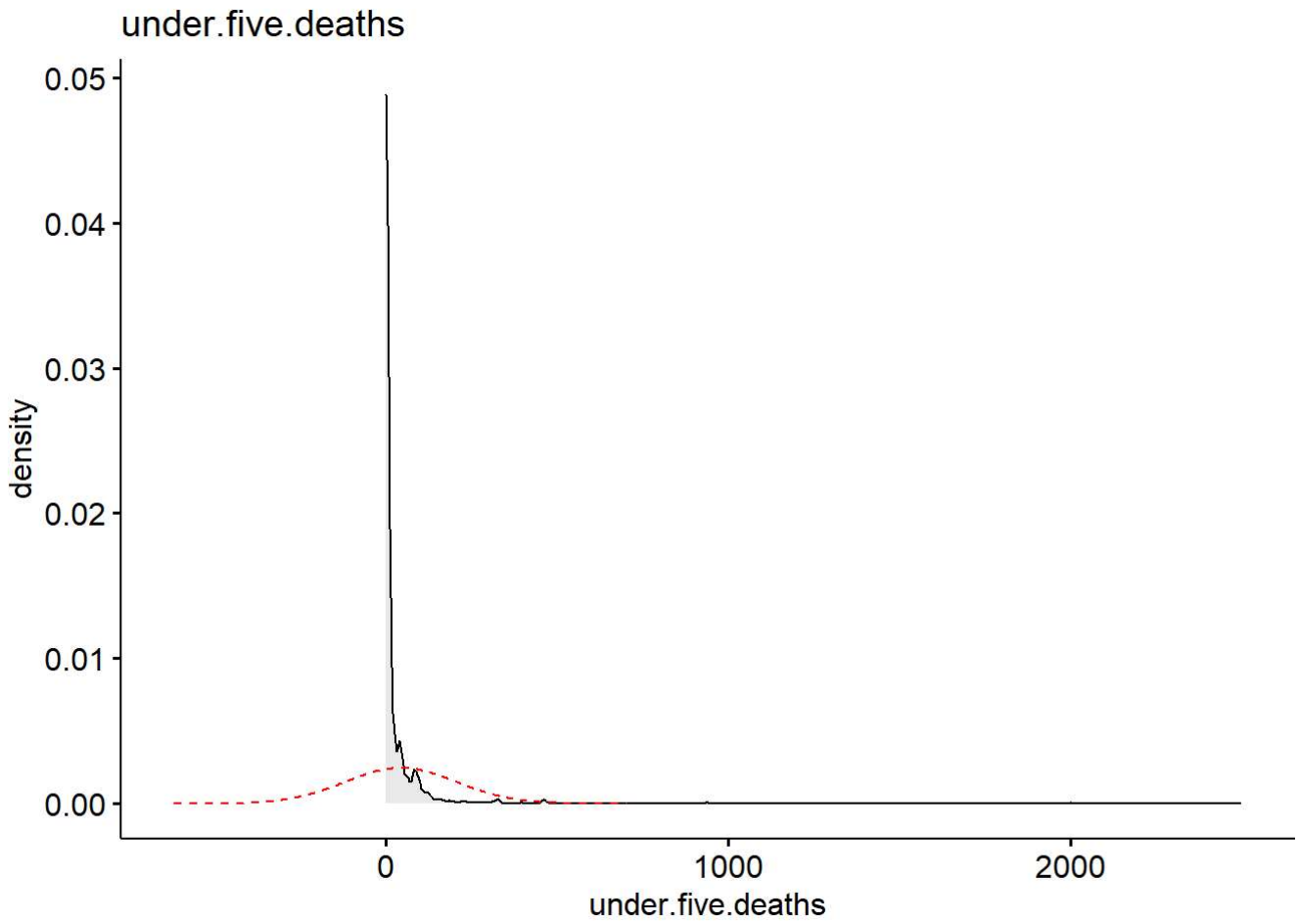

```
ggdensity(df, x = "BMI", fill = "lightgray", title = "BMI") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

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

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



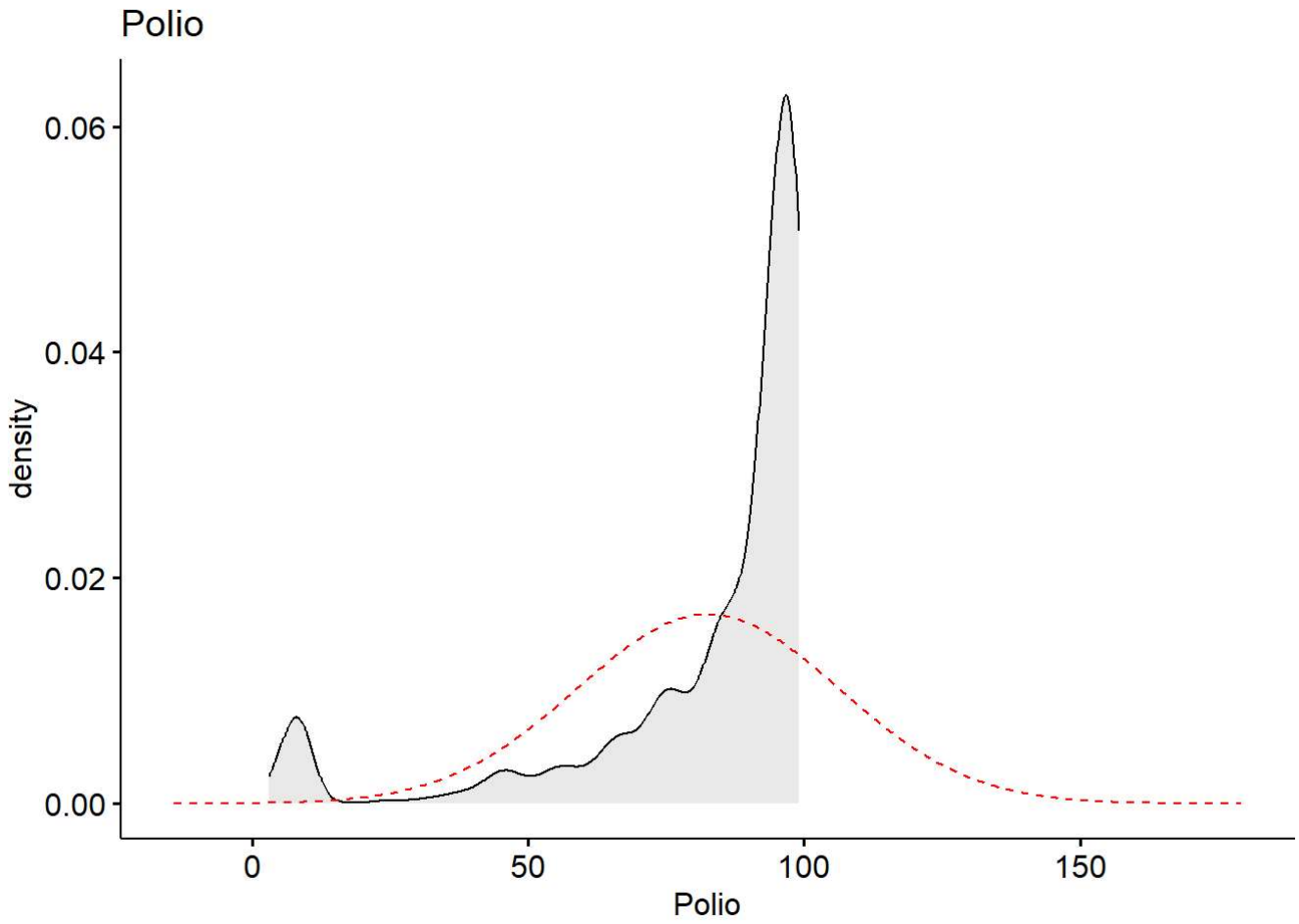

```
ggdensity(df, x = "under.five.deaths", fill = "lightgray", title = "under.five.deaths") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```




```
ggdensity(df, x = "Polio", fill = "lightgray", title = "Polio") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

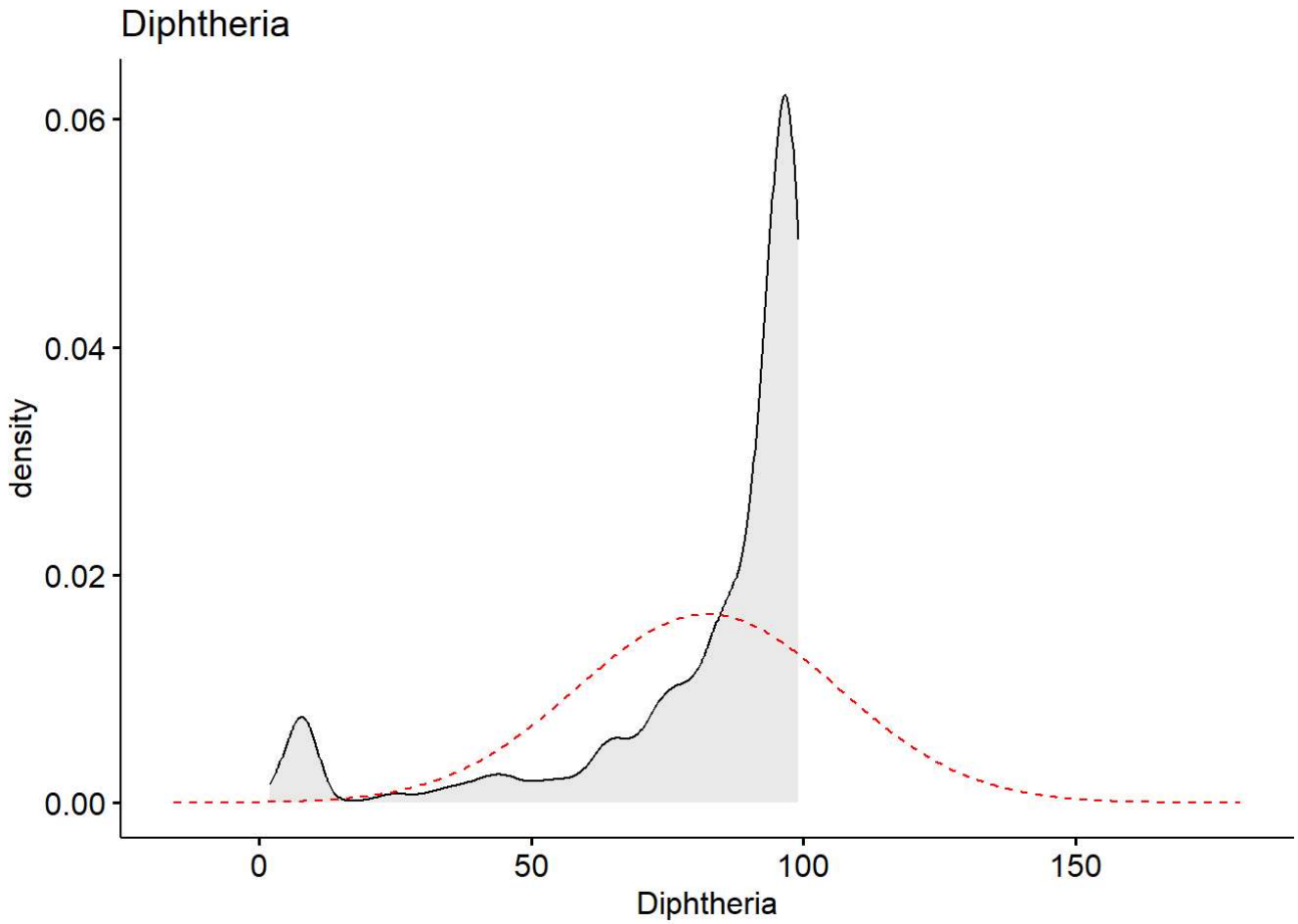
```
## Warning: Removed 19 rows containing non-finite values (`stat_density()`).
```

```
## Warning: Removed 19 rows containing non-finite values
## (`stat_overlay_normal_density()`).
```

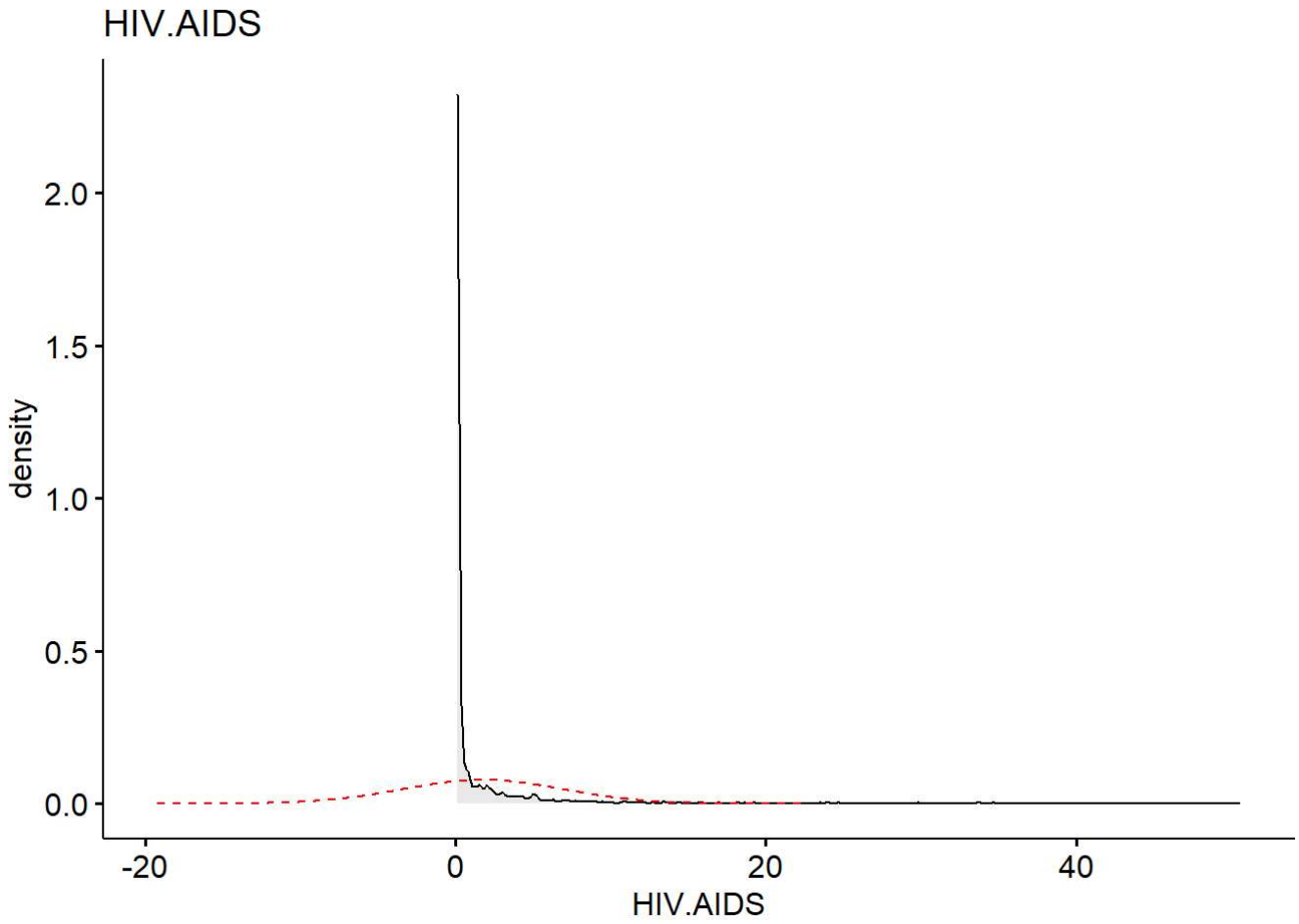



```
ggdensity(df, x = "Diphtheria", fill = "lightgray", title = "Diphtheria") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

```
## Warning: Removed 19 rows containing non-finite values (`stat_density()`).

## Warning: Removed 19 rows containing non-finite values
## (`stat_overlay_normal_density()`).
```



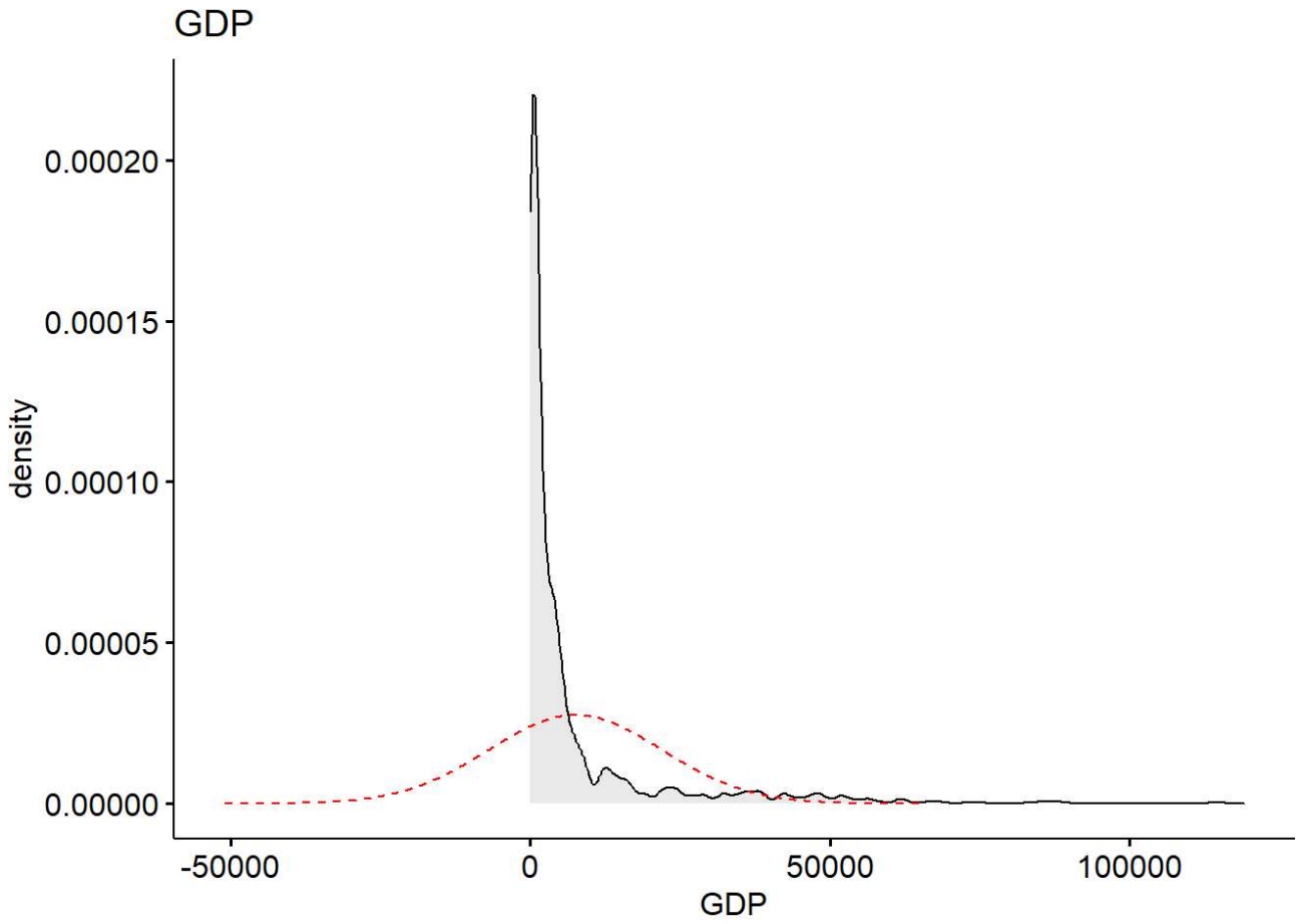

```
ggdensity(df, x = "HIV.AIDS", fill = "lightgray", title = "HIV.AIDS") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```




```
ggdensity(df, x = "GDP", fill = "lightgray", title = "GDP") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

```
## Warning: Removed 448 rows containing non-finite values (`stat_density()`).
```

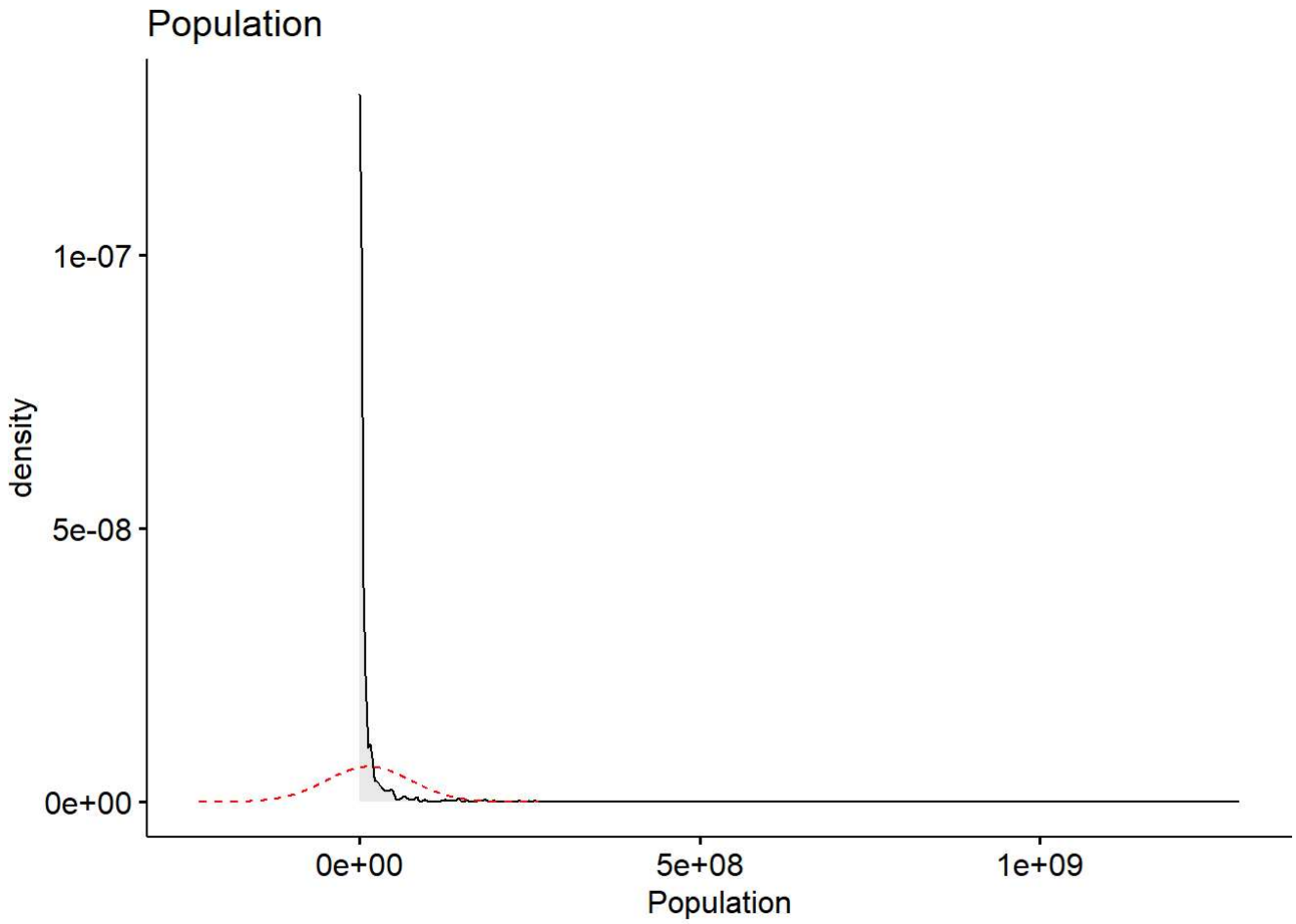
```
## Warning: Removed 448 rows containing non-finite values
## (`stat_overlay_normal_density()`).
```




```
ggdensity(df, x = "Population", fill = "lightgray", title = "Population") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

```
## Warning: Removed 652 rows containing non-finite values (`stat_density()`).
```

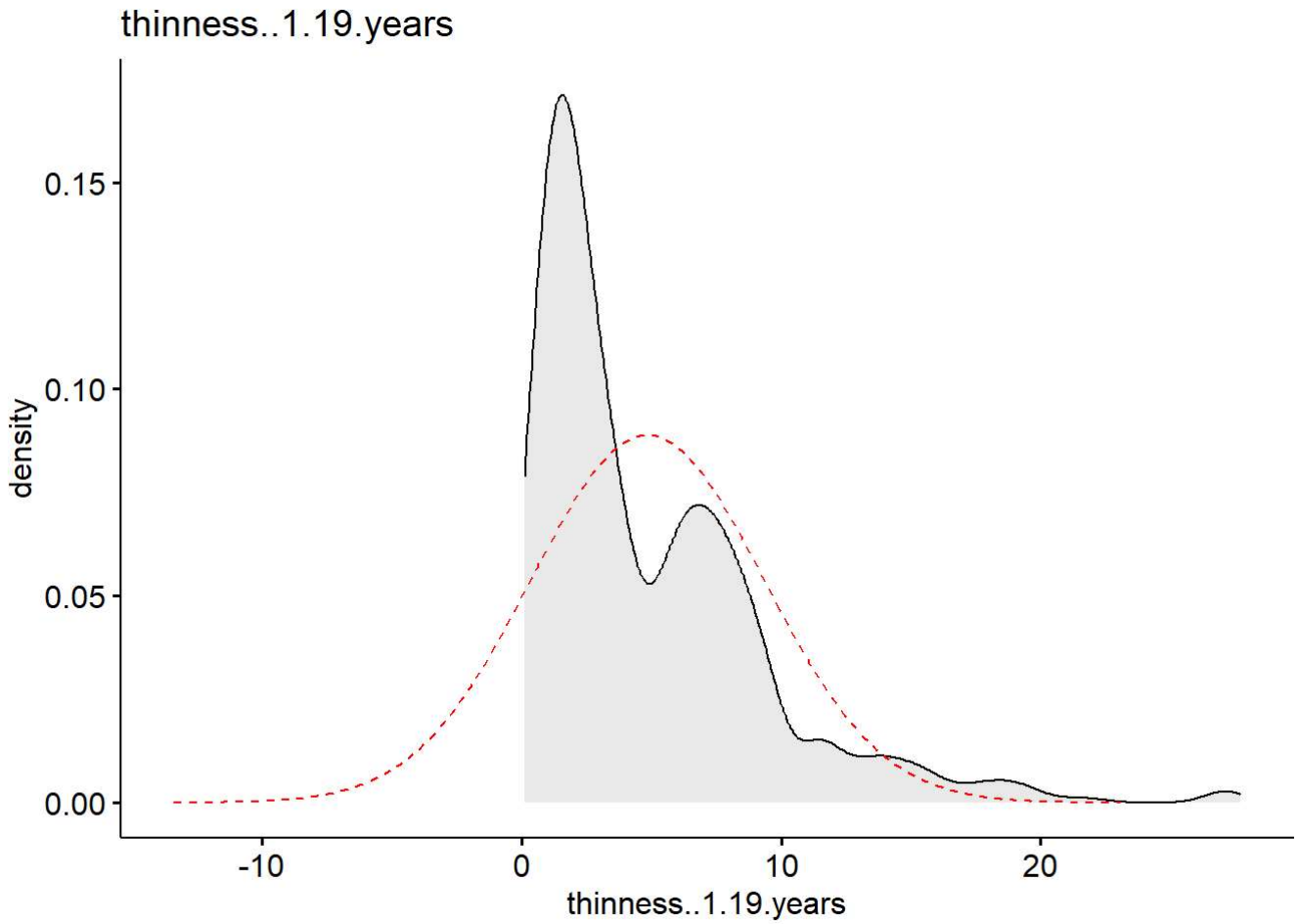
```
## Warning: Removed 652 rows containing non-finite values
## (`stat_overlay_normal_density()`).
```




```
ggdensity(df, x = "thinness..1.19.years", fill = "lightgray", title = "thinness..1.19.years") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

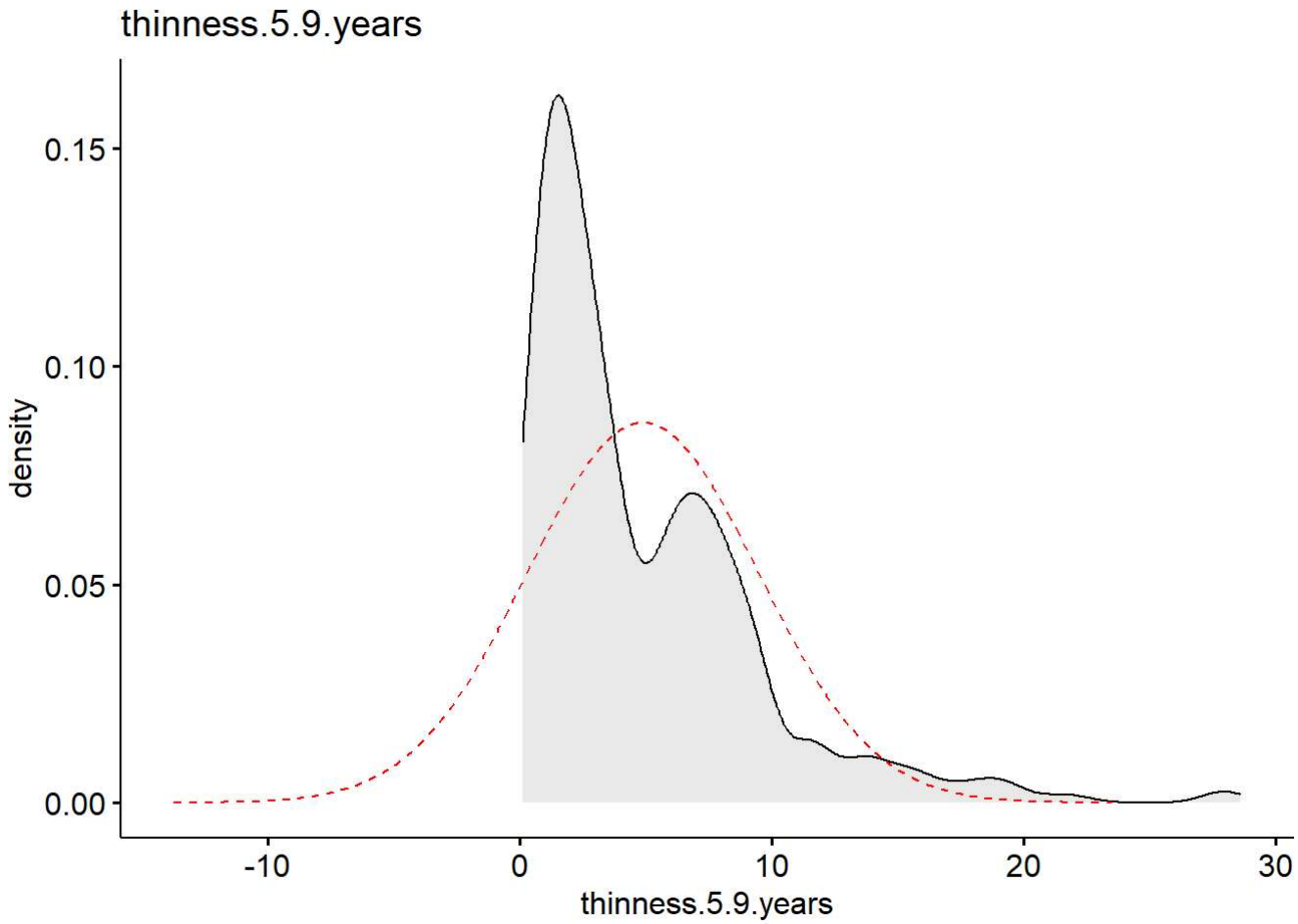
```
## Warning: Removed 34 rows containing non-finite values (`stat_density()`).
```

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




```
ggdensity(df, x = "thinness.5.9.years", fill = "lightgray", title = "thinness.5.9.years") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

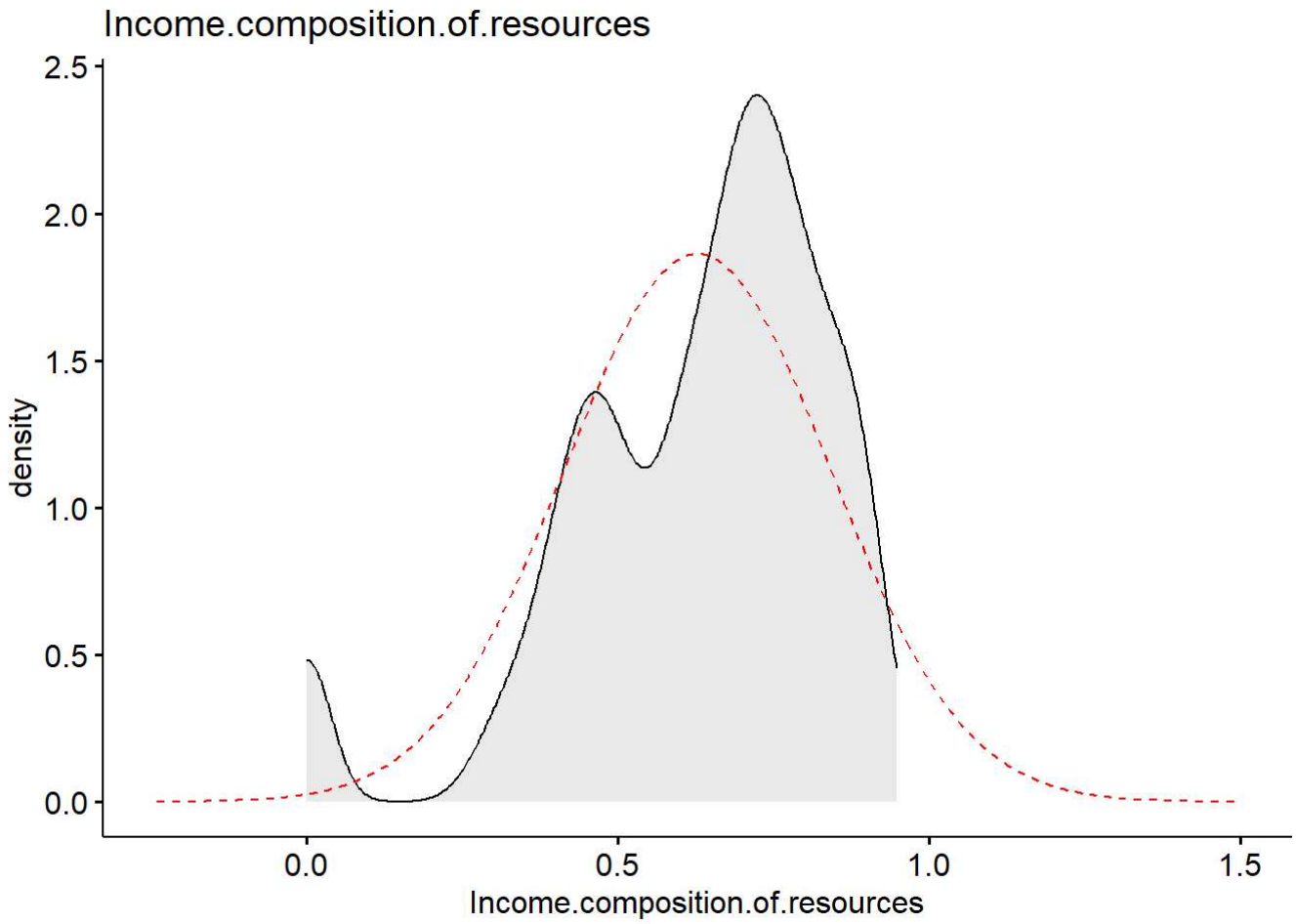
```
## Warning: Removed 34 rows containing non-finite values (`stat_density()`).

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




```
ggdensity(df, x = "Income.composition.of.resources", fill = "lightgray", title = "Income.composi
tion.of.resources") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

```
## Warning: Removed 167 rows containing non-finite values (`stat_density()`).
```

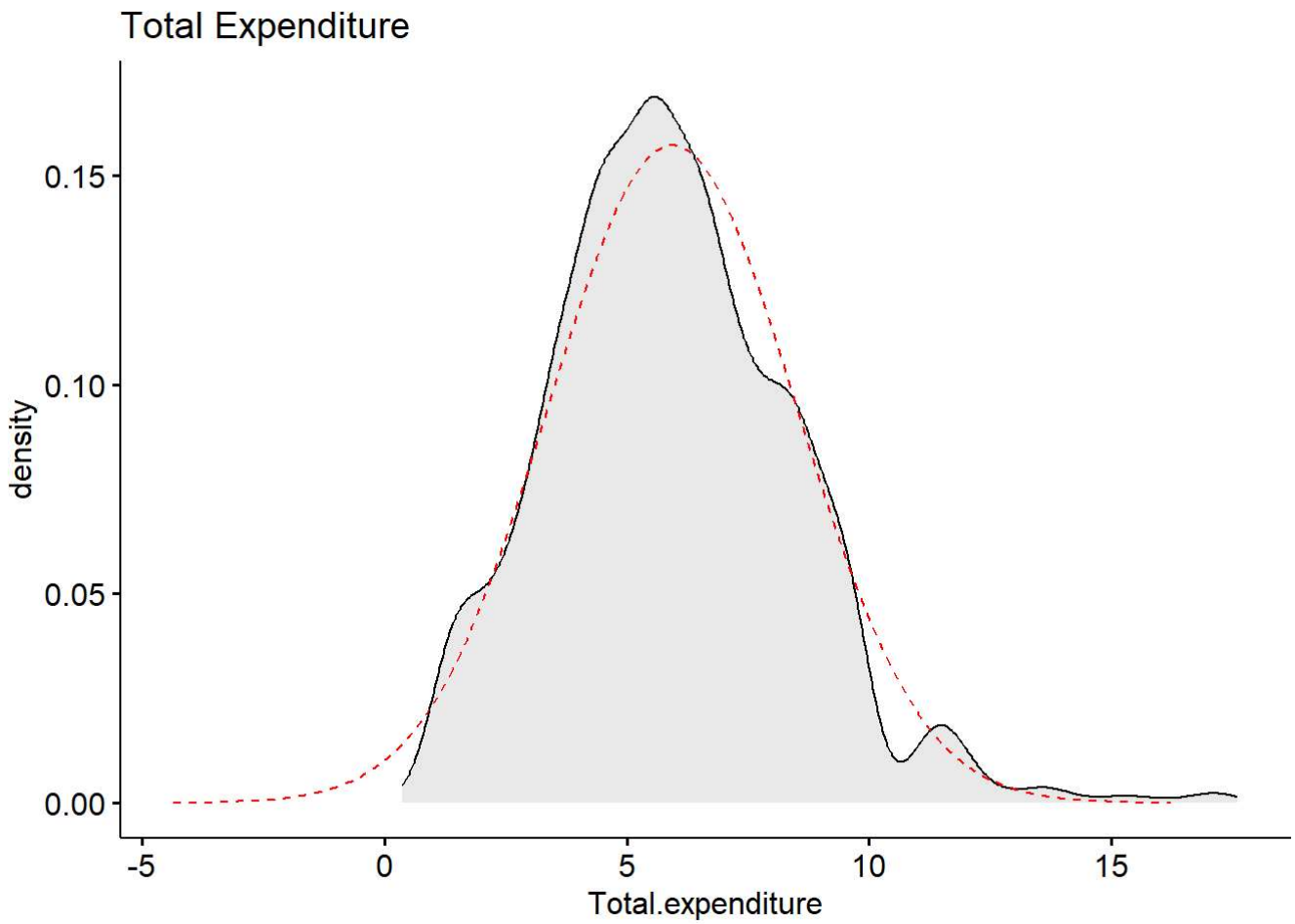
```
## Warning: Removed 167 rows containing non-finite values
## (`stat_overlay_normal_density()`).
```




```
ggdensity(df, x = "Total.expenditure", fill = "lightgray", title = "Total Expenditure") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

```
## Warning: Removed 226 rows containing non-finite values (`stat_density()`).
```

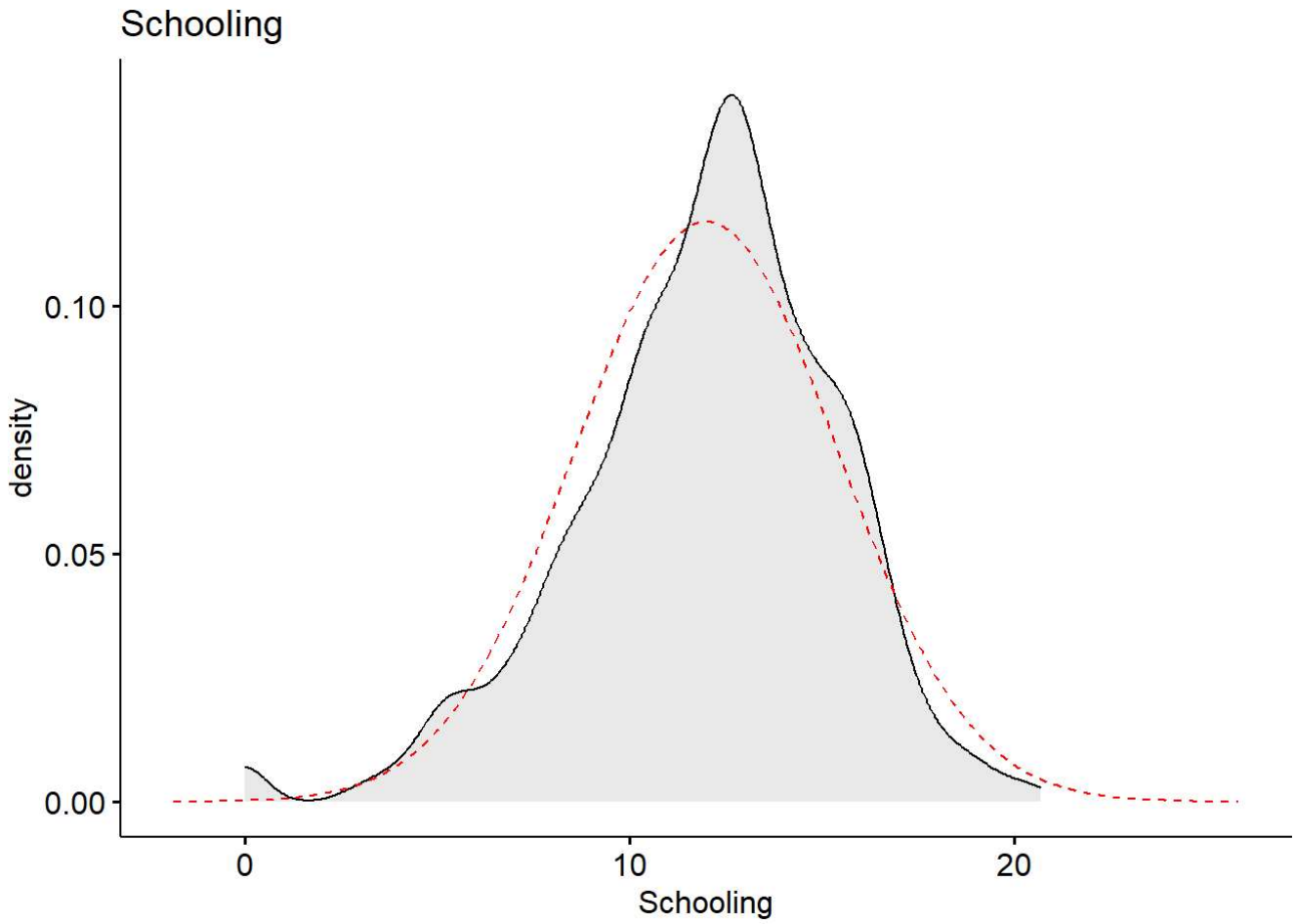
```
## Warning: Removed 226 rows containing non-finite values
## (`stat_overlay_normal_density()`).
```




```
# schooling
ggdensity(df, x = "Schooling", fill = "lightgray", title = "Schooling") +
  scale_x_continuous() +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```

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

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



Based on the boxplots and density plots above, most variables have outliers and are skewed. Therefore, we will use the median to impute missing values.

```
library(moments)
sprintf("Skewness Life.expectancy: [%s]", toString(skewness(df$Life.expectancy, na.rm = TRUE)))
```

```
## [1] "Skewness Life.expectancy: [-0.638277535245317]"
```

```
sprintf("Skewness Adult.Mortality: [%s]", toString(skewness(df$Adult.Mortality, na.rm = TRUE)))
```

```
## [1] "Skewness Adult.Mortality: [1.17376777834786]"
```

```
sprintf("Skewness Alcohol: [%s]", toString(skewness(df$Alcohol, na.rm = TRUE)))
```

```
## [1] "Skewness Alcohol: [0.589240196443008]"
```

```
sprintf("Skewness percentage.expenditure: [%s]", toString(skewness(df$percentage.expenditure, n
a.rm = TRUE)))
```

```
## [1] "Skewness percentage.expenditure: [4.64967589960929]"
```

```
sprintf("Skewness Hepatitis.B: [%s]", toString(skewness(df$Hepatitis.B, na.rm = TRUE)))
```

```
## [1] "Skewness Hepatitis.B: [-1.92963052356202]"
```

```
sprintf("Skewness Measles: [%s]", toString(skewness(df$Measles, na.rm = TRUE)))
```

```
## [1] "Skewness Measles: [9.43651097808561]"
```

```
sprintf("Skewness BMI: [%s]", toString(skewness(df$BMI, na.rm = TRUE)))
```

```
## [1] "Skewness BMI: [-0.21919830637541]"
```

```
sprintf("Skewness under.five.deaths: [%s]", toString(skewness(df$under.five.deaths, na.rm = TRU
E)))
```

```
## [1] "Skewness under.five.deaths: [9.49021625078426]"
```

```
sprintf("Skewness Polio: [%s]", toString(skewness(df$Polio, na.rm = TRUE)))
```

```
## [1] "Skewness Polio: [-2.09697495873492]"
```

```
sprintf("Skewness Population: [%s]", toString(skewness(df$Population, na.rm = TRUE)))
```

```
## [1] "Skewness Population: [15.9057899701332]"
```

```
sprintf("Skewness Diphtheria: [%s]", toString(skewness(df$Diphtheria, na.rm = TRUE)))
```

```
## [1] "Skewness Diphtheria: [-2.07168764210687]"
```

```
sprintf("Skewness HIV.AIDS: [%s]", toString(skewness(df$HIV.AIDS, na.rm = TRUE)))
```

```
## [1] "Skewness HIV.AIDS: [5.39335665878726]"
```

```
sprintf("Skewness thinness.5.9.years: [%s]", toString(skewness(df$thinness.5.9.years, na.rm = TR
UE)))
```

```
## [1] "Skewness thinness.5.9.years: [1.77650575471949]"
```

```
sprintf("Skewness Income.composition.of.resources: [%s]", toString(skewness(df$Income.compositio
n.of.resources, na.rm = TRUE)))
```

```
## [1] "Skewness Income.composition.of.resources: [-1.14314348448666]"
```

Handling Null values

```
Life.expectancy_m <- median(df$Life.expectancy,  na.rm = TRUE)
Adult.Mortality_m <- median(df$Adult.Mortality,  na.rm = TRUE)
infant.deaths_m <- median(df$infant.deaths,  na.rm = TRUE)
Hepatitis.B_m <- median(df$Hepatitis.B,  na.rm = TRUE)
percentage.expenditure_m<-median(df$percentage.expenditure,  na.rm = TRUE)
Measles_m<-median(df$Measles,  na.rm = TRUE)
under.five.deaths_m<-median(df$under.five.deaths,  na.rm = TRUE)
HIV.AIDS_m<-median(df$HIV.AIDS,  na.rm = TRUE)
Polio_m <- median(df$Polio,  na.rm = TRUE)
Diphtheria_m <- median(df$Diphtheria,  na.rm = TRUE)
Total.expenditure_m <- median(df$Total.expenditure,  na.rm = TRUE)
GDP_m <- median(df$GDP,  na.rm = TRUE)
Population_m <- median(df$Population,  na.rm = TRUE)
thinness..1.19.years_m <- median(df$thinness..1.19.years,  na.rm = TRUE)
thinness.5.9.years_m <- median(df$thinness.5.9.years,  na.rm = TRUE)
Schooling_m <- median(df$Schooling,  na.rm = TRUE)
Alcohol_m <- median(df$Alcohol,  na.rm = TRUE)
BMI_m <- median(df$BMI,  na.rm = TRUE)
Income.composition.of.resources_m <- median(df$Income.composition.of.resources,  na.rm = TRUE)
```

```
#Replace the NA with median
df$Life.expectancy[is.na(df$Life.expectancy)] <- Life.expectancy_m
df$Adult.Mortality[is.na(df$Adult.Mortality)] <- Adult.Mortality_m
df$infant.deaths[is.na(df$infant.deaths)] <- infant.deaths_m
df$percentage.expenditure[is.na(df$percentage.expenditure)] <- percentage.expenditure_m
df$Measles[is.na(df$Measles)] <- Measles_m
df$under.five.deaths[is.na(df$under.five.deaths)] <- under.five.deaths_m
df$HIV.AIDS[is.na(df$HIV.AIDS)] <- HIV.AIDS_m
df$Hepatitis.B[is.na(df$Hepatitis.B)] <- Hepatitis.B_m
df$Polio[is.na(df$Polio)] <- Polio_m
df$Diphtheria[is.na(df$Diphtheria)] <- Diphtheria_m
df$Total.expenditure[is.na(df$Total.expenditure)] <- Total.expenditure_m
df$GDP[is.na(df$GDP)] <- GDP_m
df$Population[is.na(df$Population)] <- Population_m
df$thinness..1.19.years[is.na(df$thinness..1.19.years)] <- thinness..1.19.years_m
df$thinness.5.9.years[is.na(df$thinness.5.9.years)] <- thinness.5.9.years_m
df$Schooling[is.na(df$Schooling)] <- Schooling_m
df$Alcohol[is.na(df$Alcohol)] <- Alcohol_m
df$BMI[is.na(df$BMI)] <- BMI_m
df$Income.composition.of.resources[is.na(df$Income.composition.of.resources)] <- Income.composit
ion.of.resources_m
```

```
missing_count <- data.frame(feature = factor(names(df)),
                  counts=sapply(df, function(x) sum(is.na(x))))
missing_count
```

	feature <fct>	counts <int>
Status	Status	0
Life.expectancy	Life.expectancy	0
Adult.Mortality	Adult.Mortality	0
infant.deaths	infant.deaths	0
Alcohol	Alcohol	0
percentage.expenditure	percentage.expenditure	0
Hepatitis.B	Hepatitis.B	0
Measles	Measles	0
BMI	BMI	0
under.five.deaths	under.five.deaths	0

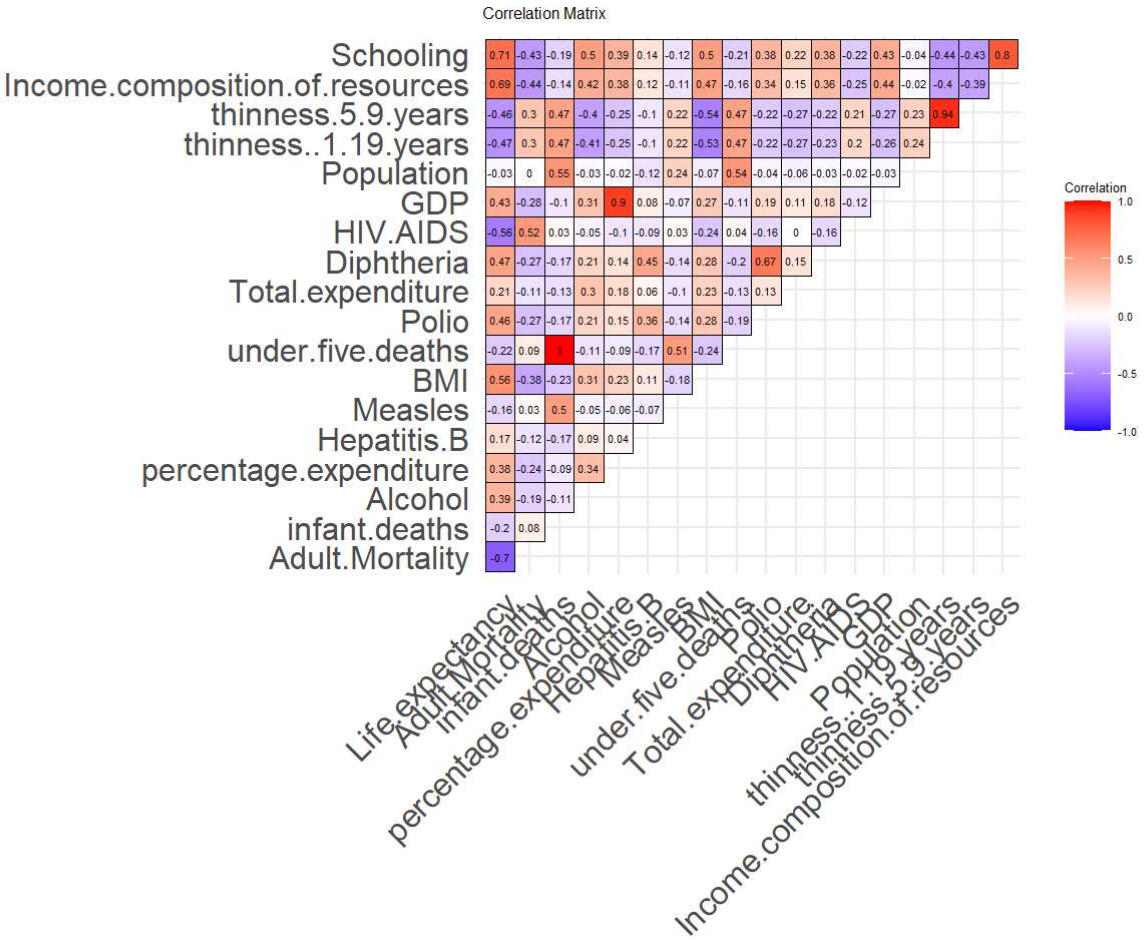
1-10 of 20 rows Previous **1** 2 Next

```
head(df)
```

	Status <chr>	Life.expectancy <dbl>	Adult.Mortality <dbl>	infant.deaths <dbl>	Alcohol <dbl>	percentage.exper
1	Developing	65.0	263	62	0.01	71.2
2	Developing	59.9	271	64	0.01	73.5
3	Developing	59.9	268	66	0.01	73.2
4	Developing	59.5	272	69	0.01	78.1
5	Developing	59.2	275	71	0.01	7.0
6	Developing	58.8	279	74	0.01	79.6

6 rows | 1-7 of 21 columns

```
library(ggcorrplot)

set_plot_dimensions(40,35)
corr <- round(cor(subset(df, select =-c(Status))),3)
ggcorrplot(corr,type = "upper", lab = TRUE, outline.color = "black", lab_size = 1.5,
           legend.title ="Correlation")+ ggtitle("Correlation Matrix")+theme(text = element_text
(size = 5))
```



Based on the correlation calculation, we can see that:

1. under_five_deaths and infant_deaths have a high correlation, with a correlation coefficient of more than 0.8.
2. GDP and total_expenditure have a high correlation, with a correlation coefficient of more than 0.8.
3. thinness_5_9_years and thinness_5_9_years have a high correlation, with a correlation coefficient of more than 0.8.
4. Schooling has a high correlation with Income_composition_of_resources and life_expectancy, with a correlation coefficient of more than 0.7.

This suggests that there is a strong linear relationship between these variables.

Since the correlation coefficients are all above 0.8, it is possible that there is collinearity between these variables. Collinearity is a statistical condition in which two or more predictor variables are highly correlated with each other. Collinearity can cause problems for statistical models, such as making it difficult to interpret the results of the model and increasing the risk of overfitting.

Collinearity

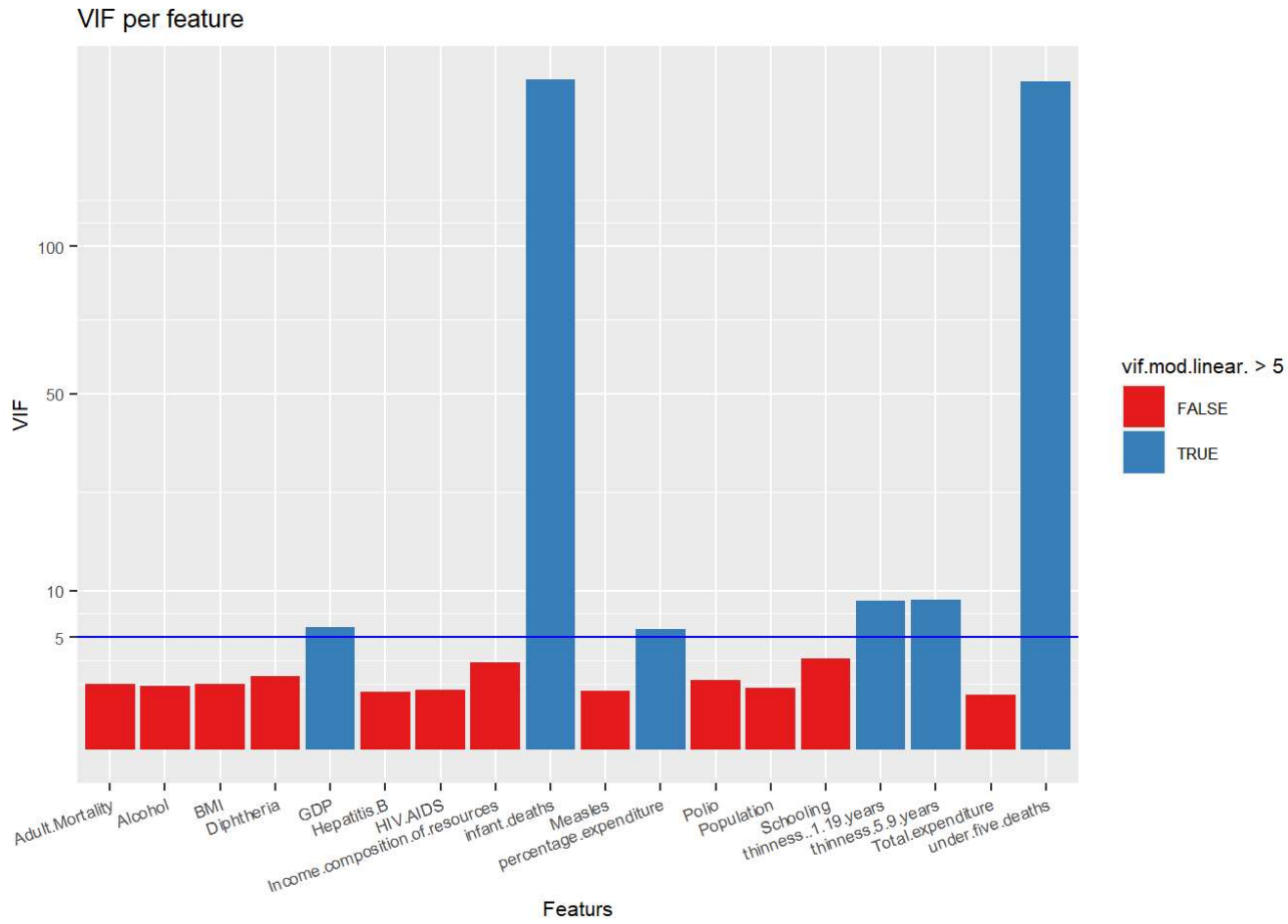
```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
mod.linear <- lm(Life.expectancy~ ., data = subset(df, select =-c(Status)))
vifs <- data.frame(vif(mod.linear))

set_plot_dimensions(16,8)
ggplot(vifs, aes(y=vif.mod.linear., x=row.names(vifs))) +
    geom_bar(aes(fill=vif.mod.linear.>5),stat="identity")+
    scale_y_continuous(trans = "sqrt",  breaks = c(5, 10, 50, 100))+
    geom_hline(yintercept = 5, colour = "blue") +
    ggtitle("VIF per feature") +
    xlab("Featurs") + ylab("VIF") +
    theme(axis.text.x=element_text(angle=20, hjust=1))+
    theme(text = element_text(size = 8))+
    scale_fill_brewer(palette="Set1")
```

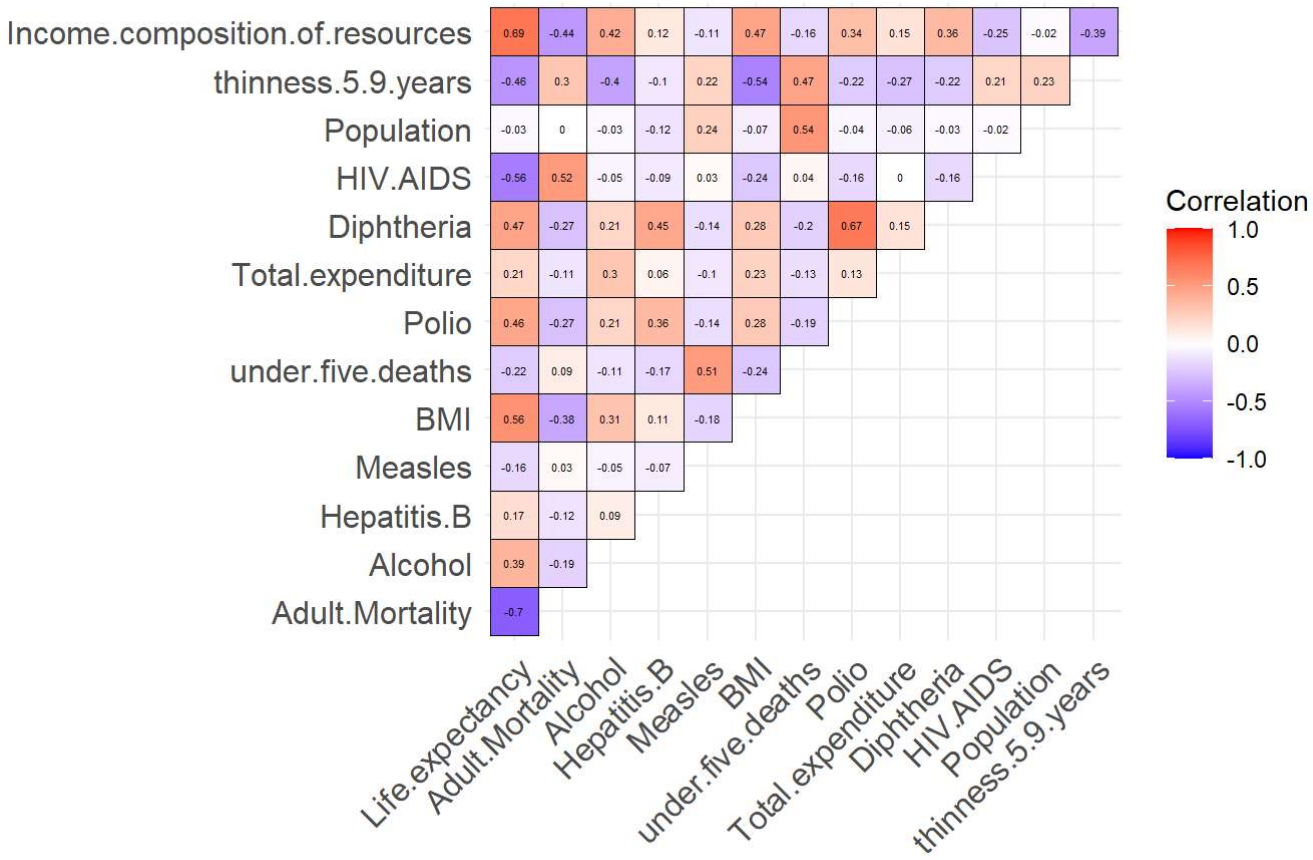


As expected, the variables GDP, infant.deaths, percentage.expenditure, thinness.5.9.years, thinness.5.9.years, and under.five.deaths have high VIF (more than 5). This means that these variables are highly correlated with each other, which can cause problems for machine learning algorithms.

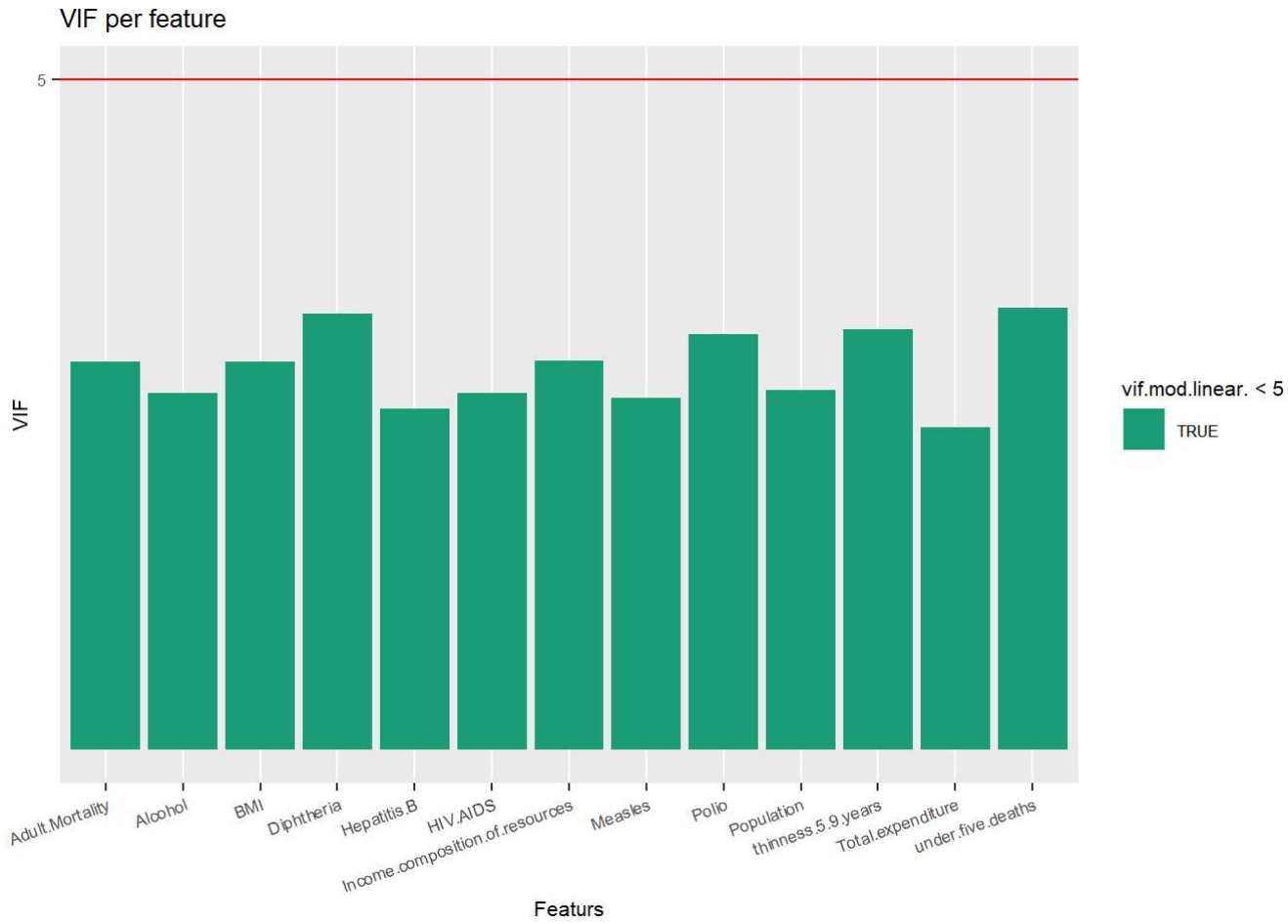
To address this issue, we will drop the variables infant.deaths, GDP, and thinness.5.9.years. This will reduce the collinearity in the data and improve the performance of the machine learning algorithm.

```
#Omit infant.deaths, GDP and thinness..1.19.years
df <- subset(df, select = -c(infant.deaths,GDP,thinness..1.19.years,Schooling,percentage.expendi
ture))
```

```
#Checking correlation again
set_plot_dimensions(16,10)
corr <- round(cor(subset(df, select =-c(Status))), 3)
ggcorrplot(corr,type = "upper", lab = TRUE, outline.color = "black", lab_size = 1.5, legend.titl
e = "Correlation")
```




```
#Check VIF again
mod.linear <- lm(Life.expectancy~ ., data = subset(df, select =-c(Status)))
vifs <- data.frame(vif(mod.linear))

set_plot_dimensions(16,8)
ggplot(vifs, aes(y=vif.mod.linear., x=row.names(vifs))) +
    geom_bar(aes(fill=vif.mod.linear.<5),stat="identity")+
    scale_y_continuous(trans = "sqrt",  breaks = c(5, 10, 50, 100))+
    geom_hline(yintercept = 5, colour = "red") +
    ggtitle("VIF per feature") +
    xlab("Featurs") + ylab("VIF") +
    theme(axis.text.x=element_text(angle=20, hjust=1))+
    theme(text = element_text(size = 8))+
    scale_fill_brewer(palette="Dark2")
```



It ca be seen that, the collinearity of all the variables are normal

#Transformation

Some of the variables are skewed, while others are bimodal (except for schooling and total expenditure). We will perform a Box-Cox transformation to transform them into a normal distribution.

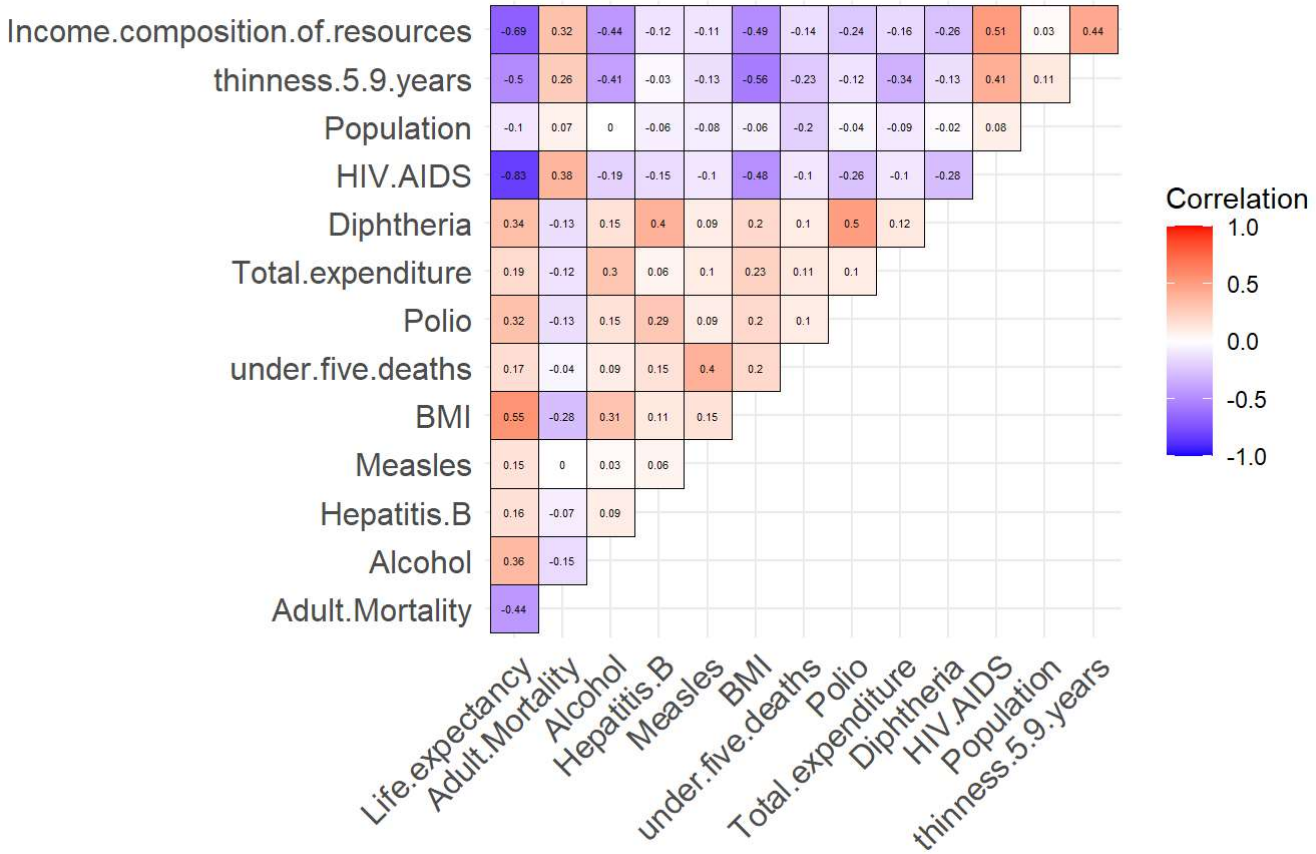
```
df$Life.expectancy<- log(df$Life.expectancy)
df$Adult.Mortality<- log(df$Adult.Mortality)
df$Measles <- sqrt(max(df$Measles+1)- df$Measles)
df$under.five.deaths <- sqrt(max(df$under.five.deaths+1)- df$under.five.deaths)
df$Population<- log(df$Population)
df$Diphtheria<- log(df$Diphtheria)
df$HIV.AIDS<- log(df$HIV.AIDS)
df$thinness.5.9.years<- log(df$thinness.5.9.years)
df$Income.composition.of.resources<- sqrt(max(df$Income.composition.of.resources+1)-df$Income.co
mposition.of.resources)

df$Polio<- log(df$Polio)
```

```
head(df)
```

Status <chr>	Life.expectancy <dbl>	Adult.Mortality <dbl>	Alcohol <dbl>	Hepatitis.B <int>	Measles <dbl>	BMI <dbl>	unde
1 Developing	4.174387	5.572154	0.01	65	459.3800	19.1	
2 Developing	4.092677	5.602119	0.01	62	460.1000	18.6	
3 Developing	4.092677	5.590987	0.01	64	460.1674	18.1	
4 Developing	4.085976	5.605802	0.01	67	457.5992	17.6	
5 Developing	4.080922	5.616771	0.01	68	457.3522	17.2	
6 Developing	4.074142	5.631212	0.01	66	458.4703	16.7	

6 rows | 1-9 of 16 columns

```
#Checking correlation again
set_plot_dimensions(16,10)
corr <- round(cor(subset(df, select =-c(Status))), 3)
ggcorrplot(corr,type = "upper", lab = TRUE, outline.color = "black", lab_size = 1.5, legend.titl
e = "Correlation")
```




```
sprintf("Skewness Life.expectancy: [%s]", toString(skewness(df$Life.expectancy, na.rm = TRUE)))
```

```
## [1] "Skewness Life.expectancy: [-0.951703811182817]"
```

```
sprintf("Skewness Adult.Mortality: [%s]", toString(skewness(df$Adult.Mortality, na.rm = TRUE)))
```

```
## [1] "Skewness Adult.Mortality: [-1.26762927176649]"
```

```
sprintf("Skewness Alcohol: [%s]", toString(skewness(df$Alcohol, na.rm = TRUE)))
```

```
## [1] "Skewness Alcohol: [0.649246459881017]"
```

```
sprintf("Skewness Hepatitis.B: [%s]", toString(skewness(df$Hepatitis.B, na.rm = TRUE)))
```

```
## [1] "Skewness Hepatitis.B: [-2.28053233848223]"
```

```
sprintf("Skewness Measles: [%s]", toString(skewness(df$Measles, na.rm = TRUE)))
```

```
## [1] "Skewness Measles: [-14.1043457121923]"
```

```
sprintf("Skewness BMI: [%s]", toString(skewness(df$BMI, na.rm = TRUE)))
```

```
## [1] "Skewness BMI: [-0.229039825519374]"
```

```
sprintf("Skewness under.five.deaths: [%s]", toString(skewness(df$under.five.deaths, na.rm = TRU
E)))
```

```
## [1] "Skewness under.five.deaths: [-12.5700235235759]"
```

```
sprintf("Skewness Polio: [%s]", toString(skewness(df$Polio, na.rm = TRUE)))
```

```
## [1] "Skewness Polio: [-3.28364467404516]"
```

```
sprintf("Skewness Population: [%s]", toString(skewness(df$Population, na.rm = TRUE)))
```

```
## [1] "Skewness Population: [-0.701699857165913]"
```

```
sprintf("Skewness Diphtheria: [%s]", toString(skewness(df$Diphtheria, na.rm = TRUE)))
```

```
## [1] "Skewness Diphtheria: [-3.21855961845011]"
```

```
sprintf("Skewness HIV.AIDS: [%s]", toString(skewness(df$HIV.AIDS, na.rm = TRUE)))
```

```
## [1] "Skewness HIV.AIDS: [1.28273332892996]"
```

```
sprintf("Skewness thinness.5.9.years: [%s]", toString(skewness(df$thinness.5.9.years, na.rm = TR
UE)))
```

```
## [1] "Skewness thinness.5.9.years: [-0.649812699966731]"
```

```
head(df)
```

Status <chr>	Life.expectancy <dbl>	Adult.Mortality <dbl>	Alcohol <dbl>	Hepatitis.B <int>	Measles <dbl>	BMI <dbl>	unde
1 Developing	4.174387	5.572154	0.01	65	459.3800	19.1	
2 Developing	4.092677	5.602119	0.01	62	460.1000	18.6	
3 Developing	4.092677	5.590987	0.01	64	460.1674	18.1	
4 Developing	4.085976	5.605802	0.01	67	457.5992	17.6	
5 Developing	4.080922	5.616771	0.01	68	457.3522	17.2	
6 Developing	4.074142	5.631212	0.01	66	458.4703	16.7	

6 rows | 1-9 of 16 columns

```
# Calculate the number of NaN values in the data frame
num_nan <- sum(is.nan(df$Income.composition.of.resources))

# Print the number of NaN values
print(num_nan)
```

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

```
#Scalling

df$Life.expectancy<- scale(df$Life.expectancy, scale=TRUE, center = TRUE)
df$Adult.Mortality<- scale(df$Adult.Mortality, scale=TRUE, center = TRUE)
df$Hepatitis.B<- scale(df$Hepatitis.B, scale=TRUE, center = TRUE)
df$Measles<- scale(df$Measles, scale=TRUE, center = TRUE)
df$BMI<- scale(df$BMI, scale=TRUE, center = TRUE)
df$under.five.deaths<- scale(df$under.five.deaths, scale=TRUE, center = TRUE)
df$Polio<- scale(df$Polio, scale=TRUE, center = TRUE)
df$Population<- scale(df$Population, scale=TRUE, center = TRUE)
df$Diphtheria<- scale(df$Diphtheria, scale=TRUE, center = TRUE)
df$thinness.5.9.years<- scale(df$Population, scale=TRUE, center = TRUE)
df$Total.expenditure<- scale(df$Total.expenditure, scale=TRUE, center = TRUE)
```

#Feature Selection

Feature selection is the process of selecting the most important features from a dataset for use in a machine learning model. There are many different feature selection methods, but two of the most common are backward and forward selection. This project will use backward and forward selection to select the most important features from the dataset for use in the machine learning model.

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.5
```

```
regfit.bwd <- suppressWarnings(regsubsets(Life.expectancy~.,data=df,nvmax=16,method="backward"))
```

```
## Reordering variables and trying again:
```

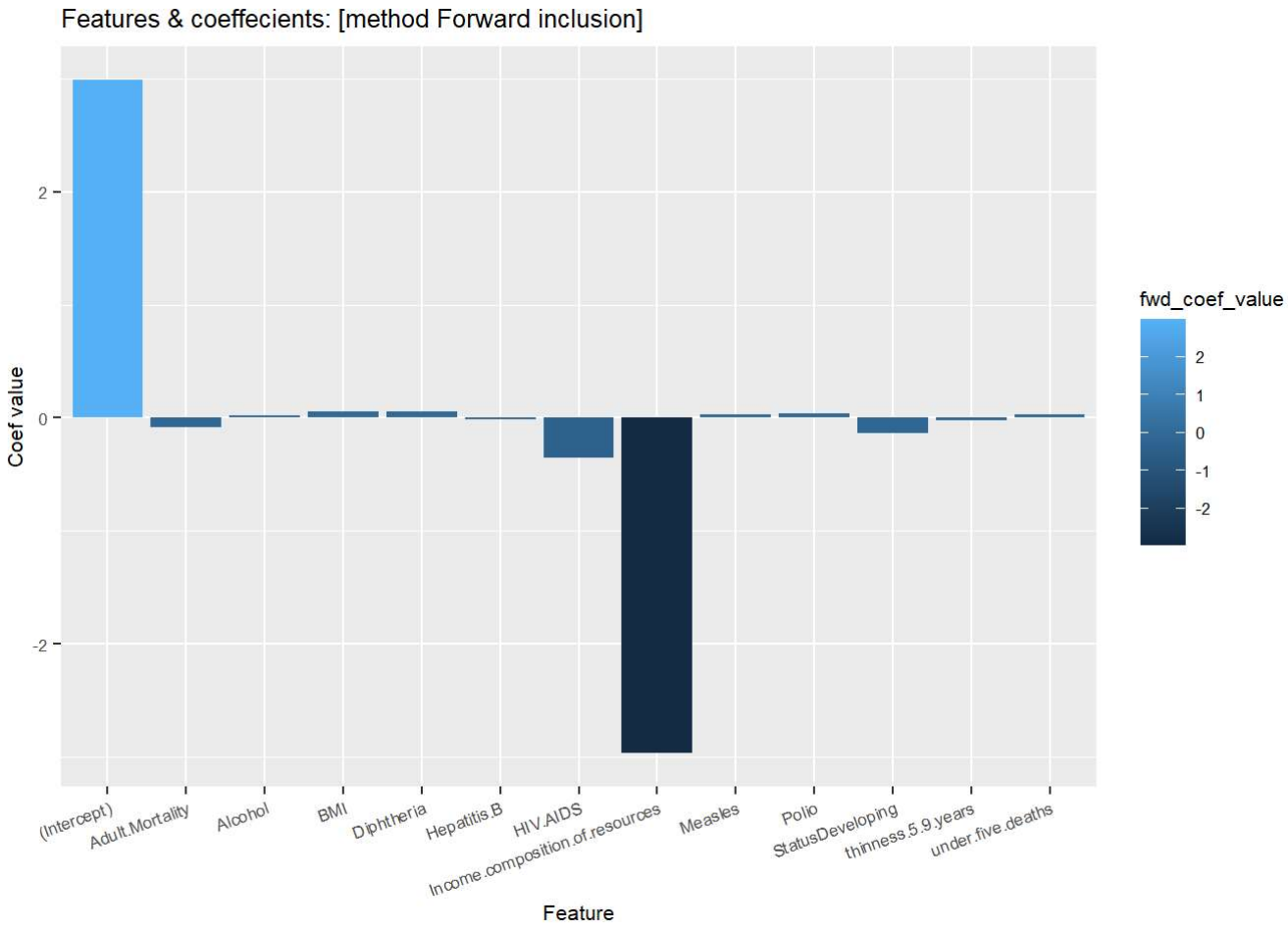
```
bwd.summary <- summary(regfit.bwd)
```

```
regfit.fwd <- suppressWarnings(regsubsets(Life.expectancy~.,data=df,nvmax=16,method="forward"))
```

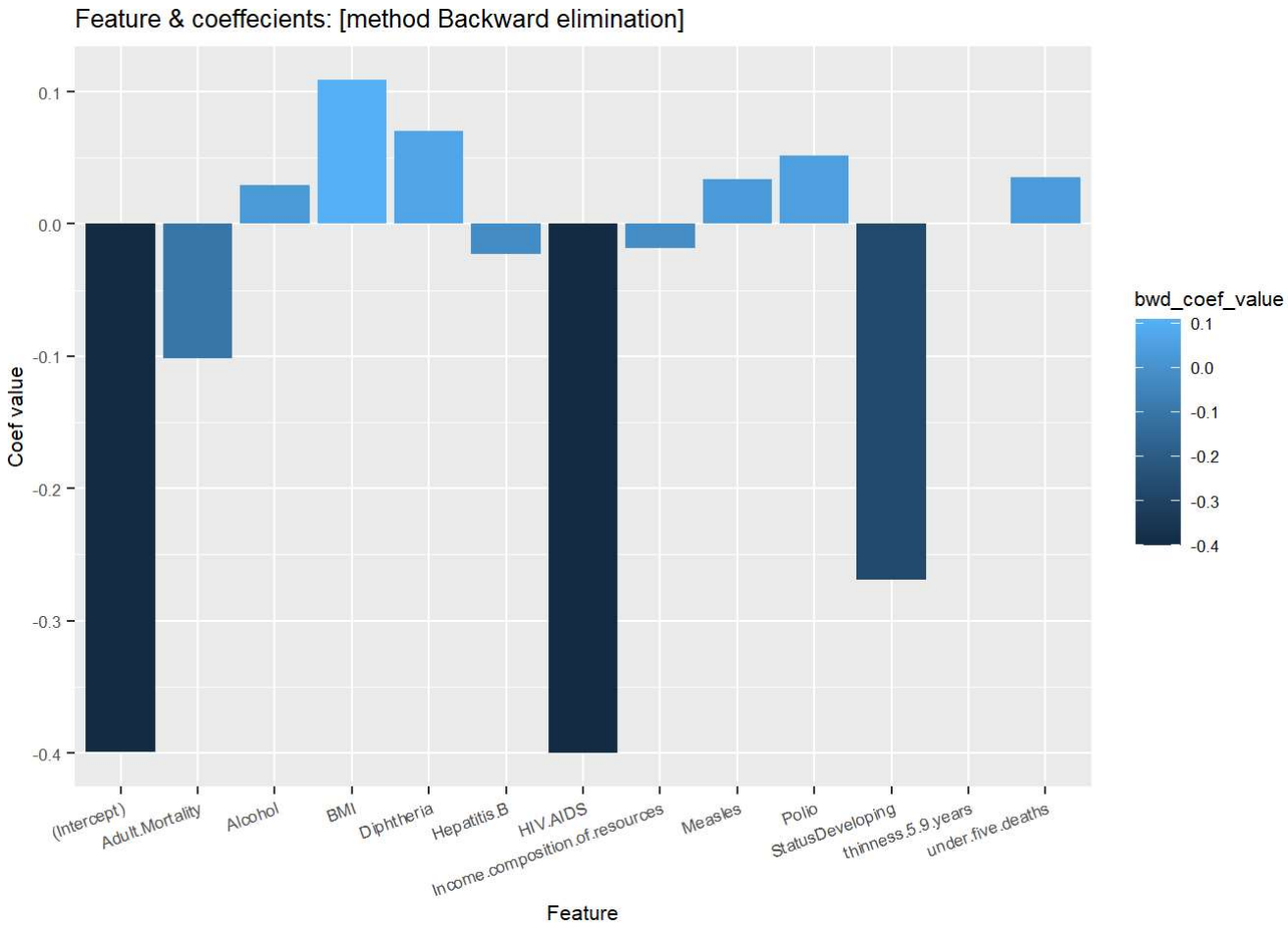
```
## Reordering variables and trying again:
```

```
fwd.summary <-summary(regfit.fwd)
```

```
v_names <- rownames(as.data.frame(coef(regfit.fwd,12)))
coefs<- data.frame(v_names)
coefs$fwd_coef_value <-  coef(regfit.fwd,12)
coefs$bwd_coef_value <-  coef(regfit.bwd,12)

set_plot_dimensions(18,4)
ggplot(coefs,
       aes(x=v_names, y=fwd_coef_value, fill=fwd_coef_value)) +
                                  geom_bar(stat="identity") +
                                  ggtitle("Features & coeffecients: [method Forward inclusion]")
+
                                  xlab("Feature") + ylab("Coef value") +
                                  theme(axis.text.x=element_text(angle=20, hjust=1))+
                                  theme(text = element_text(size = 8))
```




```
ggplot(coefs,
       aes(x=v_names, y=bwd_coef_value, fill=bwd_coef_value)) +
                                  geom_bar(stat="identity") +
                                  ggtitle("Feature & coeffecients: [method Backward eliminatio
n]") +
                                  xlab("Feature") + ylab("Coef value") +
                                  theme(axis.text.x=element_text(angle=20, hjust=1))+
                                  theme(text = element_text(size = 8))
```



Based on the plot above choose the same 12 variables. In the next step we will perform linear regression on full model and on the 12 variables from feature selection.

```
#Split train and test data

sample <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.70,0.30))
train <- df[sample, ]
x.test <-df[!sample, ]
y.test <- df[!sample, ]$Life.expectancy
```

1. Linear Regression with all variables

```
model1<- lm(Life.expectancy~., data = train)
summary(model1)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2846 -0.2211  0.0090  0.2452  2.1184
##
## Coefficients: (1 not defined because of singularities)
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.964596   0.177131  16.737  < 2e-16 ***
## StatusDeveloping               -0.120833   0.033842  -3.571 0.000364 ***
## Adult.Mortality                -0.081040   0.010669  -7.596 4.63e-14 ***
## Alcohol                         0.016680   0.003208   5.199 2.21e-07 ***
## Hepatitis.B                    -0.019266   0.010769  -1.789 0.073760 .
## Measles                         0.030417   0.012491   2.435 0.014974 *
## BMI                             0.053658   0.012152   4.416 1.06e-05 ***
## under.five.deaths               0.027910   0.010978   2.542 0.011082 *
## Polio                           0.046822   0.011563   4.049 5.33e-05 ***
## Total.expenditure              -0.003078   0.010339  -0.298 0.765937
## Diphtheria                      0.046094   0.012396   3.719 0.000206 ***
## HIV.AIDS                       -0.360138   0.007704 -46.747  < 2e-16 ***
## Population                     -0.018919   0.009900  -1.911 0.056143 .
## thinness.5.9.years                   NA         NA      NA       NA
## Income.composition.of.resources -2.951022   0.151041 -19.538  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4374 on 2045 degrees of freedom
## Multiple R-squared:  0.8097, Adjusted R-squared:  0.8085
## F-statistic: 669.3 on 13 and 2045 DF,  p-value: < 2.2e-16
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.0.5
```

```
pred <- predict(model1, newdata=x.test)
```

```
## Warning in predict.lm(model1, newdata = x.test): prediction from a rank-
## deficient fit may be misleading
```

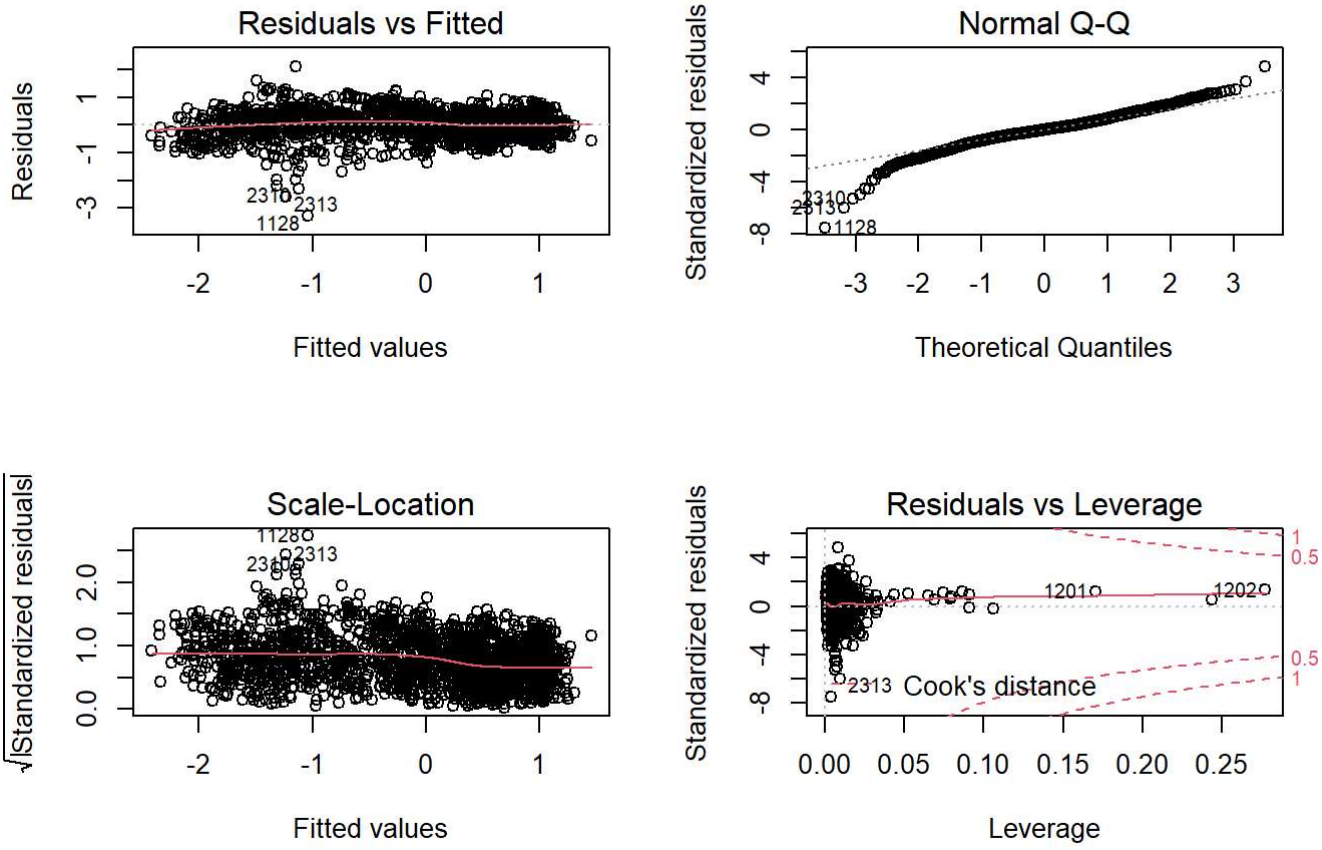
```
rmse_mod1=rmse(pred,y.test)
cat("RMSE model 1: ", rmse_mod1)
```

```
## RMSE model 1:  0.4352913
```

```
adj_rsqrd_mod1=summary(model1)$adj.r.squared
cat("    Adj R Squared model 1: ",adj_rsqrd_mod1)
```

```
##     Adj R Squared model 1:  0.808475
```

```
par(mfrow=c(2,2))
plot(model1)
```




```
df1 <- subset(df, select=c(Life.expectancy,Status, Adult.Mortality,
                                    Hepatitis.B, Polio, BMI, thinness.5.9.years, Measles,
                                    Diphtheria,HIV.AIDS,
                           Income.composition.of.resources, under.five.deaths))

#Split train and test data

sample <- sample(c(TRUE, FALSE), nrow(df1), replace=TRUE, prob=c(0.70,0.30))
train1 <- df1[sample, ]
x.test1 <-df1[!sample, ]
y.test1 <- df1[!sample, ]$Life.expectancy
```

```
model2<- lm(Life.expectancy~., data = train1)
summary(model2)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ ., data = train1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2795 -0.2204  0.0008  0.2360  1.6540
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.383912   0.166002  20.385  < 2e-16 ***
## StatusDeveloping               -0.206668   0.029043  -7.116 1.53e-12 ***
## Adult.Mortality                -0.074072   0.010421  -7.108 1.62e-12 ***
## Hepatitis.B                    -0.016483   0.010471  -1.574  0.11560
## Polio                           0.045688   0.011372   4.018 6.09e-05 ***
## BMI                             0.078488   0.011395   6.888 7.52e-12 ***
## thinness.5.9.years             -0.017315   0.009742  -1.777  0.07566 .
## Measles                         0.029233   0.010521   2.779  0.00551 **
## Diphtheria                      0.054056   0.011867   4.555 5.54e-06 ***
## HIV.AIDS                       -0.346496   0.007474 -46.358  < 2e-16 ***
## Income.composition.of.resources -3.177163   0.147775 -21.500  < 2e-16 ***
## under.five.deaths               0.035061   0.011544   3.037  0.00242 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4321 on 2038 degrees of freedom
## Multiple R-squared:  0.8133, Adjusted R-squared:  0.8123
## F-statistic: 806.9 on 11 and 2038 DF,  p-value: < 2.2e-16
```

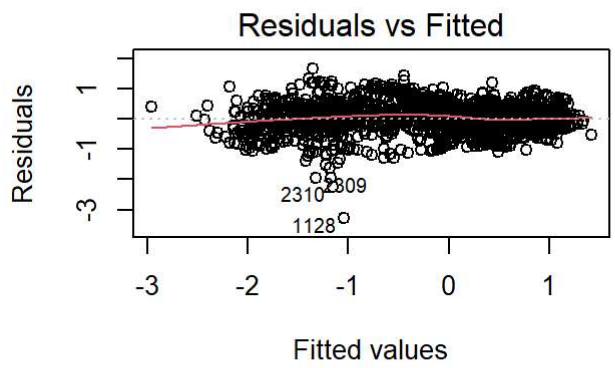
```
library(Metrics)
pred <- predict(model2, newdata=x.test1)
rmse_mod2=rmse(pred,y.test1)
cat("RMSE model 2: ",rmse_mod2)
```

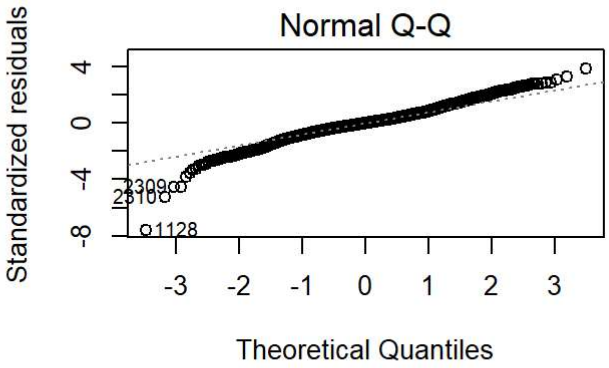
```
## RMSE model 2:  0.4566364
```

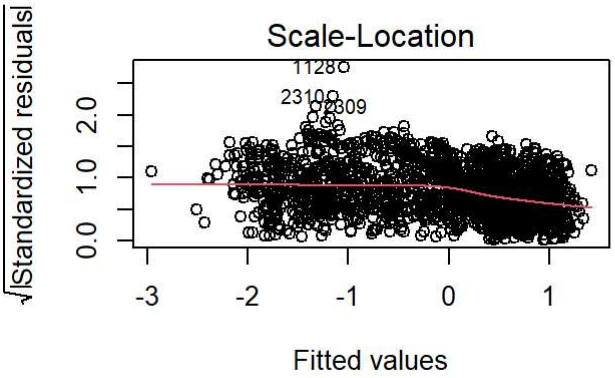
```
adj_rsqrd_mod2=summary(model2)$adj.r.squared
cat("    Adj R Squared model 2: ",  adj_rsqrd_mod2)
```

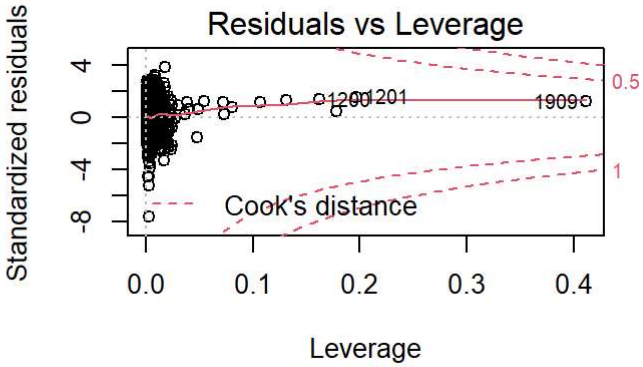
```
##     Adj R Squared model 2:  0.8122556
```

```
par(mfrow=c(2,2))
plot(model2)
```










```
# Create a list of vectors
vector_list <- list(
  Model = c("Model 1 (all Variables)", "Model 2 (11 variables)"),
  RMSE = c(0.4466444, 0.4294861),
  Adj_R_Squared = c(0.8111984, 0.8031346)
)

# Create a data frame from the list of vectors
df <- data.frame(vector_list)

# Print the data frame
print(df)
```

```
##                    Model      RMSE Adj_R_Squared
## 1 Model 1 (all Variables) 0.4466444     0.8111984
## 2  Model 2 (11 variables) 0.4294861     0.8031346
```

We will choose Model 2 because it has a lower RMSE and a higher adjusted R-squared score, and it uses fewer variables. This means that Model 2 is more accurate and parsimonious than Model 1. Additionally, Model 2 has an accuracy of 80.3%, which is considered to be good.