Project V2

Kishan

2023-04-15

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

##Load the Packages

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.0     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.1     ✓ tibble    3.1.8
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the  ]8;;http://conflicted.r-lib.org/ conflicted package ]8;;  to force all conflicts t
o become errors
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
#install.packages("psych")
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.2.3
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:scales':
##
##     alpha, rescale
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

##Load the Data

```
hd_data <- read.csv("C:/Users/megha/OneDrive/Desktop/Machine Learning Project/Heart_Disease_Dat
a.csv")
head(hd_data)
```

```
##   HeartDisease   BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth
## 1           No 16.60     Yes              No     No              3           30
## 2           No 20.34      No              No    Yes              0            0
## 3           No 26.58     Yes              No     No             20           30
## 4           No 24.21      No              No     No              0            0
## 5           No 23.71      No              No     No             28            0
## 6          Yes 28.87     Yes              No     No              6            0
##   DiffWalking    Sex AgeCategory  Race Diabetic PhysicalActivity GenHealth
## 1          No Female       55-59 White      Yes              Yes Very good
## 2          No Female 80 or older White       No              Yes Very good
## 3          No   Male       65-69 White      Yes              Yes      Fair
## 4          No Female       75-79 White       No               No      Good
## 5         Yes Female       40-44 White       No              Yes Very good
## 6         Yes Female       75-79 Black       No               No      Fair
##   SleepTime Asthma KidneyDisease SkinCancer
## 1         5    Yes            No        Yes
## 2         7     No            No         No
## 3         8    Yes            No         No
## 4         6     No            No        Yes
## 5         8     No            No         No
## 6        12     No            No         No
```

Data Description and Information

#No of tuples and attributes in the data set

```
n_rows <- nrow(hd_data)
n_cols <- ncol(hd_data)
cat("The dataset has", n_rows, "rows and", n_cols, "columns.")
```

```
## The dataset has 319795 rows and 18 columns.
```

#Names of attributes present in the data set

```
names(hd_data)
```

```
##  [1] "HeartDisease"     "BMI"              "Smoking"          "AlcoholDrinking"
##  [5] "Stroke"           "PhysicalHealth"   "MentalHealth"     "DiffWalking"
##  [9] "Sex"              "AgeCategory"      "Race"             "Diabetic"
## [13] "PhysicalActivity" "GenHealth"        "SleepTime"        "Asthma"
## [17] "KidneyDisease"    "SkinCancer"
```

#Finding missing Data

```
missing_values <- sum(is.na(hd_data))
cat("The dataset has", missing_values, "missing values.")
```

```
## The dataset has 0 missing values.
```

#Display Variables and their data types

```
# Create a tibble with variable names and data types
variable_info <- tibble(
  Variable = names(hd_data),
  Type = sapply(hd_data, class)
)

# View the variable names and their data types in table format
variable_info
```

```
## # A tibble: 18 × 2
##    Variable         Type
##    <chr>            <chr>
##  1 HeartDisease     character
##  2 BMI              numeric
##  3 Smoking          character
##  4 AlcoholDrinking  character
##  5 Stroke           character
##  6 PhysicalHealth   integer
##  7 MentalHealth     integer
##  8 DiffWalking      character
##  9 Sex              character
## 10 AgeCategory      character
## 11 Race             character
## 12 Diabetic         character
## 13 PhysicalActivity character
## 14 GenHealth        character
## 15 SleepTime        integer
## 16 Asthma           character
## 17 KidneyDisease    character
## 18 SkinCancer       character
```

#Remove Duplicates

```
heart_disease_data <- unique(hd_data)
cat("The dataset now has", nrow(hd_data), "rows after removing duplicates.")
```

```
## The dataset now has 319795 rows after removing duplicates.
```

##Age is a categorical variable so I am converting it into continuous variable

```
# Define the encoding for AgeCategory
mean_AgeCategory <- c('55-59'=57, '80 or older'=80, '65-69'=67,
                        '75-79'=77,'40-44'=42,'70-74'=72,'60-64'=62,
                        '50-54'=52,'45-49'=47,'18-24'=21,'35-39'=37,
                        '30-34'=32,'25-29'=27)

# Apply the encoding to AgeCategory
hd_data$AgeCategory <- mean_AgeCategory[hd_data$AgeCategory]

# Convert AgeCategory to numeric
hd_data$AgeCategory <- as.numeric(hd_data$AgeCategory)
```

```
head(hd_data)
```

```
##   HeartDisease   BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth
## 1           No 16.60     Yes              No     No              3           30
## 2           No 20.34      No              No    Yes              0            0
## 3           No 26.58     Yes              No     No             20           30
## 4           No 24.21      No              No     No              0            0
## 5           No 23.71      No              No     No             28            0
## 6          Yes 28.87     Yes              No     No              6            0
##   DiffWalking    Sex AgeCategory  Race Diabetic PhysicalActivity GenHealth
## 1          No Female          57 White      Yes              Yes Very good
## 2          No Female          80 White       No              Yes Very good
## 3          No   Male          67 White      Yes              Yes      Fair
## 4          No Female          77 White       No               No      Good
## 5         Yes Female          42 White       No              Yes Very good
## 6         Yes Female          77 Black       No               No      Fair
##   SleepTime Asthma KidneyDisease SkinCancer
## 1         5    Yes            No        Yes
## 2         7     No            No         No
## 3         8    Yes            No         No
## 4         6     No            No        Yes
## 5         8     No            No         No
## 6        12     No            No         No
```

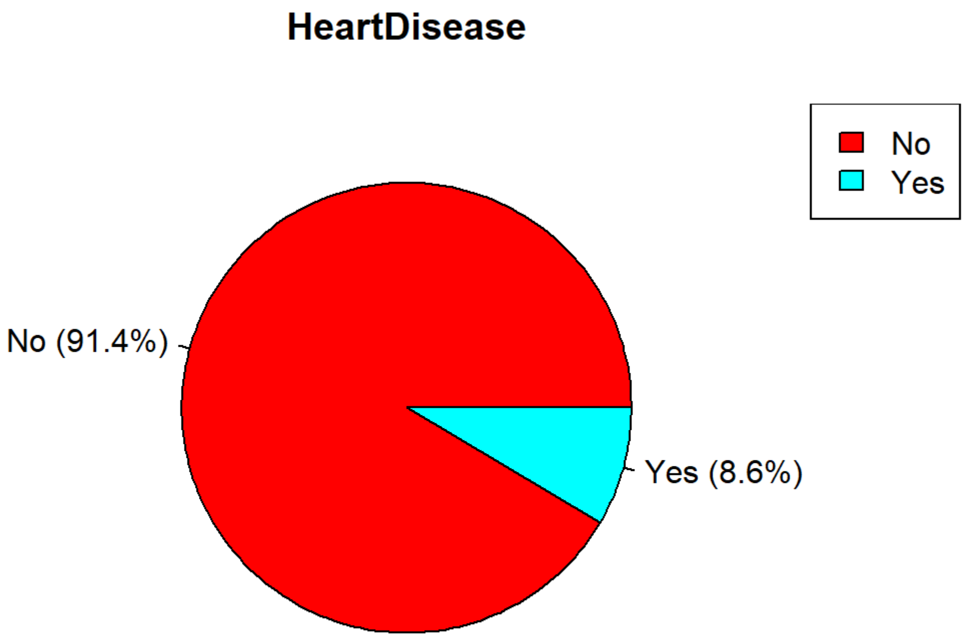
##Statistics for Numerical Data

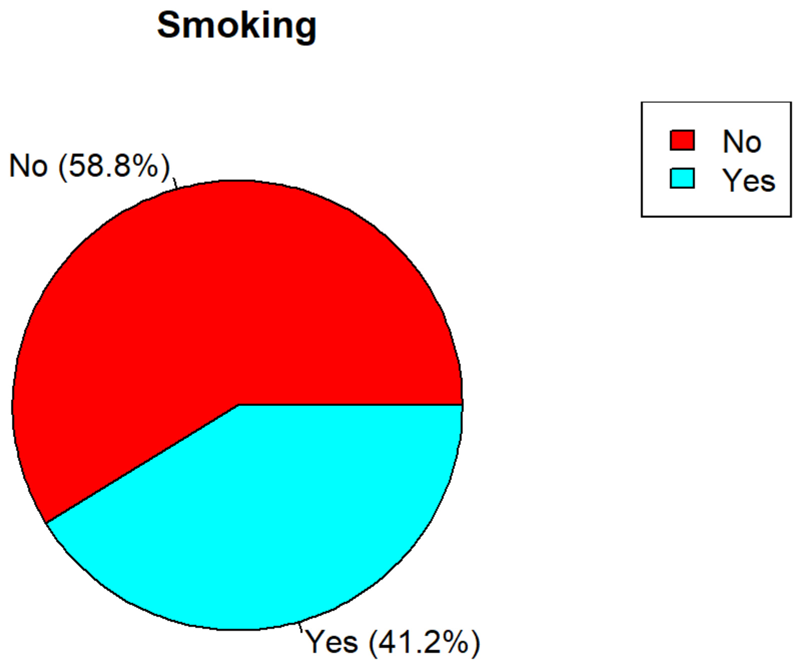
```
library(psych)

cols_to_describe <- c("BMI", "PhysicalHealth", "MentalHealth", "AgeCategory", "SleepTime")
hd_data[cols_to_describe] %>% describe()
```

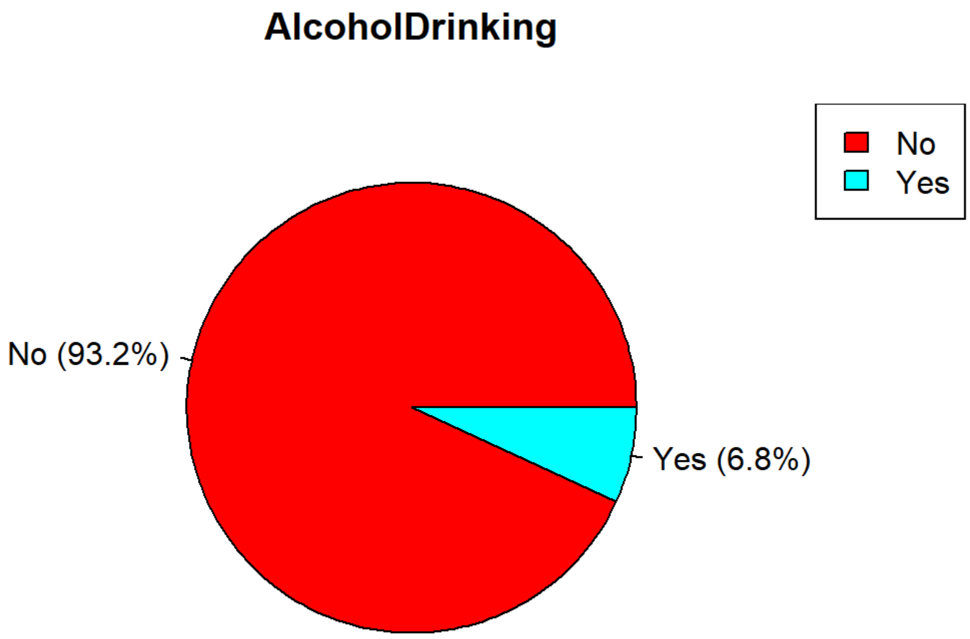
```
##                vars      n  mean    sd median trimmed   mad   min   max range
## BMI               1 319795 28.33  6.36  27.34   27.71  5.43 12.02 94.85 82.83
## PhysicalHealth    2 319795  3.37  7.95   0.00    1.02  0.00  0.00 30.00 30.00
## MentalHealth      3 319795  3.90  7.96   0.00    1.73  0.00  0.00 30.00 30.00
## AgeCategory       4 319795 54.36 17.72  57.00   55.16 22.24 21.00 80.00 59.00
## SleepTime         5 319795  7.10  1.44   7.00    7.11  1.48  1.00 24.00 23.00
##                 skew kurtosis   se
## BMI             1.33     3.89 0.01
## PhysicalHealth  2.60     5.53 0.01
## MentalHealth    2.33     4.40 0.01
## AgeCategory    -0.33    -1.01 0.03
## SleepTime       0.68     7.85 0.00
```

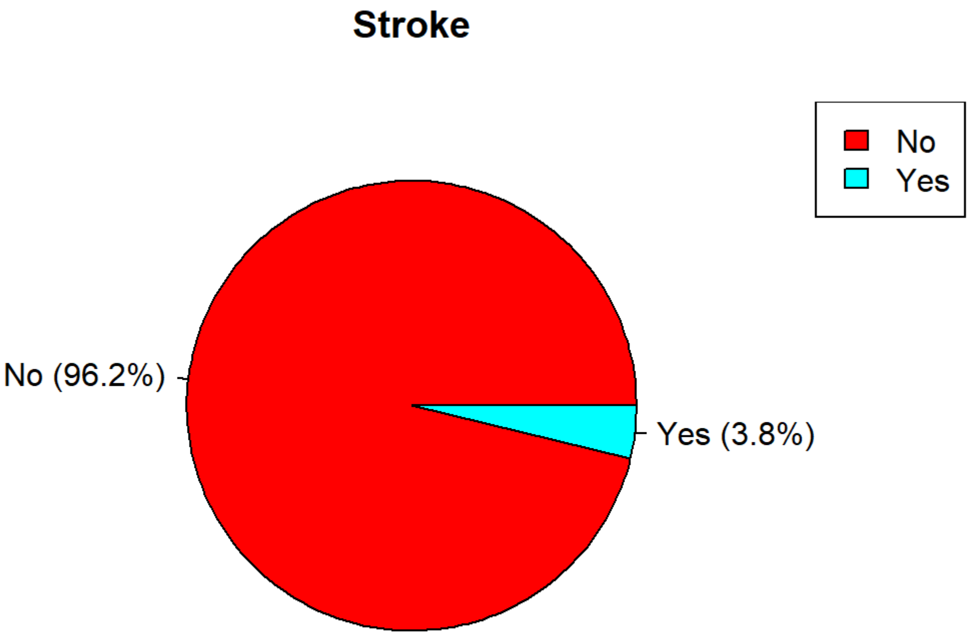
##Plots for categorical variable

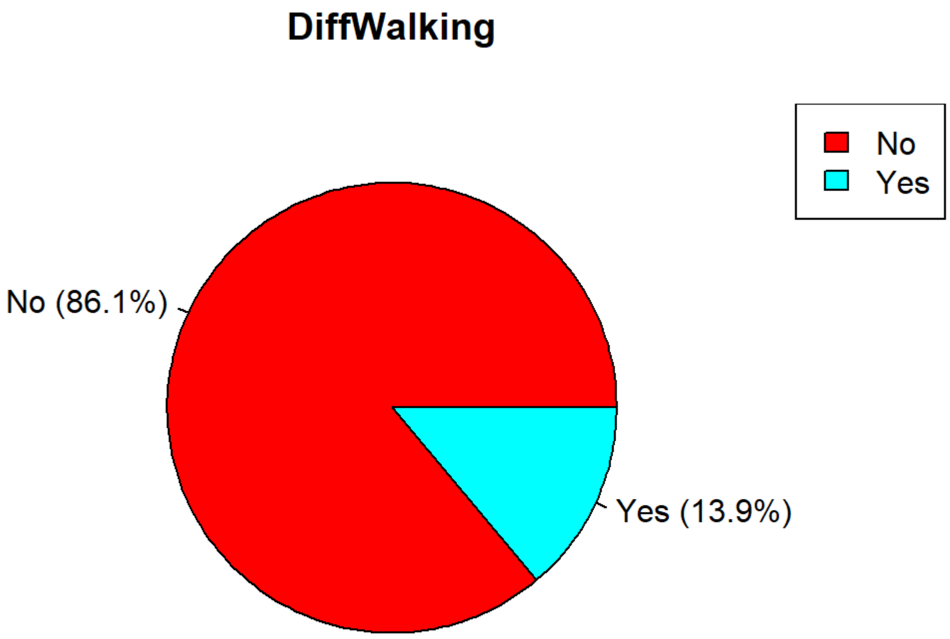
```
# Create a vector of variable names for the pie chart
vars <- c("HeartDisease", "Smoking", "AlcoholDrinking", "Stroke", "DiffWalking",
          "Sex", 'Race', 'Diabetic', 'PhysicalActivity', 'GenHealth', 'Asthma',
          'KidneyDisease', 'SkinCancer')

# Loop through each variable and create a pie chart
for(var in vars) {
  # Get the table of frequencies for the variable
  freq_table <- table(hd_data[[var]])

  # Create a color palette with one color for each category in the variable
  colors <- rainbow(length(freq_table))

  # Calculate percentage for each category
  pct <- round(100 * freq_table / sum(freq_table), 1)

  # Add percentage to labels
  labels <- paste(names(freq_table), " (", pct, "%)", sep="")

  # Create the pie chart
  pie(freq_table, col = colors, labels = labels, main = var)
  legend("topright", legend = names(freq_table), fill = colors)
}
```

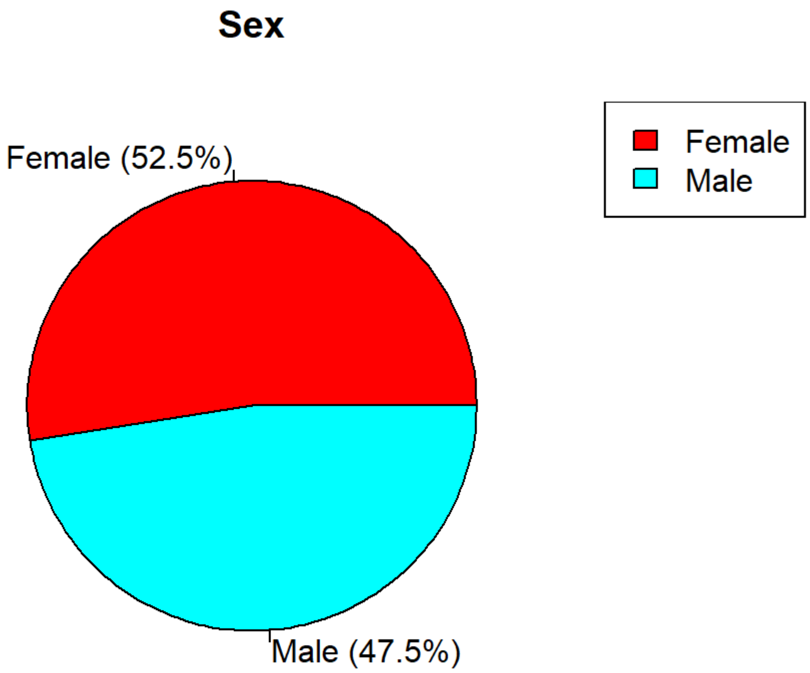

HeartDisease
🟥 No
🟦 Yes
No (91.4%)
Yes (8.6%)

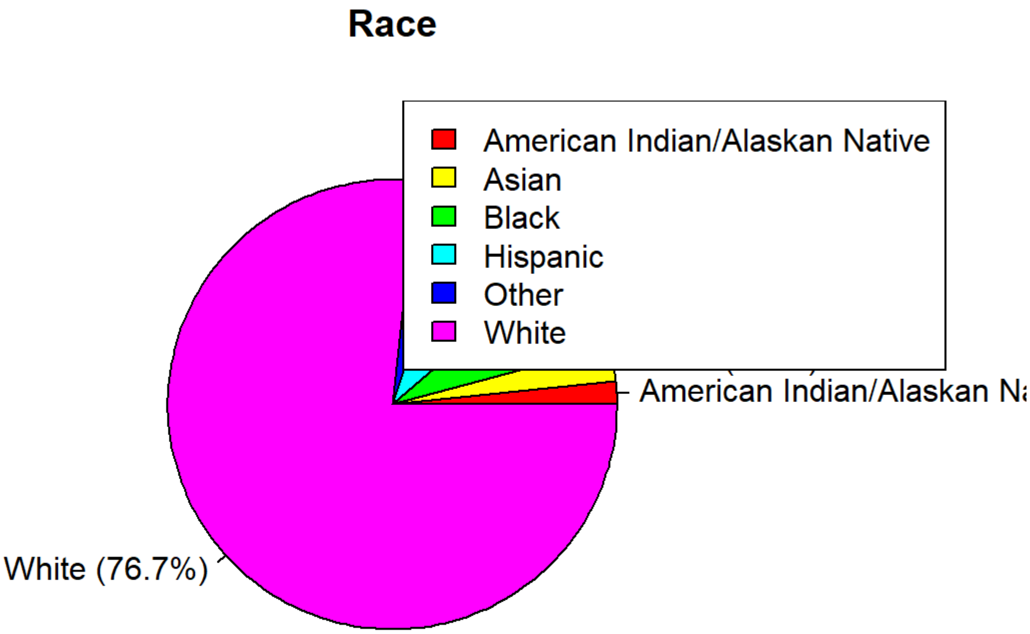

Smoking
🟥 No
🟦 Yes
No (58.8%)
Yes (41.2%)

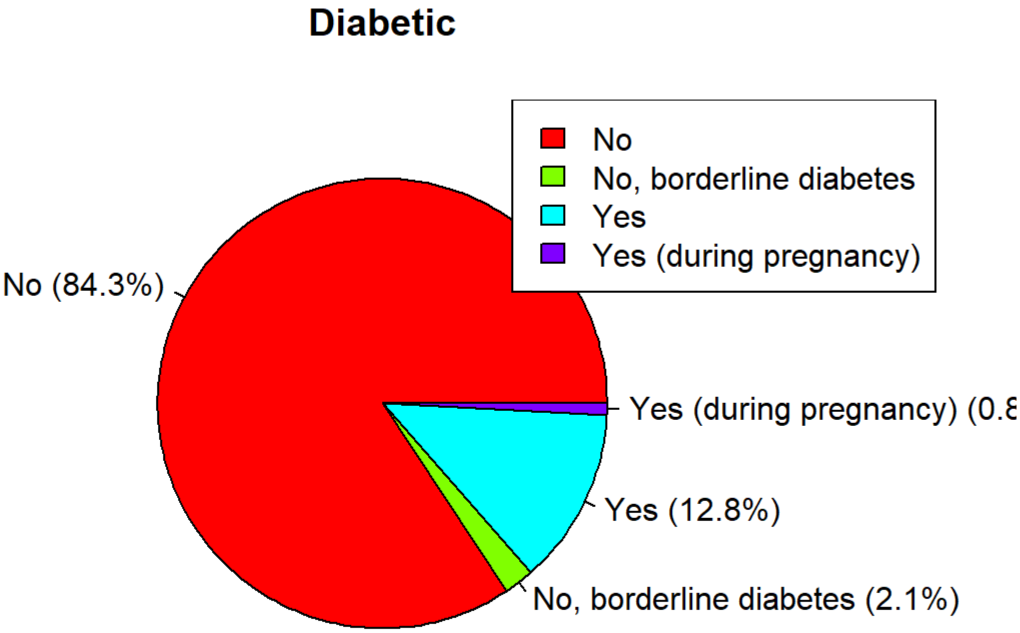

AlcoholDrinking
No
Yes
No (93.2%)
Yes (6.8%)

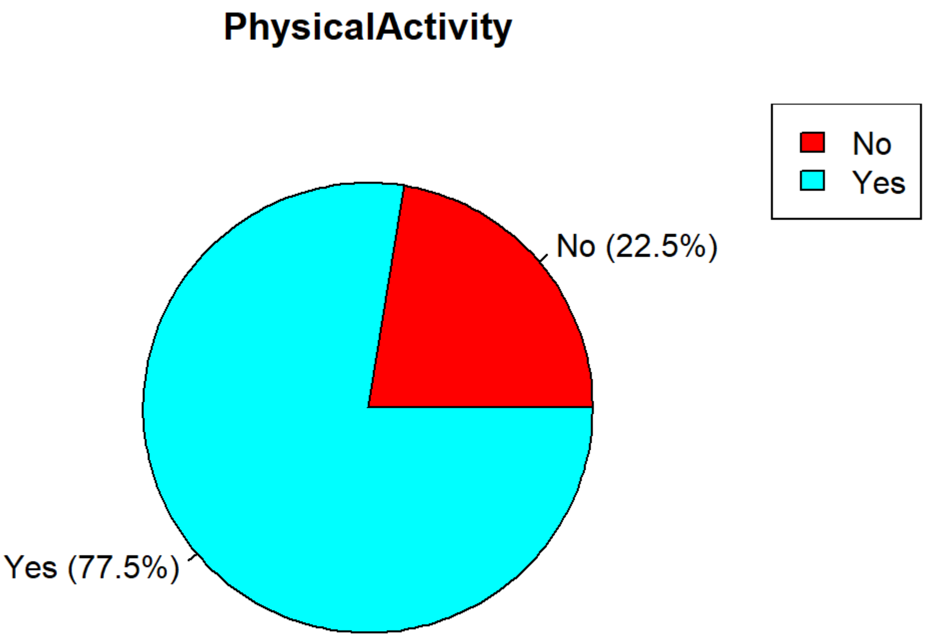

Stroke
No
Yes
No (96.2%)
Yes (3.8%)

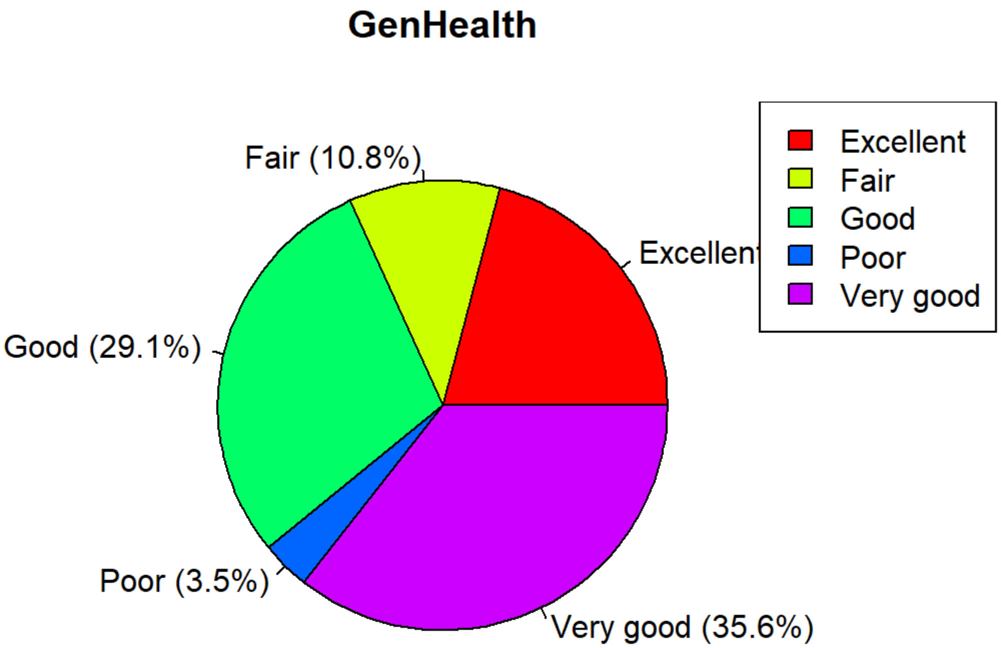

DiffWalking
No
Yes
No (86.1%)
Yes (13.9%)


Sex
Female
Male
Female (52.5%)
Male (47.5%)

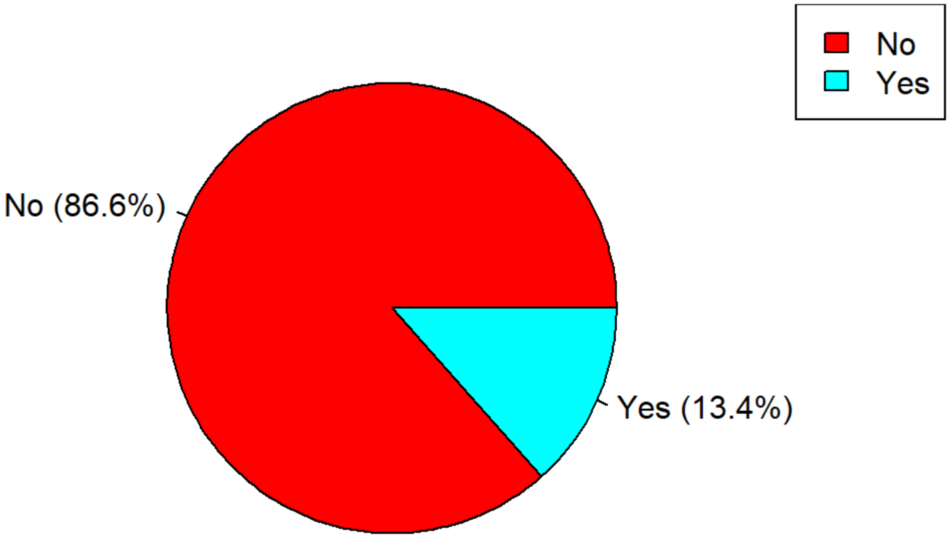

Race
American Indian/Alaskan Native
Asian
Black
Hispanic
Other
White
American Indian/Alaskan Na
White (76.7%)


Diabetic
No
No, borderline diabetes
Yes
Yes (during pregnancy)
No (84.3%)
Yes (during pregnancy) (0.8
Yes (12.8%)
No, borderline diabetes (2.1%)

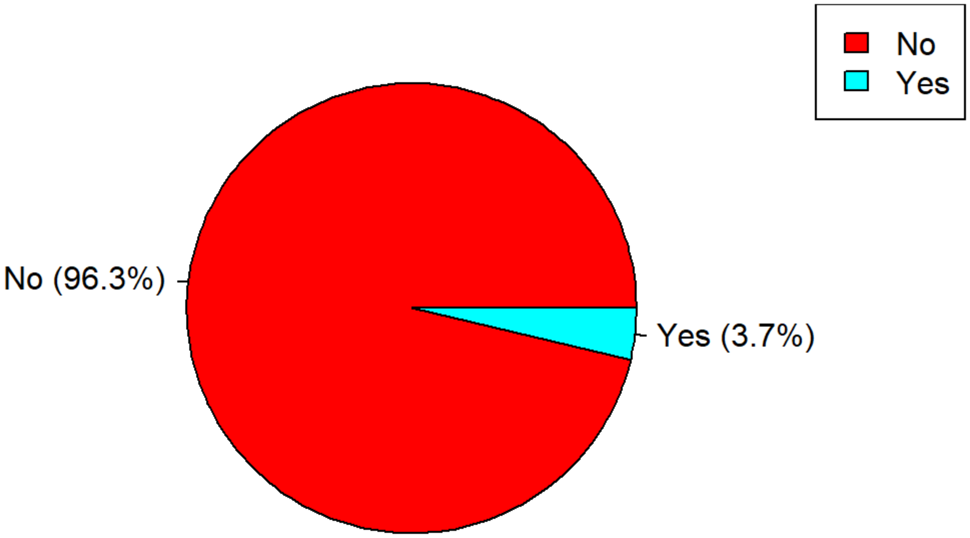

PhysicalActivity
🟥 No
🟦 Yes
No (22.5%)
Yes (77.5%)

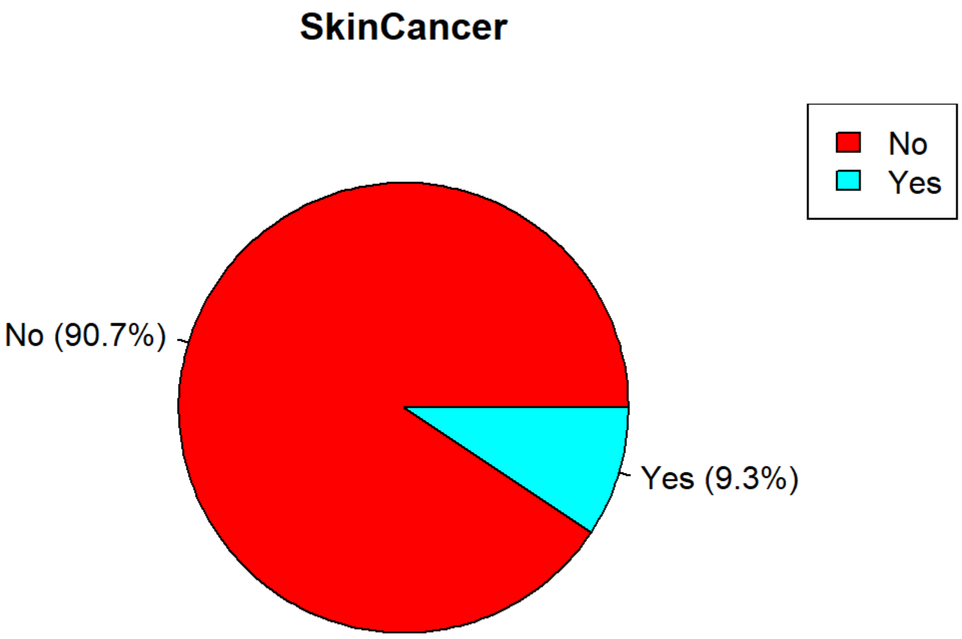

GenHealth
🟥 Excellent
🟨 Fair
🟩 Good
🟦 Poor
🟪 Very good
Fair (10.8%)
Excellen
Good (29.1%)
Poor (3.5%)
Very good (35.6%)

Asthma



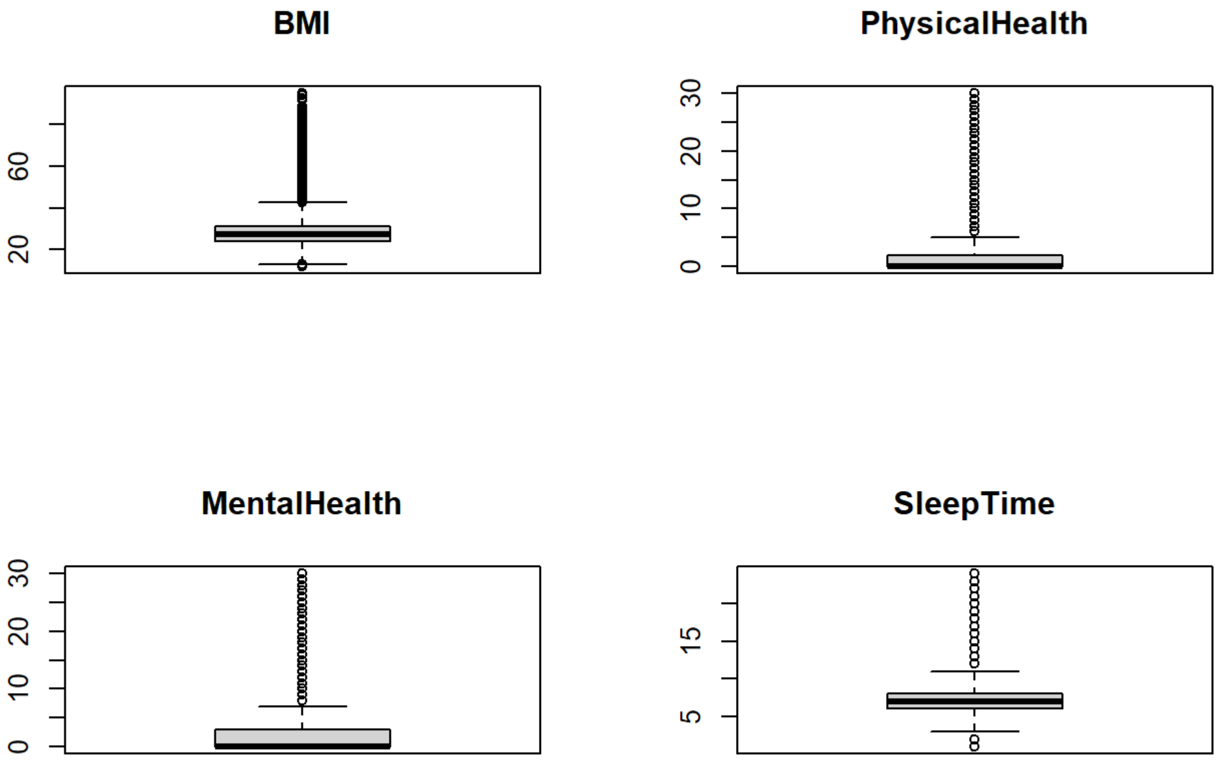
KidneyDisease





##checking for outliers for continuous data

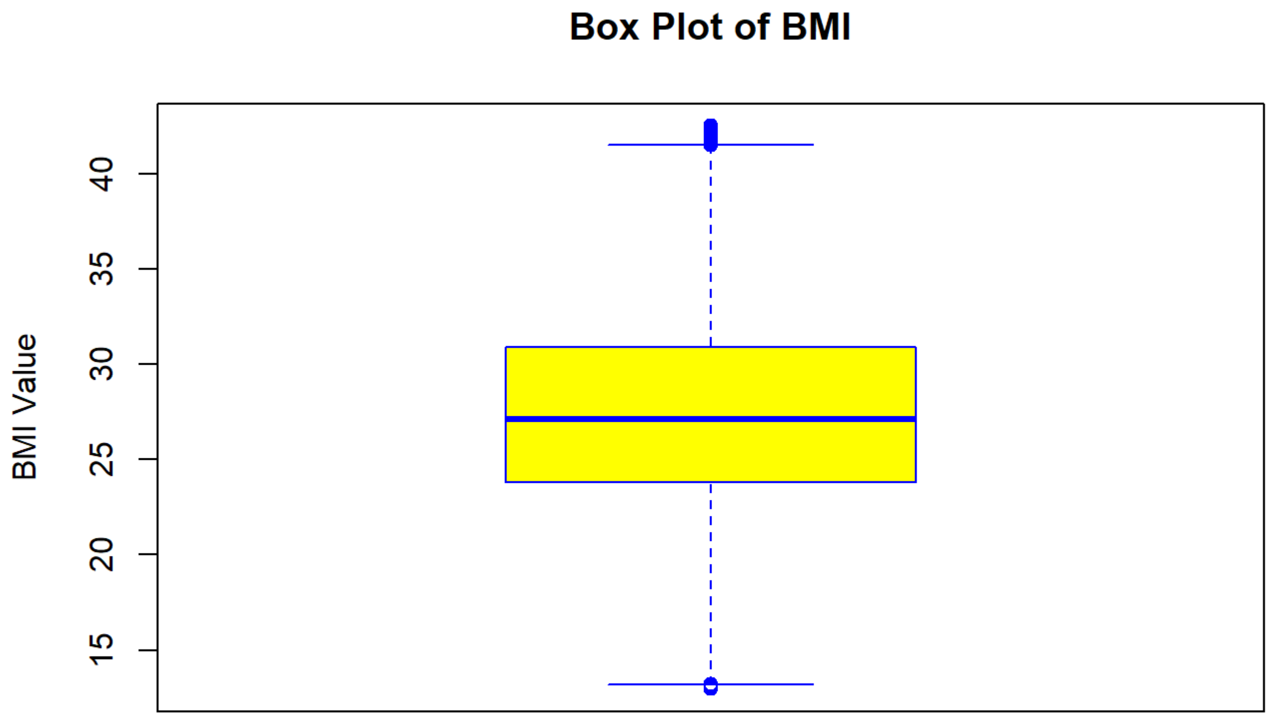
```
# Create a vector of variable names for the boxplots
vars <- c("BMI", "PhysicalHealth", "MentalHealth", "SleepTime")

# Loop through each variable and create a boxplot
par(mfrow=c(2,2)) # arrange the plots in a 2x2 grid
for(var in vars) {
  # Create the boxplot
  boxplot(hd_data[[var]], main = var)
}
```



removing outliers for BMI data and creating a box plot again

```
# Find the lower and upper bounds of the interquartile range (IQR)
Q1 <- quantile(hd_data$BMI, 0.25)
Q3 <- quantile(hd_data$BMI, 0.75)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

# Remove outliers from the dataset
hd_data <- hd_data[hd_data$BMI >= lower_bound & hd_data$BMI <= upper_bound,]
```

```
boxplot(hd_data$BMI,main = "Box Plot of BMI",ylab = "BMI Value",col = "yellow",border = "blue",h
orizontal = FALSE)
```



##Converting data into numerical variables Here we have to convert our data set into numerical data.

```
##REDO THE CODE THIS IS NOT WORKING
# Convert "yes" and "no" data to numeric using if statement
hd_data$HeartDisease <- ifelse(hd_data$HeartDisease == "Yes", 1, 0)
hd_data$Smoking <- ifelse(hd_data$Smoking == "Yes", 1, 0)
hd_data$AlcoholDrinking <- ifelse(hd_data$AlcoholDrinking == "Yes", 1, 0)
hd_data$Stroke <- ifelse(hd_data$Stroke == "Yes", 1, 0)
hd_data$DiffWalking <- ifelse(hd_data$DiffWalking == "Yes", 1, 0)
hd_data$PhysicalActivity <- ifelse(hd_data$PhysicalActivity == "Yes", 1, 0)
hd_data$Asthma <- ifelse(hd_data$Asthma == "Yes", 1, 0)
hd_data$KidneyDisease <- ifelse(hd_data$KidneyDisease == "Yes", 1, 0)
hd_data$SkinCancer <- ifelse(hd_data$SkinCancer == "Yes", 1, 0)
```

##Convert GenHealth into numeric

```
# Convert GenHealth to numeric
hd_data$GenHealth <- ifelse(hd_data$GenHealth == "Excellent", 5,
                                             ifelse(hd_data$GenHealth == "Very good", 4,
                                                    ifelse(hd_data$GenHealth == "Good", 3,
                                                           ifelse(hd_data$GenHealth == "Fair",
2, 1))))
```

##Convert Sex into numeric

```
hd_data$Sex <- ifelse(hd_data$Sex == "Male", 2,
                                             ifelse(hd_data$Sex == "Female", 1,0))
```

##Convert Diabetic into numeric

```
hd_data$Diabetic <- ifelse(hd_data$Diabetic == "Yes", 4,
                                             ifelse(hd_data$Diabetic == "No", 3,
                                                    ifelse(hd_data$Diabetic == "No, borderline d
iabetes", 2,
                                                           ifelse(hd_data$Diabetic == "Yes (duri
ng pregnancy)", 1, 0))))
```

##Convert Race into numeric

```
hd_data$Race <- ifelse(hd_data$Race == "White", 6,
                                             ifelse(hd_data$Race == "Black", 5,
                                                    ifelse(hd_data$Race == "Asian", 4,
                                                           ifelse(hd_data$Race == "Hispanic", 3,
                                                           ifelse(hd_data$Race == "American Indi
an/Alaskan Native", 2, 1)))))
```

##Check the data After converting it into numeric

```
head(hd_data)
```

```
##   HeartDisease   BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth
## 1            0 16.60       1               0      0              3           30
## 2            0 20.34       0               0      1              0            0
## 3            0 26.58       1               0      0             20           30
## 4            0 24.21       0               0      0              0            0
## 5            0 23.71       0               0      0             28            0
## 6            1 28.87       1               0      0              6            0
##   DiffWalking Sex AgeCategory Race Diabetic PhysicalActivity GenHealth
## 1           0   1          57    6        4                1         4
## 2           0   1          80    6        3                1         4
## 3           0   2          67    6        4                1         2
## 4           0   1          77    6        3                0         3
## 5           1   1          42    6        3                1         4
## 6           1   1          77    5        3                0         2
##   SleepTime Asthma KidneyDisease SkinCancer
## 1         5      1             0          1
## 2         7      0             0          0
## 3         8      1             0          0
## 4         6      0             0          1
## 5         8      0             0          0
## 6        12      0             0          0
```

##Save the updated data set for future use

```
#write.csv(hd_data, file = "C:/Users/megha/OneDrive/Desktop/Machine Learning Project/heart_disea
se_data_v2.csv", row.names = FALSE)
```

By the above command we can save the new data set and use it for future.

##Find the summary

```
summary(hd_data)
```

```
##   HeartDisease          BMI           Smoking      AlcoholDrinking
##  Min.   :0.00000   Min.   :12.97   Min.   :0.000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:23.81   1st Qu.:0.000   1st Qu.:0.00000
##  Median :0.00000   Median :27.12   Median :0.000   Median :0.00000
##  Mean   :0.08465   Mean   :27.67   Mean   :0.412   Mean   :0.06895
##  3rd Qu.:0.00000   3rd Qu.:30.90   3rd Qu.:1.000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :42.50   Max.   :1.000   Max.   :1.00000
##      Stroke        PhysicalHealth    MentalHealth   DiffWalking
##  Min.   :0.00000   Min.   : 0.000   Min.   : 0.0   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.: 0.000   1st Qu.: 0.0   1st Qu.:0.0000
##  Median :0.00000   Median : 0.000   Median : 0.0   Median :0.0000
##  Mean   :0.03746   Mean   : 3.239   Mean   : 3.8   Mean   :0.1306
##  3rd Qu.:0.00000   3rd Qu.: 2.000   3rd Qu.: 3.0   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :30.000   Max.   :30.0   Max.   :1.0000
##       Sex        AgeCategory         Race          Diabetic
##  Min.   :1.00   Min.   :21.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.00   1st Qu.:42.00   1st Qu.:6.000   1st Qu.:3.000
##  Median :1.00   Median :57.00   Median :6.000   Median :3.000
##  Mean   :1.48   Mean   :54.48   Mean   :5.389   Mean   :3.085
##  3rd Qu.:2.00   3rd Qu.:67.00   3rd Qu.:6.000   3rd Qu.:3.000
##  Max.   :2.00   Max.   :80.00   Max.   :6.000   Max.   :4.000
##  PhysicalActivity   GenHealth      SleepTime          Asthma
##  Min.   :0.0000   Min.   :1.00   Min.   : 1.000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:3.00   1st Qu.: 6.000   1st Qu.:0.0000
##  Median :1.0000   Median :4.00   Median : 7.000   Median :0.0000
##  Mean   :0.7826   Mean   :3.62   Mean   : 7.105   Mean   :0.1299
##  3rd Qu.:1.0000   3rd Qu.:4.00   3rd Qu.: 8.000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :5.00   Max.   :24.000   Max.   :1.0000
##  KidneyDisease       SkinCancer
##  Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000
##  Mean   :0.03588   Mean   :0.09459
##  3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000
```

V5

Kishan

2023-04-16

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

##Load the Packages

```
library(ggplot2)
```

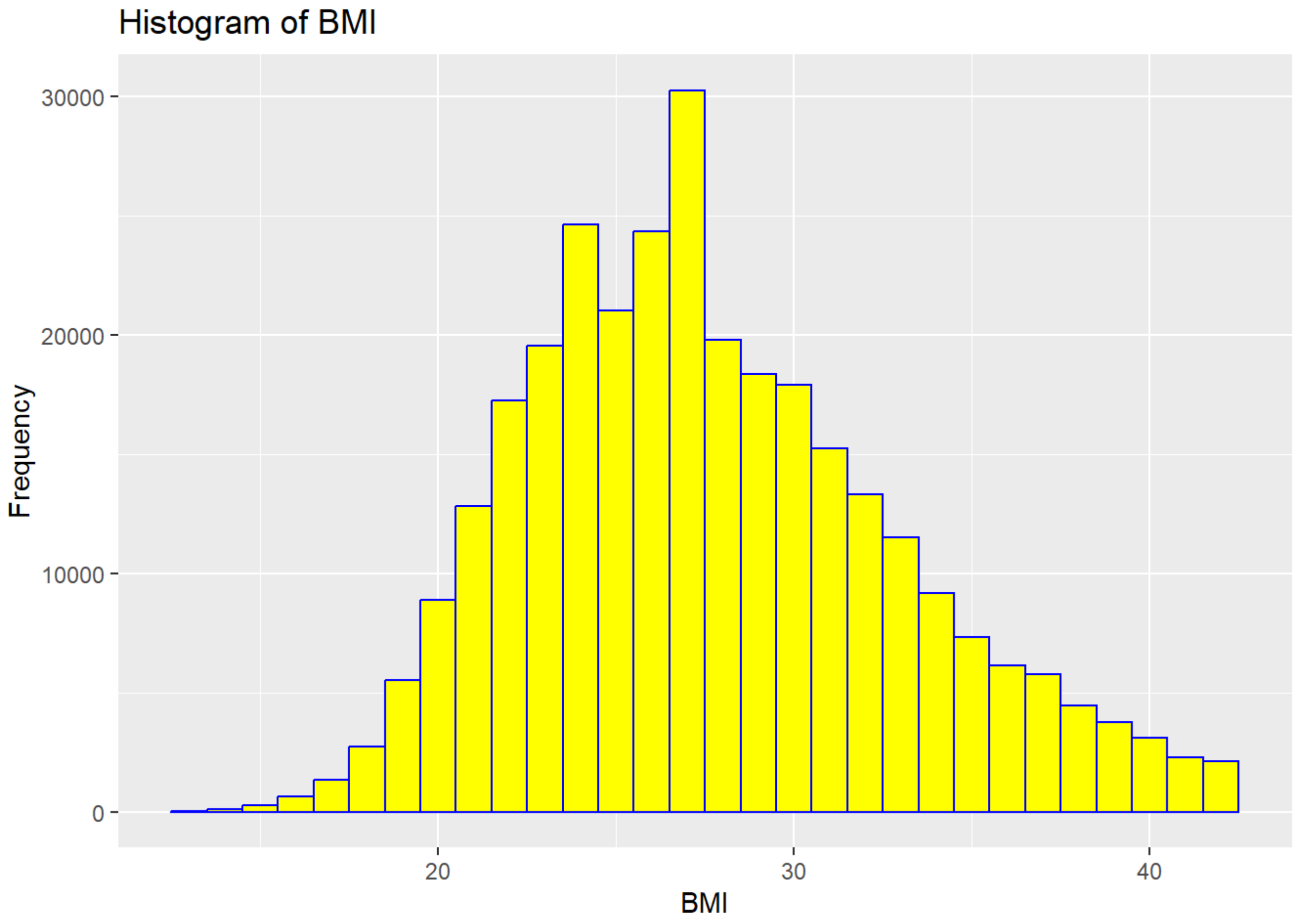
##Load the dataset

```
heart_disease <- read.csv("C:/Users/megha/OneDrive/Desktop/Machine Learning Project/heart_disease_data_v2.csv")
head(heart_disease)
```

```
##   HeartDisease   BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth
## 1            0 16.60       1               0      0              3           30
## 2            0 20.34       0               0      1              0            0
## 3            0 26.58       1               0      0             20           30
## 4            0 24.21       0               0      0              0            0
## 5            0 23.71       0               0      0             28            0
## 6            1 28.87       1               0      0              6            0
##   DiffWalking Sex AgeCategory Race Diabetic PhysicalActivity GenHealth
## 1           0   1          57    6        4                1         4
## 2           0   1          80    6        3                1         4
## 3           0   2          67    6        4                1         2
## 4           0   1          77    6        3                0         3
## 5           1   1          42    6        3                1         4
## 6           1   1          77    5        3                0         2
##   SleepTime Asthma KidneyDisease SkinCancer
## 1         5      1             0          1
## 2         7      0             0          0
## 3         8      1             0          0
## 4         6      0             0          1
## 5         8      0             0          0
## 6        12      0             0          0
```

##Plot the graph

```
##Plot a histogram to check the continuous variable BMI


ggplot(heart_disease, aes(x = BMI)) +
  geom_histogram(binwidth = 1, color = "blue", fill = "yellow") +
  labs(x = "BMI", y = "Frequency", title = "Histogram of BMI")
```



V3

Kishan

2023-04-15

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

##Loading Packages

```
library(mlbench)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-6
```

```
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(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

Load the necessary packages required for regression analysis.

##Load the dataset

```
heart_disease <- read.csv("C:/Users/megha/OneDrive/Desktop/Machine Learning Project/heart_diseas
e_data_v2.csv")
head(heart_disease)
```

```
##   HeartDisease   BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth
## 1            0 16.60       1               0      0              3           30
## 2            0 20.34       0               0      1              0            0
## 3            0 26.58       1               0      0             20           30
## 4            0 24.21       0               0      0              0            0
## 5            0 23.71       0               0      0             28            0
## 6            1 28.87       1               0      0              6            0
##   DiffWalking Sex AgeCategory Race Diabetic PhysicalActivity GenHealth
## 1           0   1          57    6        4                1         4
## 2           0   1          80    6        3                1         4
## 3           0   2          67    6        4                1         2
## 4           0   1          77    6        3                0         3
## 5           1   1          42    6        3                1         4
## 6           1   1          77    5        3                0         2
##   SleepTime Asthma KidneyDisease SkinCancer
## 1         5      1             0          1
## 2         7      0             0          0
## 3         8      1             0          0
## 4         6      0             0          1
## 5         8      0             0          0
## 6        12      0             0          0
```

Here I have loaded the head of data set to check whether all the data is in numeric to proceed further.

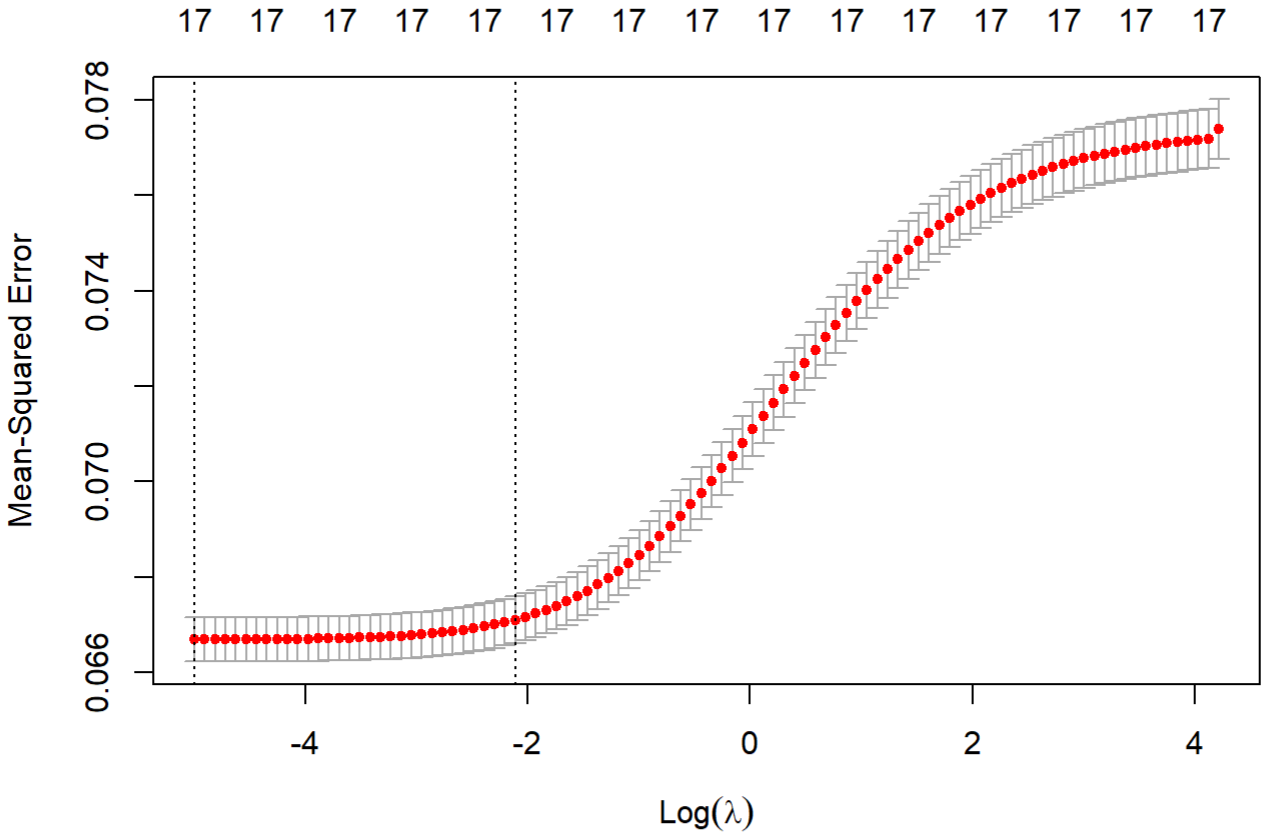
##Spliting the data into training and test set

```
# Split the data into training and test set
set.seed(123)
training.samples <- heart_disease$HeartDisease %>%
createDataPartition(p = 0.75, list = FALSE)
train.data <- heart_disease[training.samples, ]
test.data <- heart_disease[-training.samples, ]

# Create the matrix of predictors for glmnet function
x <- as.matrix(train.data[2:18])

# Convert the outcome (class) to a numerical variable
y <- train.data$HeartDisease
```

Split the data into train and test to fit them into models.

##Ridge Regression

```
# Find the optimal lambda that minimizes the 10-fold cross-validation error:
ridge <- glmnet(x, y, alpha = 0, lambda = NULL)
cv.ridge <- cv.glmnet(x, y, alpha = 0)
plot(cv.ridge)
```



##Find the lambda values

```
cv.ridge$lambda.min
```

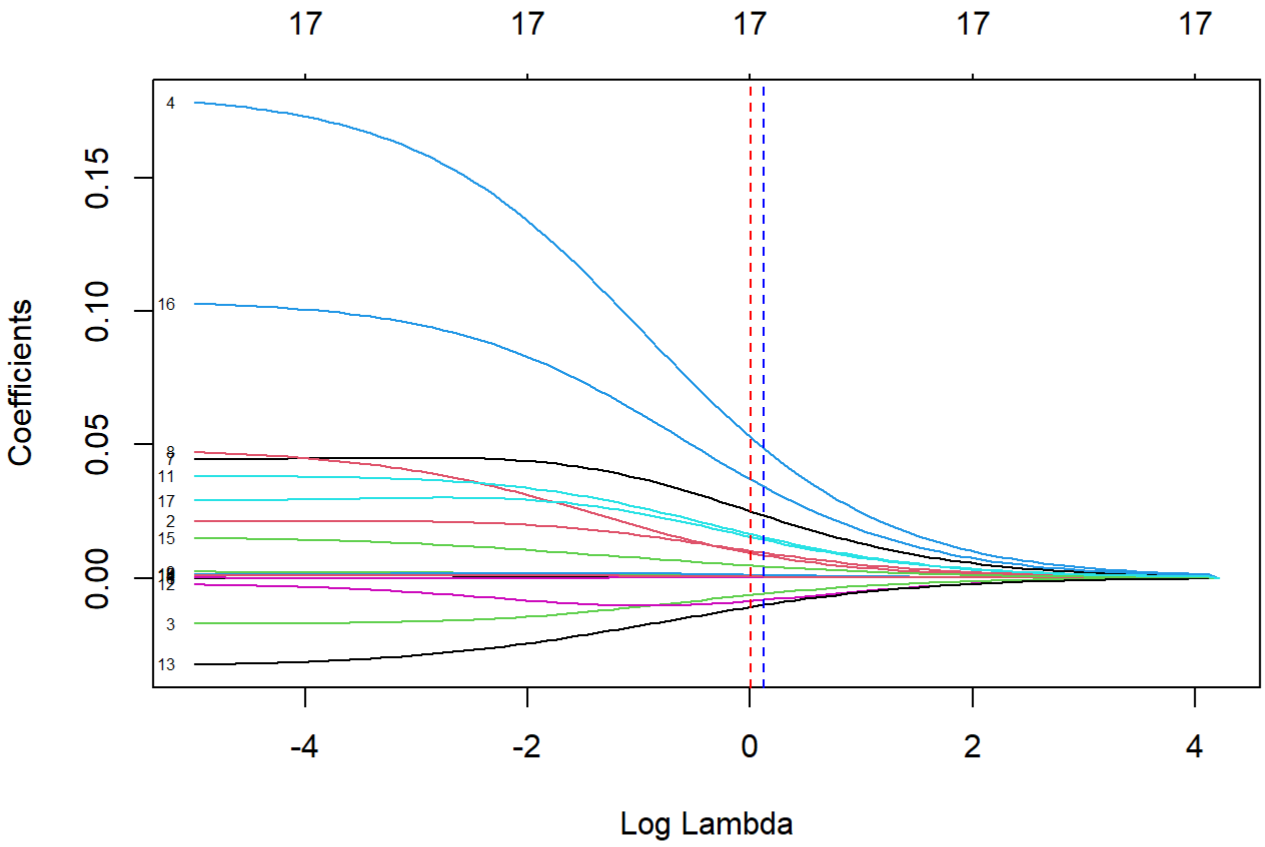
```
## [1] 0.006771639
```

```
cv.ridge$lambda.1se
```

```
## [1] 0.1211209
```

##Plot the graph for lambda coefficients

```
# Plot the coefficients
plot(ridge, xvar = "lambda", label=T)
abline(v=cv.ridge$lambda.min, col = "red", lty=2)
abline(v=cv.ridge$lambda.1se, col="blue", lty=2)
```



##Calculate RMSE

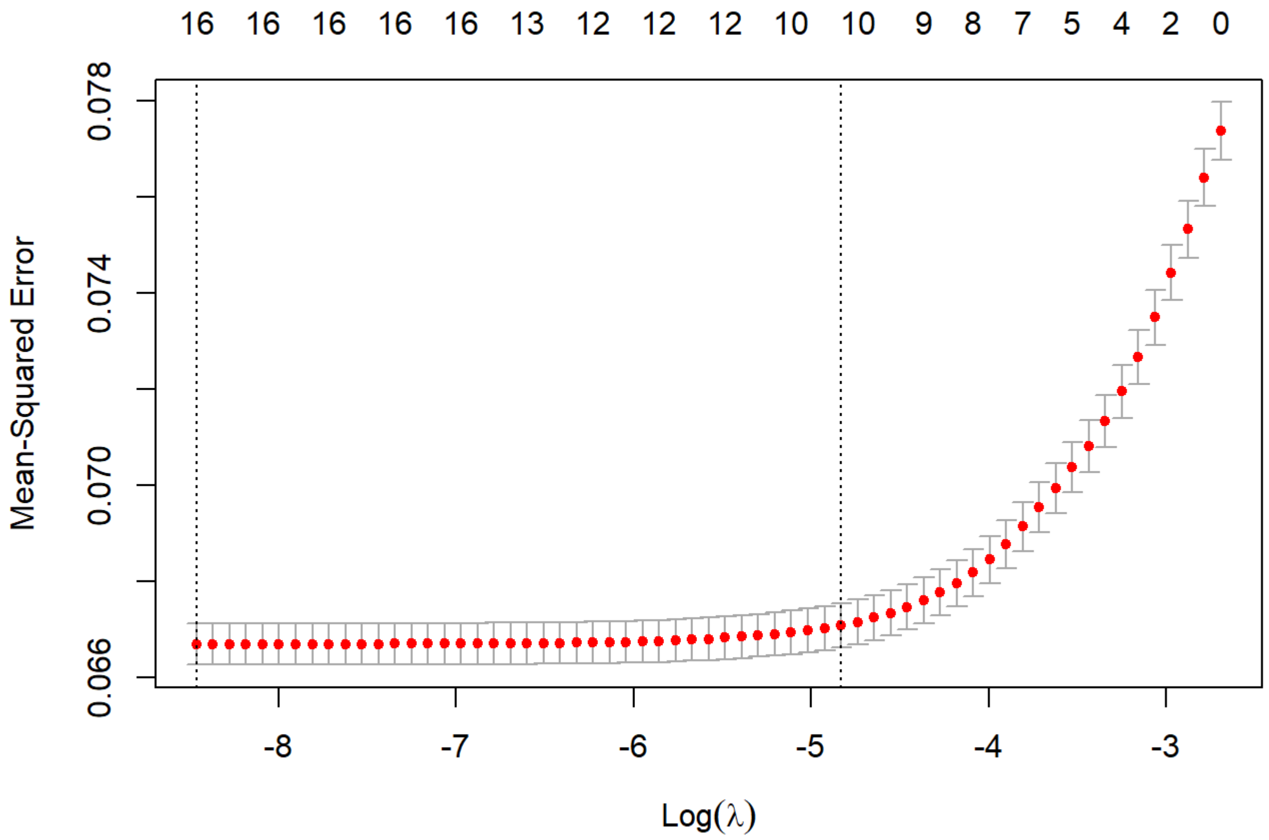
```
# Make predictions on the test data
x.test <- as.matrix(test.data[2:18])
predictions <- ridge %>% predict(x.test)

# Model performance metrics
data.frame(
RMSE.ridge = caret::RMSE(predictions, test.data$HeartDisease)
)
```

```
##   RMSE.ridge
## 1  0.2674484
```

```
RMSE.ridge <- caret::RMSE(predictions, test.data$HeartDisease)
```

##Lasso Regression

```
lasso <- glmnet(x, y, alpha = 1, lambda = NULL)
# Cross-validation to find the optimal lambda penalization
cv <- cv.glmnet(x, y, alpha = 1)
plot(cv) # Display the best lambda value
```



##Find the lambda coefficients

```
cv$lambda.min
```

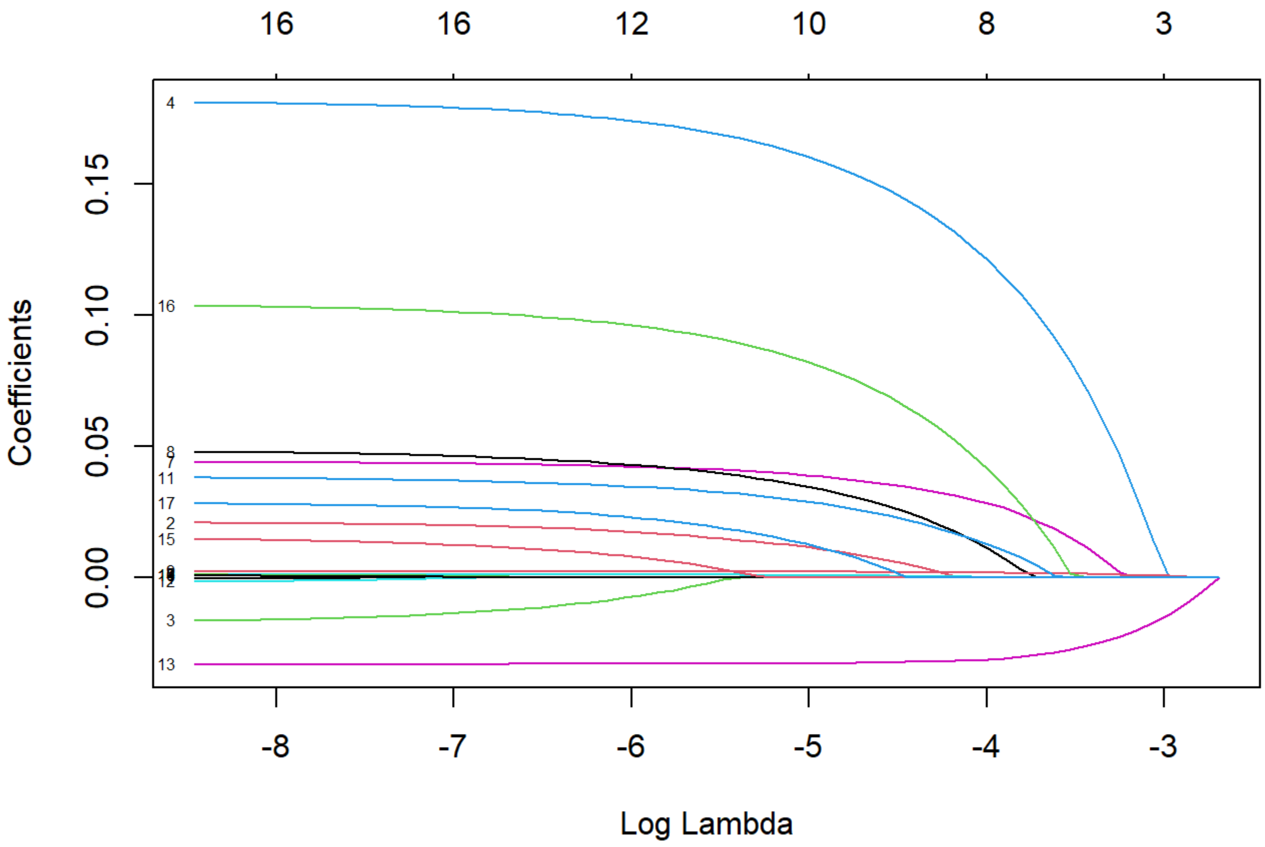
```
## [1] 0.0002116622
```

```
cv$lambda.1se
```

```
## [1] 0.007968946
```

##Plot the graph for lambda coefficients

```
# Plot the coefficients
plot(lasso, xvar = "lambda", label=T)
abline(v=cv$lambda.min, col = "red", lty=2)
abline(v=cv$lambda.1se, col="blue", lty=2)
```



##Calculate RMSE

```
# Make predictions on the test data
predictions <- lasso %>% predict(x.test)
# Model performance metrics
data.frame(
RMSE.lasso = caret::RMSE(predictions, test.data$HeartDisease)
)
```

```
##   RMSE.lasso
## 1  0.2617881
```

```
RMSE.lasso <- caret::RMSE(predictions, test.data$HeartDisease)
```

##Elastic net

```
elastic <- train(
HeartDisease ~., data = train.data, method = "glmnet",trControl = trainControl("cv", number = 1
0),tuneLength = 10)
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

##Calculate RMSE

```
# Make predictions
predictions <- elastic %>% predict(test.data)
# Model prediction performance
data.frame(
RMSE.elastic = caret::RMSE(predictions, test.data$HeartDisease)
)
```

```
##   RMSE.elastic
## 1    0.2591735
```

```
RMSE.elastic <- caret::RMSE(predictions, test.data$HeartDisease)
```

##Comparision

```
RMSE <- data.frame(model= c("Ridge Regression","Lasso Regression","Elastic Regression"), RMSE =
c(RMSE.ridge,RMSE.lasso,RMSE.elastic))
RMSE
```

```
##                model      RMSE
## 1   Ridge Regression 0.2674484
## 2   Lasso Regression 0.2617881
## 3 Elastic Regression 0.2591735
```

V4

Kishan

2023-04-15

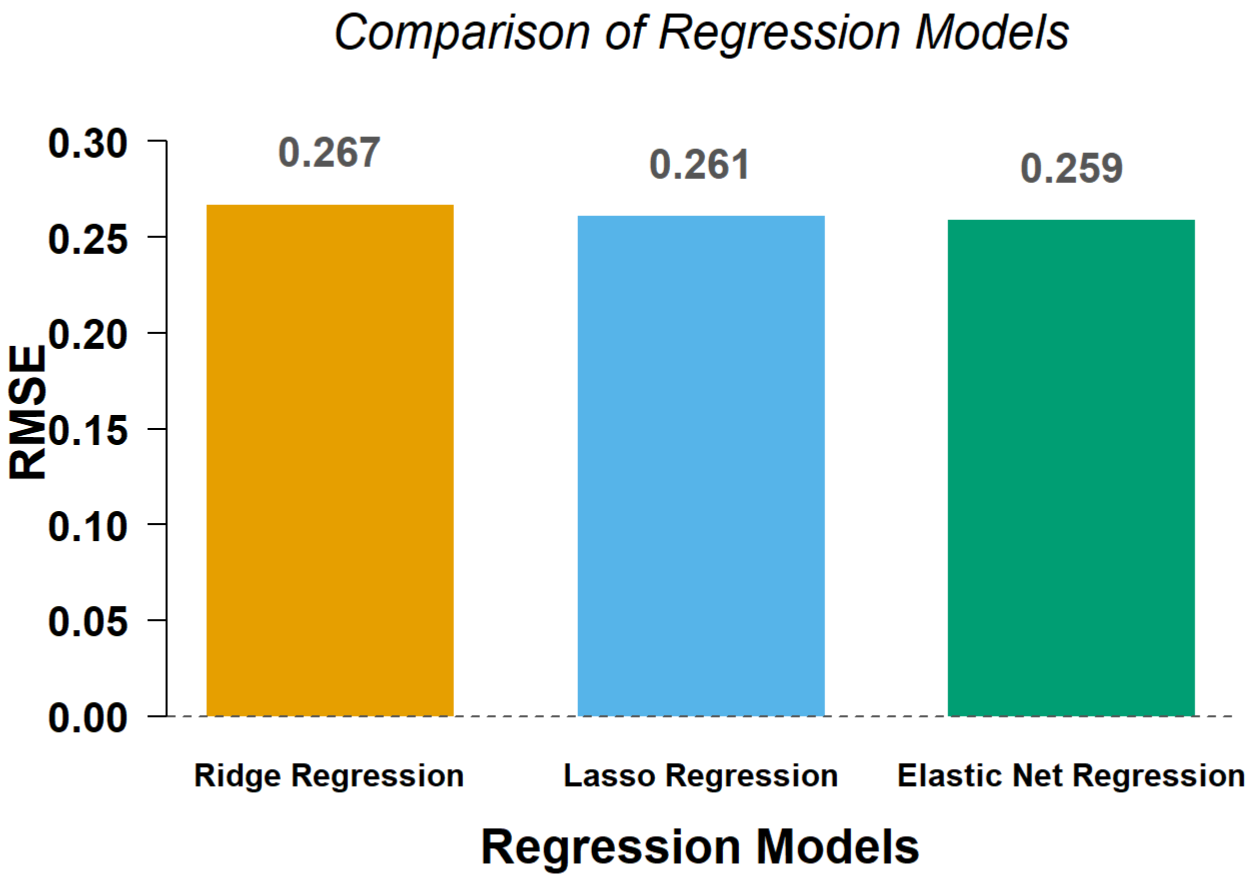
R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

##Box plot to compare the results

```
# Set the RMSE values
RMSE.ridge <- 0.267
RMSE.lasso <- 0.261
RMSE.elastic <- 0.259

# Create a color palette
colors <- c("#E69F00", "#56B4E9", "#009E73")

# Set up the plot area
par(mar = c(5, 5, 4, 2) + 0.1)

# Create the barplot with values
bp <- barplot(c(RMSE.ridge, RMSE.lasso, RMSE.elastic),
        names.arg = c("Ridge Regression", "Lasso Regression", "Elastic Net Regression"),
        col = colors,
        xlab = "Regression Models",
        ylab = "RMSE",
        main = "Comparison of Regression Models",
        ylim = c(0, max(c(RMSE.ridge, RMSE.lasso, RMSE.elastic)) + 0.05),
        border = NA,
        space = 0.5,
        font.lab = 2,
        font.axis = 2,
        font.main = 3,
        las = 1,
        cex.lab = 1.5,
        cex.axis = 1.3,
        cex.main = 1.5)

# Add the values to the plot
text(x = bp, y = c(RMSE.ridge, RMSE.lasso, RMSE.elastic) + 0.01,
     labels = c(RMSE.ridge, RMSE.lasso, RMSE.elastic),
     col = "#555555",
     font = 2,
     pos = 3,
     cex = 1.3)

# Add a horizontal line at y = 0
abline(h = 0, lty = 2, col = "#555555")
```


Comparison of Regression Models
RMSE
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.267
0.261
0.259
Ridge Regression
Lasso Regression
Elastic Net Regression
Regression Models