cdc_findings

2023-12-08

Data Set Information:

This data set encompasses detailed statistics from the CDC's National Center for Health Statistics regarding live births in the United States, spanning from 2016 to 2022. It features a comprehensive range of demographic and health information derived from birth certificates. Key attributes include maternal age, pre-pregnancy BMI, birth weight, gestational age, county of residence, and various other health-related factors. This rich dataset offers insights into socio-economic, racial, and maternal and infant health aspects across different counties. The dataset, part of the National Vital Statistics System, is a valuable resource for in-depth analysis and research in public health and is accessible via the CDC WONDER Online Database.

Objective:

To understand the dataset's basic characteristics, identify patterns, outliers, anomalies, and underlying structures.

Research Goal

1.What are the key socio-economic and demographic predictors of maternal and infant health outcomes, as indicated by machine learning models? 2. Which machine learning algorithm provides the most accurate predictions of high-risk pregnancies based on socio-economic and racial factors?

EDA - Exploratory Data Analysis

```
# Load necessary libraries
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.1

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.1

##
## 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
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

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

```r
library(tidyverse)
```

```
## Warning: package 'stringr' was built under R version 4.3.1

## Warning: package 'lubridate' was built under R version 4.3.1

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.
0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.0
## ✓ readr     2.1.4

## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## ✕ randomForest::combine() masks dplyr::combine()
## ✕ dplyr::filter()         masks stats::filter()
## ✕ dplyr::lag()            masks stats::lag()
## ✕ randomForest::margin()  masks ggplot2::margin()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```r
library(tidyr)
library(caret)
```

```
## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.3.1

##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

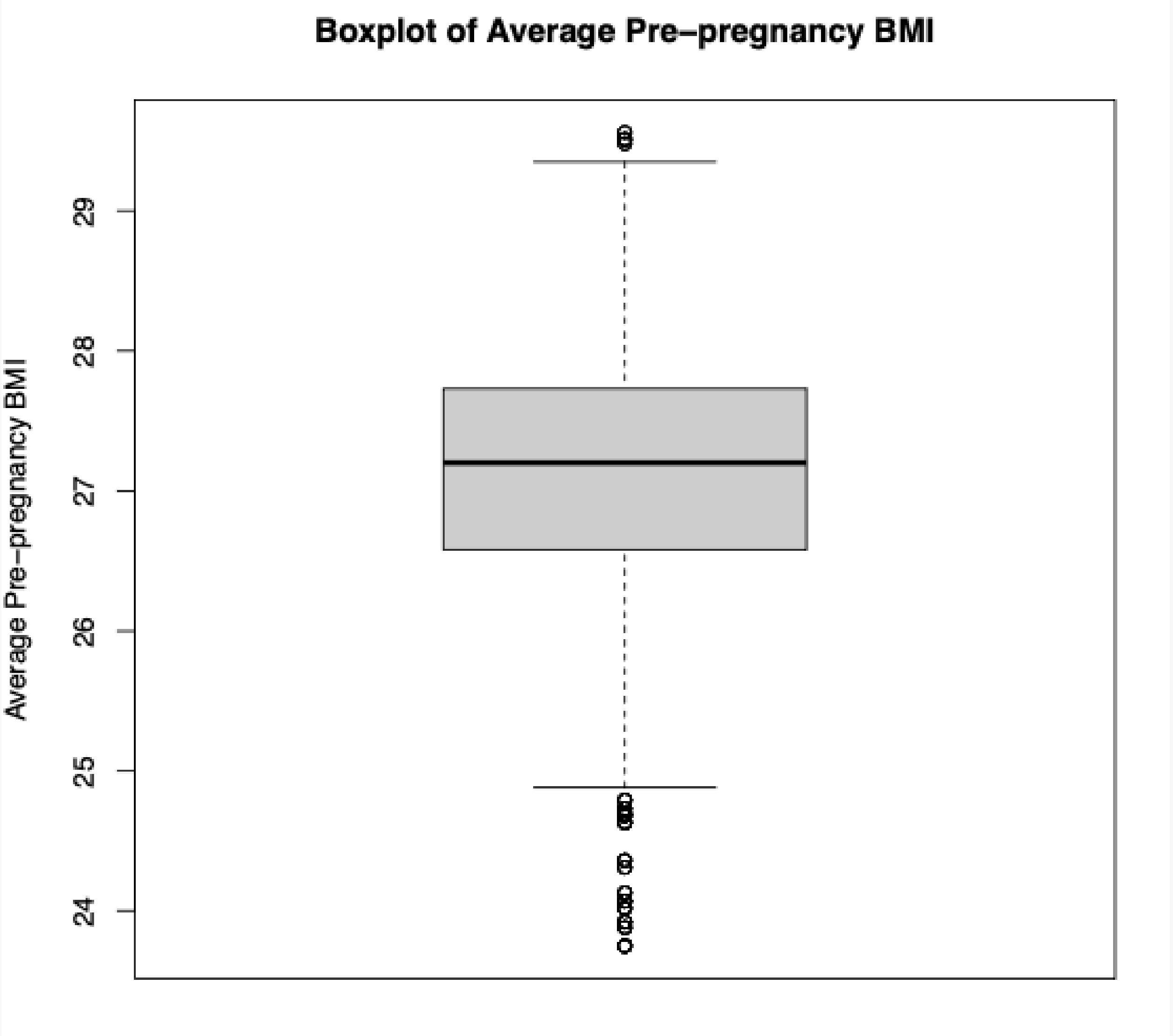
```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

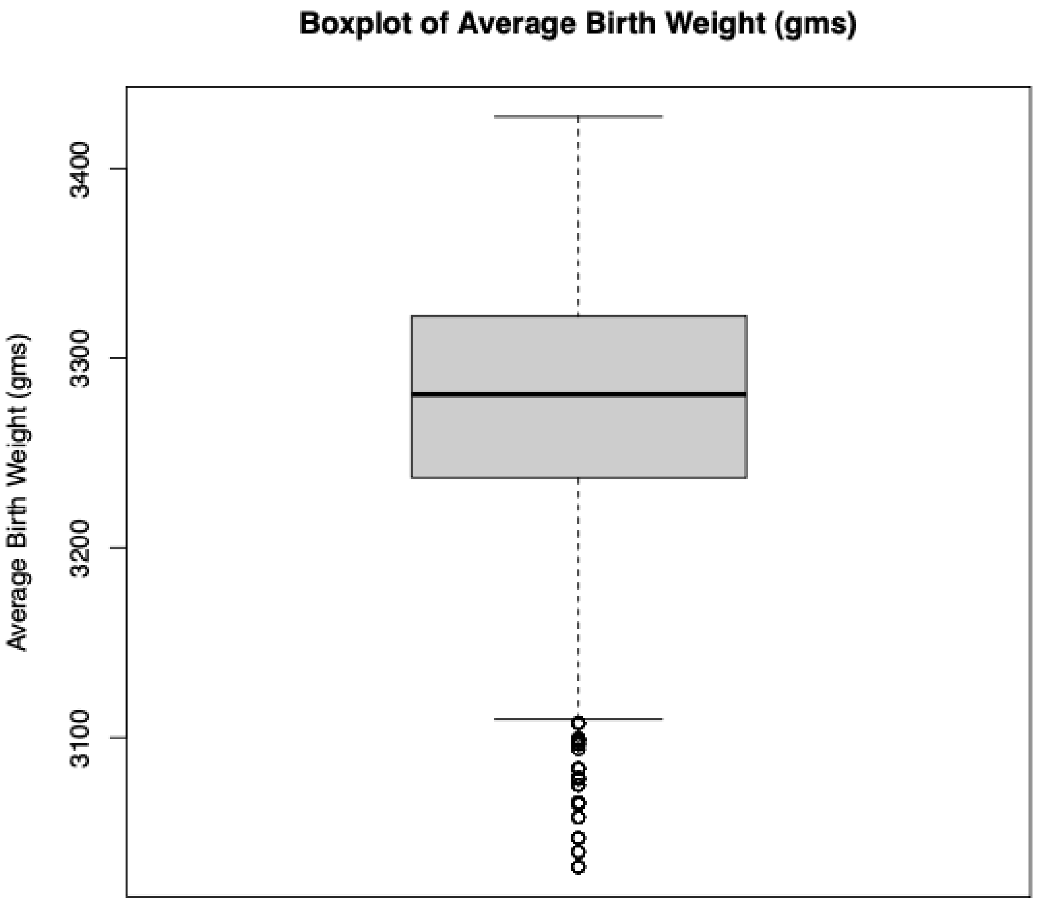
```r
# Read the data
data <- read.csv("cdc_county_data.csv")
head(data,5)
```

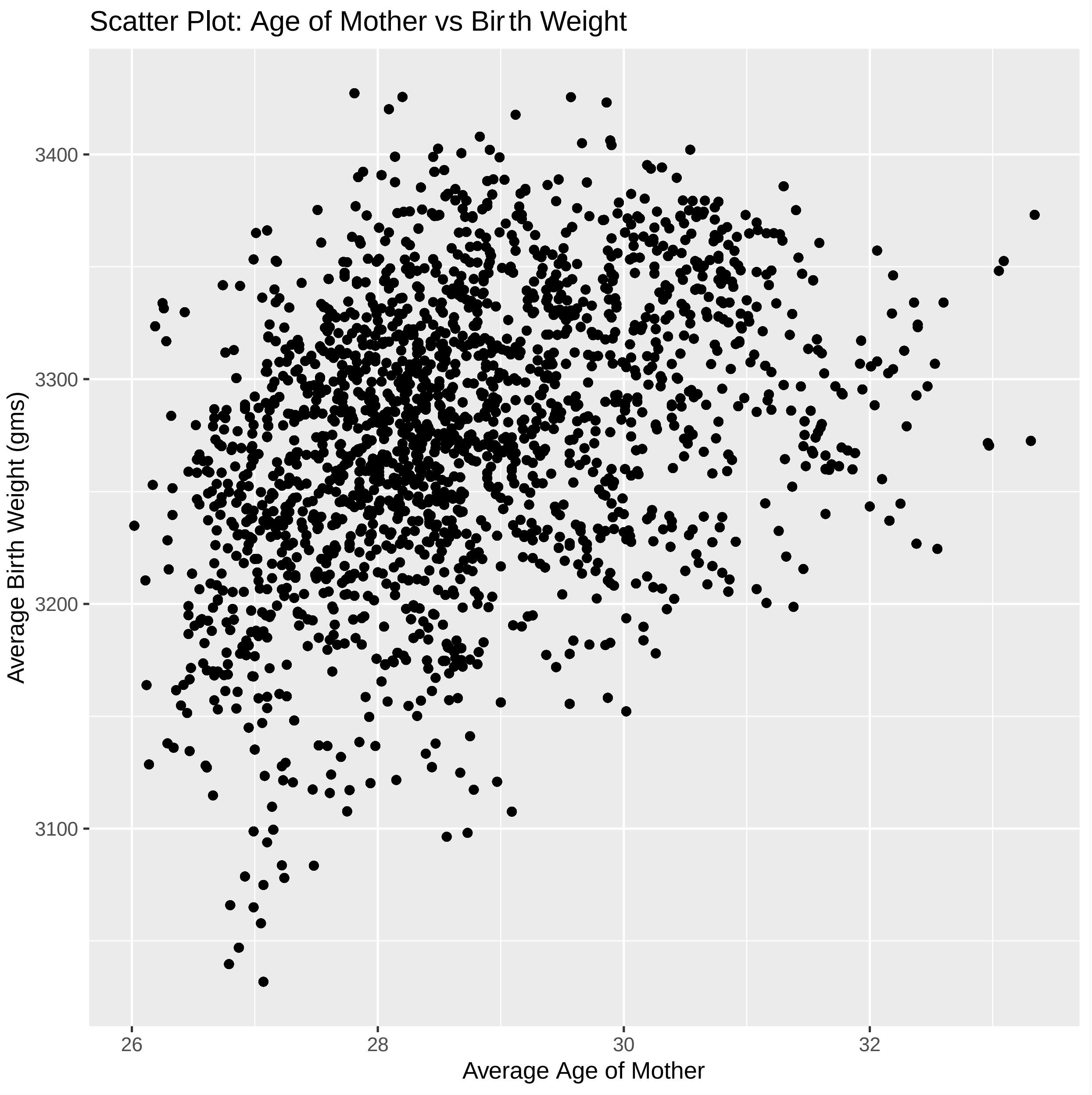
```
##         Year County_of_Residence County_of_Residence_FIPS Births
## 1 2018-01-01   Mobile County, AL                     1097   5549
## 2 2017-01-01   Mobile County, AL                     1097   5607
## 3 2016-01-01   Mobile County, AL                     1097   5504
## 4 2018-01-01   Morgan County, AL                     1103   1432
## 5 2017-01-01   Morgan County, AL                     1103   1435
##   Ave_Age_of_Mother Ave_OE_Gestational_Age_Wks Ave_LMP_Gestational_Age_Wks
## 1             27.25                      37.95                       38.02
## 2             27.10                      38.07                       38.27
## 3             26.95                      37.96                       38.22
## 4             26.91                      38.14                       38.18
## 5             26.70                      38.12                       38.15
##   Ave_Birth_Weight_gms Ave_Pre_pregnancy_BMI Ave_Number_of_Prenatal_Wks
## 1              3129.31                 28.05                      10.87
## 2              3158.62                 27.87                      10.84
## 3              3144.94                 27.58                      10.95
## 4              3205.34                 28.04                      10.65
## 5              3201.54                 27.76                      10.78
##   Maternal_Morbidity_Desc Maternal_Morbidity_YN Births_Morbidity
## 1            None checked                     1             5549
## 2            None checked                     1             5549
## 3            None checked                     1             5549
## 4            None checked                     1             1410
## 5            None checked                     1             1410
##   Ave_Age_of_Mother_Morbidity Ave_OE_Gestational_Age_Wks_Morbidity
## 1                       27.08                                38.07
## 2                       27.08                                38.07
## 3                       27.08                                38.07
## 4                       26.71                                38.11
## 5                       26.71                                38.11
##   Ave_LMP_Gestational_Age_Wks_Morbidity Ave_Birth_Weight_gms_Morbidity
## 1                                 38.27                        3156.37
## 2                                 38.27                        3156.37
## 3                                 38.27                        3156.37
## 4                                 38.15                        3196.95
## 5                                 38.15                        3196.95
##   Ave_Pre_pregnancy_BMI_Morbidity Ave_Number_of_Prenatal_Wks_Morbidity
## 1                           27.90                                10.83
## 2                           27.90                                10.83
## 3                           27.90                                10.83
## 4                           27.77                                10.78
## 5                           27.77                                10.78
```

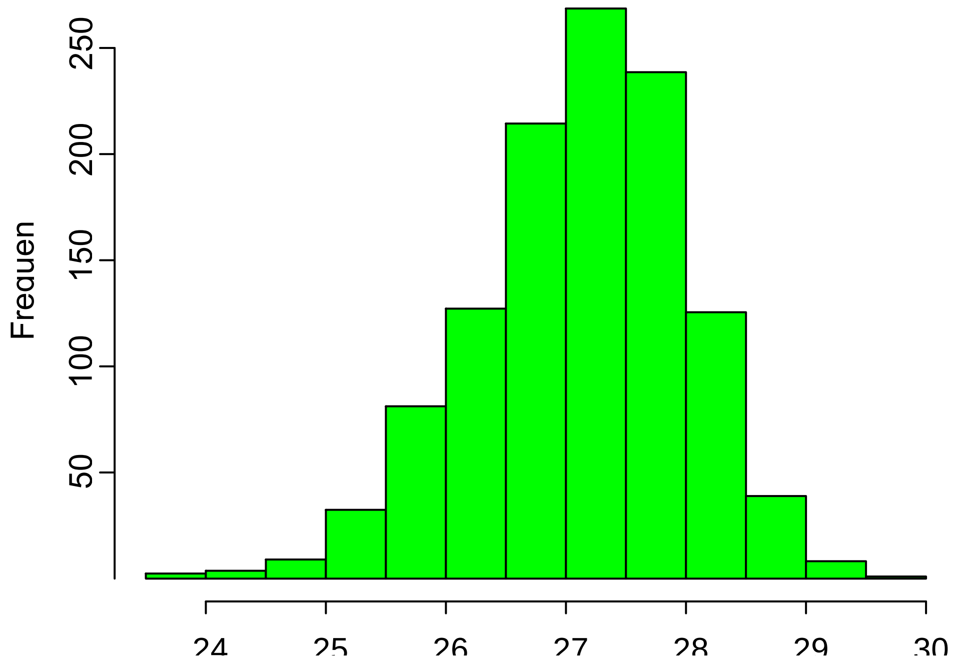
```
# EDA: Descriptive Statistics
descriptive_stats <- summary(data)

# EDA: Distribution Analysis
pdf("distribution_analysis.pdf")
hist(data$Ave_Age_of_Mother, main="Distribution of Average Age of Mother", xl
ab="Average Age of Mother", col="blue", border="black")
hist(data$Ave_Pre_pregnancy_BMI, main="Distribution of Average Pre-pregnancy
BMI", xlab="Average Pre-pregnancy BMI", col="green", border="black")
hist(data$Ave_Birth_Weight_gms, main="Distribution of Average Birth Weight (g
ms)", xlab="Average Birth Weight (gms)", col="red", border="black")
dev.off()

## quartz_off_screen
##                  2

# EDA: Outlier Detection
pdf("boxplot_analysis.pdf")
boxplot(data$Ave_Age_of_Mother, main="Boxplot of Average Age of Mother", ylab
="Average Age of Mother")
boxplot(data$Ave_Pre_pregnancy_BMI, main="Boxplot of Average Pre-pregnancy BM
I", ylab="Average Pre-pregnancy BMI")
boxplot(data$Ave_Birth_Weight_gms, main="Boxplot of Average Birth Weight (gms
)", ylab="Average Birth Weight (gms)")
dev.off()

## quartz_off_screen
##                  2

# EDA: Initial Correlation Assessment
key_variables <- data[, c("Ave_Age_of_Mother", "Ave_Pre_pregnancy_BMI", "Ave_
Birth_Weight_gms")]
correlation_matrix <- cor(key_variables, use="complete.obs")
corrp
```

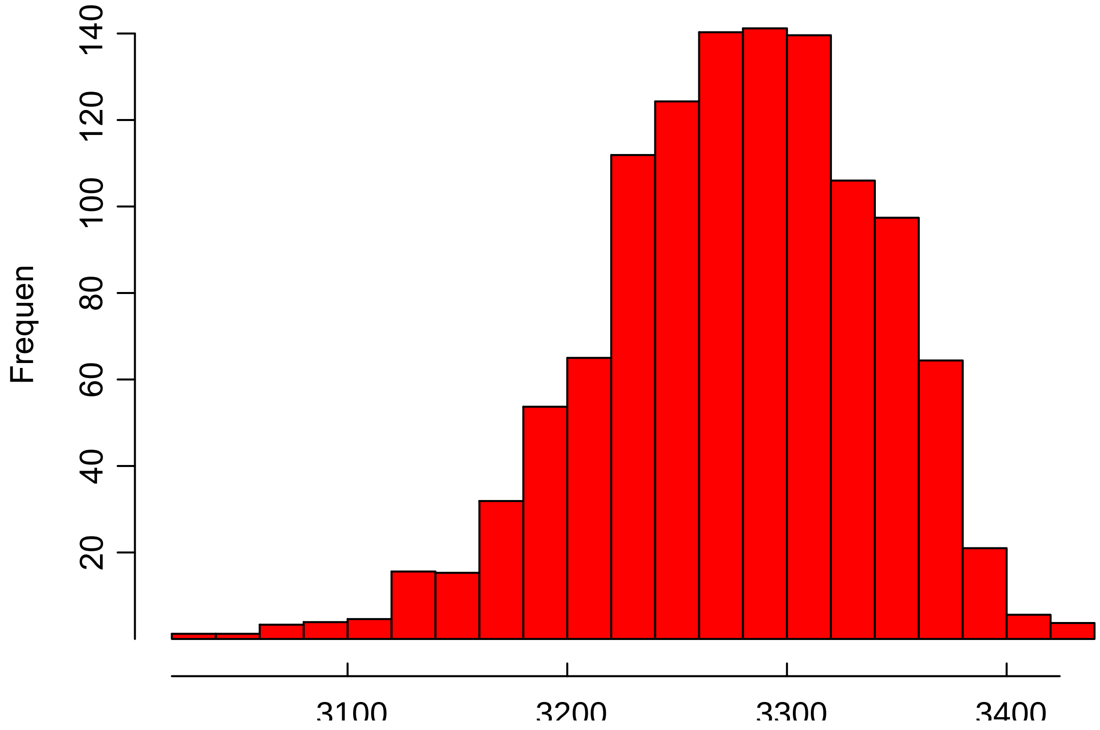


Boxplot of Average Pre−pregnancy BMI

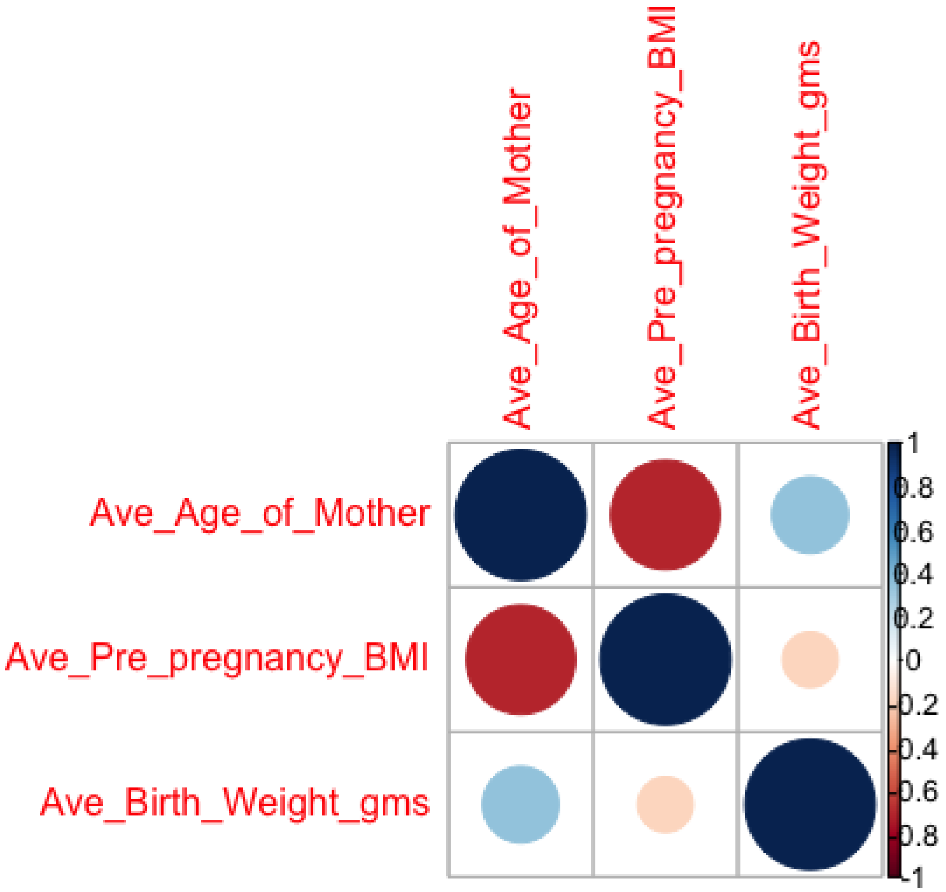

Boxplot of Average Birth Weight (gms)
Average Birth Weight (gms)
3400
3300
3200
3100


Scatter Plot: Age of Mother vs Birth Weight
Average Birth Weight (gms)
3400
3300
3200
3100
26
28
30
32
Average Age of Mother


Frequen
50
100
150
200
250
24
25
26
27
28
29
30


Frequen
20
40
60
80
100
120
140
3100
3200
3300
3400

```
lot(correlation_matrix, method="circle")
```



```
# Save the correlation plot
pdf("correlation_matrix.pdf")
corrplot(correlation_matrix, method="circle")
dev.off()

## quartz_off_screen
##                 2

# Initial Visualization: Scatter Plot - Example
pdf("scatter_plot.pdf")
ggplot(data, aes(x=Ave_Age_of_Mother, y=Ave_Birth_Weight_gms)) +
  geom_point() +
  labs(title="Scatter Plot: Age of Mother vs Birth Weight", x="Average Age of
Mother", y="Average Birth Weight (gms)")
dev.off()

## quartz_off_screen
##                 2
```

Descriptive Statistics

Ave_Age_of_Mother: Shows the average age of mothers in different counties and years.
Ave_Pre_pregnancy_BMI: Indicates the average Body Mass Index before pregnancy.
Ave_Birth_Weight_gms: Reflects the average birth weight of newborns in grams.

Distribution Analysis

Maternal Age: The distribution of the average age of mothers seems roughly normal but may be slightly skewed towards younger ages. Pre-pregnancy BMI: This variable also appears to follow a normal distribution, indicating a range of BMI values across the dataset. Birth Weight: The birth weight distribution is roughly normal, suggesting a typical spread of newborn weights.

Outlier Detection

Maternal Age: There are some outliers, indicating a few counties with unusually high or low average maternal ages. Pre-pregnancy BMI: There are outliers present, suggesting some extremes in BMI values. Birth Weight: Outliers are observed here as well, indicating some births with significantly high or low weights.

Research Question 1: Key Socio-Economic and Demographic Predictors

We will use a machine learning model to identify the most significant predictors of maternal and infant health outcomes.

```r
# Install and load necessary libraries
if (!require("caret")) install.packages("caret")
library(caret)
if (!require("randomForest")) install.packages("randomForest")
library(randomForest)
if (!require("ROSE")) install.packages("ROSE")

## Loading required package: ROSE

## Loaded ROSE 0.0-4

library(ROSE)

# Selecting relevant predictors and the target variable
predictors <- data %>% select(Ave_Age_of_Mother, Ave_Pre_pregnancy_BMI, Ave_B
irth_Weight_gms, Ave_Number_of_Prenatal_Wks, Maternal_Morbidity_YN)
target <- data$Births # Or any other relevant target variable

# Splitting data into training and test sets
set.seed(123)
trainIndex <- createDataPartition(target, p = .8, list = FALSE, times = 1)
trainData <- predictors[trainIndex, ]
testData <- predictors[-trainIndex, ]
trainTarget <- target[trainIndex]
```

```
testTarget <- target[-trainIndex]

# Training a Random Forest model
rf_model <- randomForest(trainData, as.factor(trainTarget), ntree = 100)
rf_importance <- importance(rf_model)

# Displaying feature importance
print(rf_importance)

##                            MeanDecreaseGini
## Ave_Age_of_Mother                2290.49622
## Ave_Pre_pregnancy_BMI            2243.88793
## Ave_Birth_Weight_gms             2429.82188
## Ave_Number_of_Prenatal_Wks       2202.71006
## Maternal_Morbidity_YN              27.29905
```

Ave_Birth_Weight_gms (2429.82): This variable, representing the average birth weight in grams, appears to be the most significant predictor in the model. A higher Mean Decrease Gini value indicates that this feature plays a crucial role in predicting the target variable.

Ave_Age_of_Mother (2290.50): The average age of the mother is also a strong predictor, second only to birth weight. This suggests that maternal age significantly influences the outcome of interest.

Ave_Pre_pregnancy_BMI (2243.89): The average pre-pregnancy BMI is another important predictor. Its influence is slightly less than that of maternal age but still substantial.

Ave_Number_of_Prenatal_Wks (2202.71): The average number of prenatal weeks also shows considerable importance in the model. This reflects the impact of prenatal care duration on the target variable.

Maternal_Morbidity_YN (27.30): This variable has a much lower importance score compared to others. It indicates whether maternal morbidity was noted, but it seems to have a lesser impact on the prediction of the target variable in the model.

note: - The model suggests that birth weight, maternal age, pre-pregnancy BMI, and the number of prenatal weeks are critical factors.

2. ML Algorithm Accuracy

- using radial algorithm

```
# Load additional libraries
if (!require("randomForest")) install.packages("randomForest")
library(randomForest)
if (!require("e1071")) install.packages("e1071")

## Loading required package: e1071

## Warning: package 'e1071' was built under R version 4.3.1
```

```r
library(e1071) # For SVM
if (!require("caret")) install.packages("caret")
library(caret)

# Ommitting data with Null values in order to do predicitors
data <- na.omit(data)


# Define the target variable as 'Births'
data$TargetVariable <- data$Births

# Sample a smaller subset of the data (e.g., 30% of the data)
set.seed(123)
sampled_data <- data[sample(nrow(data), nrow(data) * 0.3), ]


# Confirming there are no NA values in the data
if (sum(is.na(sampled_data)) > 0) {
    sampled_data <- na.omit(sampled_data)
}


# Sampling a smaller portion for SVM training
set.seed(123)
svm_sample_size <- round(nrow(sampled_data) * 0.1) # Example: 10% of the samp
led data
svm_indices <- sample(nrow(sampled_data), svm_sample_size)
svm_trainData <- sampled_data[svm_indices, -which(names(sampled_data) == "Bir
ths")]
svm_trainTarget <- sampled_data$Births[svm_indices]

# Convert all predictor variables to numeric, addressing non-numeric columns
svm_trainData <- data.frame(lapply(svm_trainData, function(x) {
  if (is.numeric(x) || is.integer(x)) {
    return(x)
  } else if (is.factor(x) || is.character(x)) {
    return(as.numeric(as.factor(x)))
  } else {
    # Handle other types if present
    return(as.numeric(x))
  }
}))

# Ensure target variable is numeric
svm_trainTarget <- as.numeric(svm_trainTarget)

# Check for NA values after conversion
if (sum(is.na(svm_trainData)) > 0 || sum(is.na(svm_trainTarget)) > 0) {
  stop("NA values found after conversion")
}

# Retrain the SVM model
svm_model <- svm(x = svm_trainData, y = svm_trainTarget, kernel = "radial", c
ost = 0.5)

# Checkpoint 1: Model training completed
print("SVM model trained successfully.")
```

```
## [1] "SVM model trained successfully."
```

```r
# Prepare test data for prediction
testData <- data.frame(lapply(testData, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    as.numeric(x)
  }
}))

# Checkpoint 2: Test data prepared for prediction
print("Test data prepared.")
```

```
## [1] "Test data prepared."
```

```r
testData <- sampled_data[-svm_indices, -which(names(sampled_data) == "Births"
)]
testTarget <- sampled_data$Births[-svm_indices]

# Prepare test data for prediction
testData <- data.frame(lapply(testData, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    as.numeric(x)
  }
}))


# Make predictions
svm_prediction <- predict(svm_model, testData)

# Checkpoint 3: Prediction completed
print("Prediction completed.")
```

```
## [1] "Prediction completed."
```

```r
# Calculate RMSE
testTarget <- as.numeric(testTarget)  # Ensure testTarget is numeric
svm_rmse <- sqrt(mean((svm_prediction - testTarget)^2))
```

```r
# Checkpoint 4: RMSE calculated
print("RMSE calculated.")
```

```
## [1] "RMSE calculated."
```

```r
# Output RMSE
print(svm_rmse)
```

```
## [1] 8296.19
```

-using sigmoid

```r
# Retrain the SVM model
svm_model <- svm(x = svm_trainData, y = svm_trainTarget, kernel = "sigmoid",
cost = 0.5)

# Checkpoint 1: Model training completed
print("SVM model trained successfully.")
```

```
## [1] "SVM model trained successfully."
```

```r
# Prepare test data for prediction
testData <- data.frame(lapply(testData, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    as.numeric(x)
  }
}))

# Checkpoint 2: Test data prepared for prediction
print("Test data prepared.")
```

```
## [1] "Test data prepared."
```

```r
testData <- sampled_data[-svm_indices, -which(names(sampled_data) == "Births"
)]
testTarget <- sampled_data$Births[-svm_indices]

# Prepare test data for prediction
testData <- data.frame(lapply(testData, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    as.numeric(x)
  }
}))



# Make predictions
```

```r
svm_prediction <- predict(svm_model, testData)

# Checkpoint 3: Prediction completed
print("Prediction completed.")
```

```
## [1] "Prediction completed."
```

```r
# Calculate RMSE
testTarget <- as.numeric(testTarget)  # Ensure testTarget is numeric
svm_rmse <- sqrt(mean((svm_prediction - testTarget)^2))

# Checkpoint 4: RMSE calculated
print("RMSE calculated.")
```

```
## [1] "RMSE calculated."
```

```r
# Output RMSE
print(svm_rmse)
```

```
## [1] 12857.87
```

```r
# Retrain the SVM model
svm_model <- svm(x = svm_trainData, y = svm_trainTarget, kernel = "polynomial
", cost = 0.5)

# Checkpoint 1: Model training completed
print("SVM model trained successfully.")
```

```
## [1] "SVM model trained successfully."
```

```r
# Prepare test data for prediction
testData <- data.frame(lapply(testData, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    as.numeric(x)
  }
}))

# Checkpoint 2: Test data prepared for prediction
print("Test data prepared.")
```

```
## [1] "Test data prepared."
```

```r
testData <- sampled_data[-svm_indices, -which(names(sampled_data) == "Births"
)]
testTarget <- sampled_data$Births[-svm_indices]

# Prepare test data for prediction
testData <- data.frame(lapply(testData, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    as.numeric(x)
  }
}))


# Make predictions
svm_prediction <- predict(svm_model, testData)

# Checkpoint 3: Prediction completed
print("Prediction completed.")

## [1] "Prediction completed."

# Calculate RMSE
testTarget <- as.numeric(testTarget)  # Ensure testTarget is numeric
svm_rmse <- sqrt(mean((svm_prediction - testTarget)^2))

# Checkpoint 4: RMSE calculated
print("RMSE calculated.")

## [1] "RMSE calculated."

# Output RMSE
print(svm_rmse)

## [1] 16844.82
```

2. Which machine learning algorithm provides the most accurate predictions of high-risk pregnancies based on socio-economic and racial factors? The radial algorithm did the best in rsme to train the model! This model assumption is worth looking into high pregnancies assumptions for a further deep dive.

Preprocess the data, converting non-numeric columns to numeric, ensuring the target variable is numeric, and checking for NAs. The SVM model is trained with a radial kernel function and a cost parameter of 0.5. Test data is prepared for prediction and used to make predictions. The RMSE is calculated for this model, and the result is printed.

Similar preprocessing steps are performed as in the radial algorithm. The SVM model is trained with a sigmoid kernel function and a cost parameter of 0.5. Test data is prepared, predictions are made, and RMSE is calculated for this model.

Once again, data preprocessing is carried out. The SVM model is trained with a polynomial kernel function and a cost parameter of 0.5. Test data is prepared, predictions are made, and RMSE is calculated for this model.