

Untitled

Preethi Natesan Sridhar

2023-10-16

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 necessary libraries
library(caret)
```

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

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

```
library(rpart)
library(rpart.plot)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

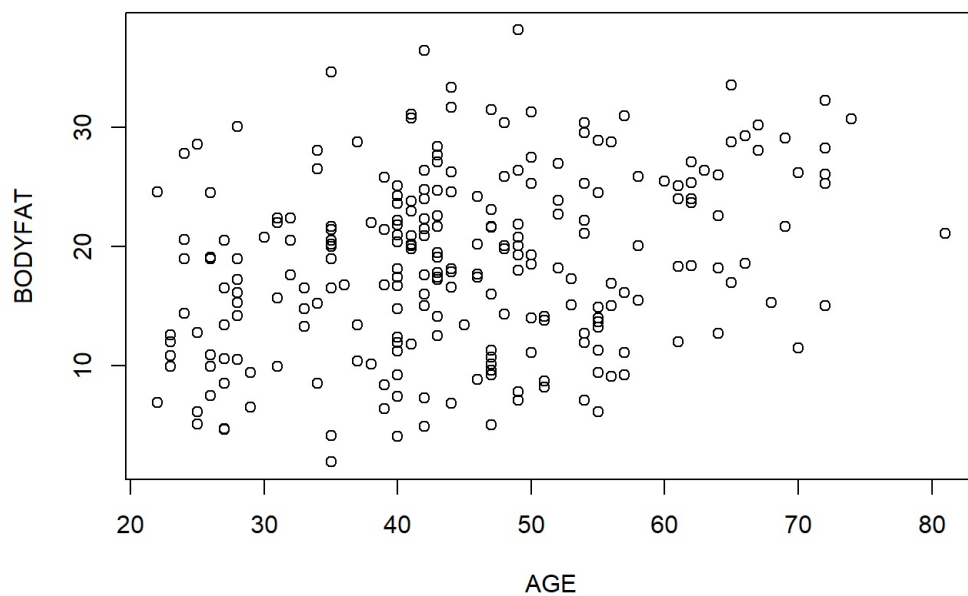
```
# Read the CSV file
data <- read.csv("C:/Users/preet/Downloads/cleaned_bodyfatfinal.csv")

# Define selected features
selectedColumns <- c('AGE', 'WEIGHT', 'HEIGHT', 'NECK', 'CHEST', 'ABDOMEN', 'HIP', 'THIGH', 'KNEE', 'ANKLE', 'BICEPS', 'FOREARM', 'WRIST')

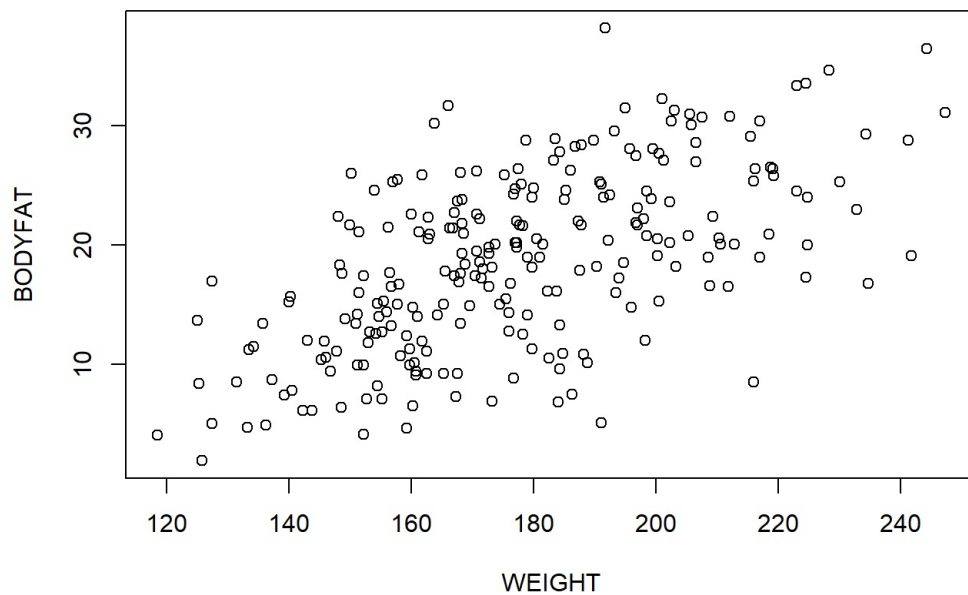
# Extract selected features and target variable
selectedData <- data[selectedColumns]
target <- data$BODYFAT

# Plot each feature against the target variable
for (column in selectedColumns) {
  plot(data[[column]], target, main=paste(column, "vs BODYFAT"), xlab=column, ylab="BODYFAT")
}
```

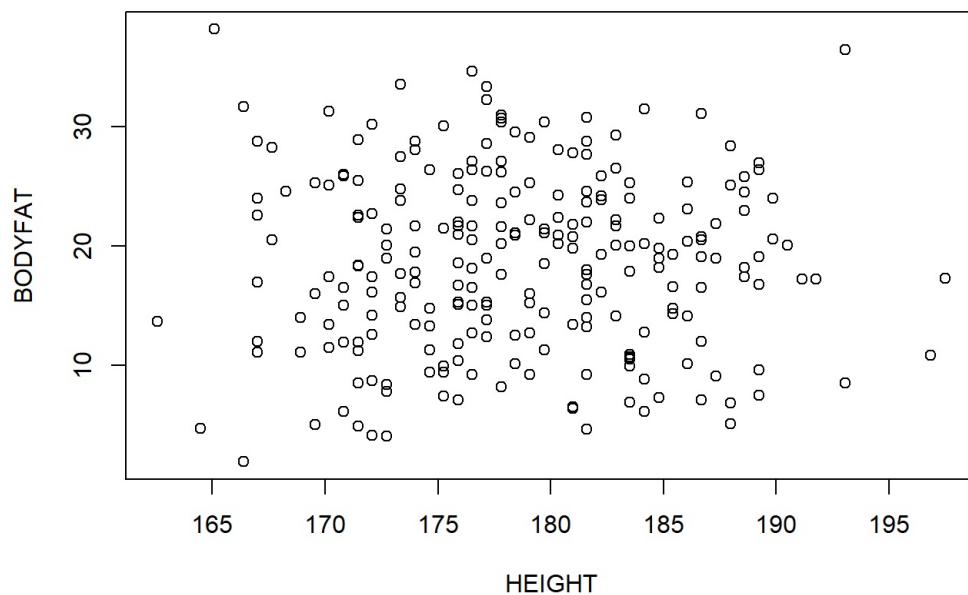
AGE vs BODYFAT



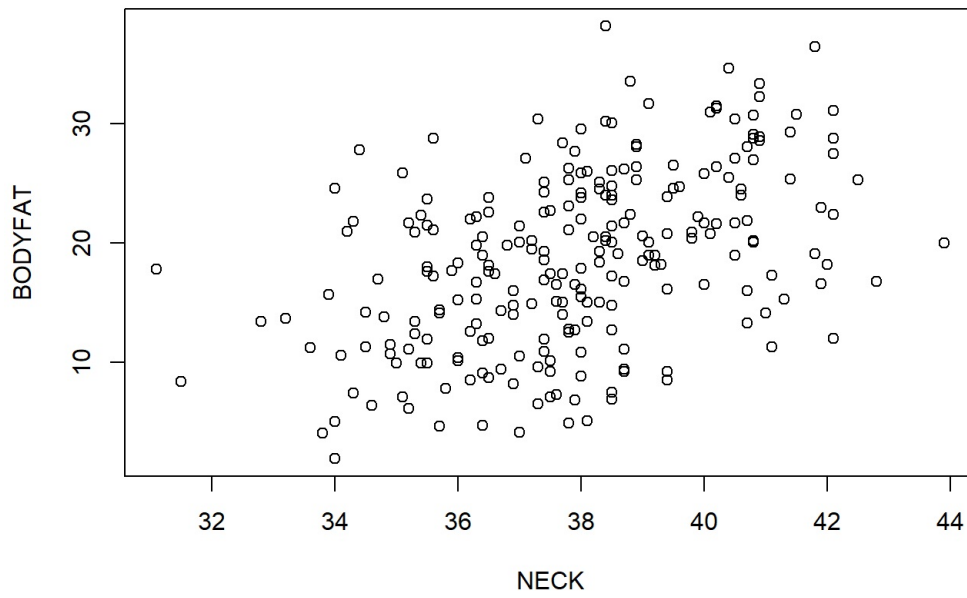
WEIGHT vs BODYFAT



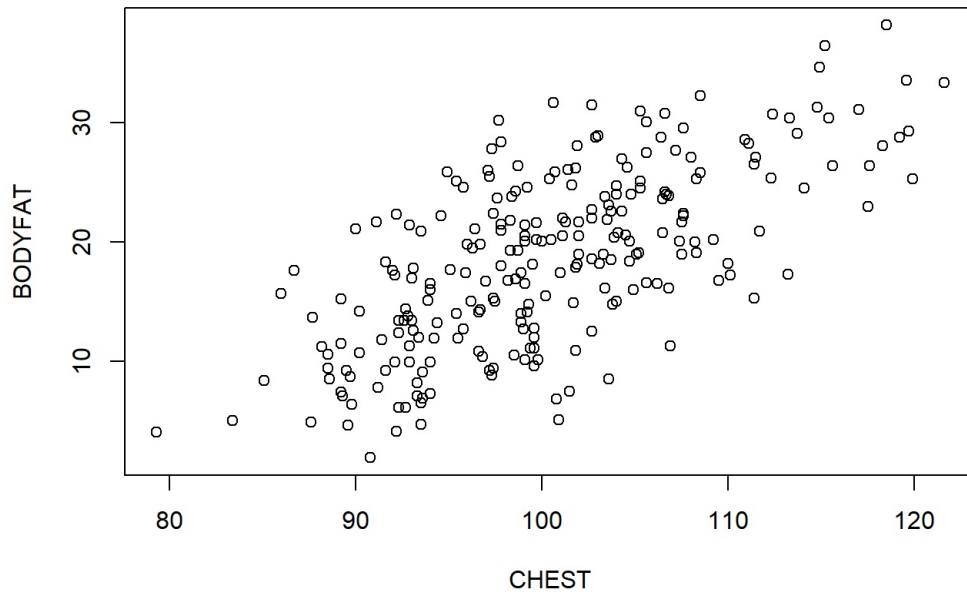
HEIGHT vs BODYFAT



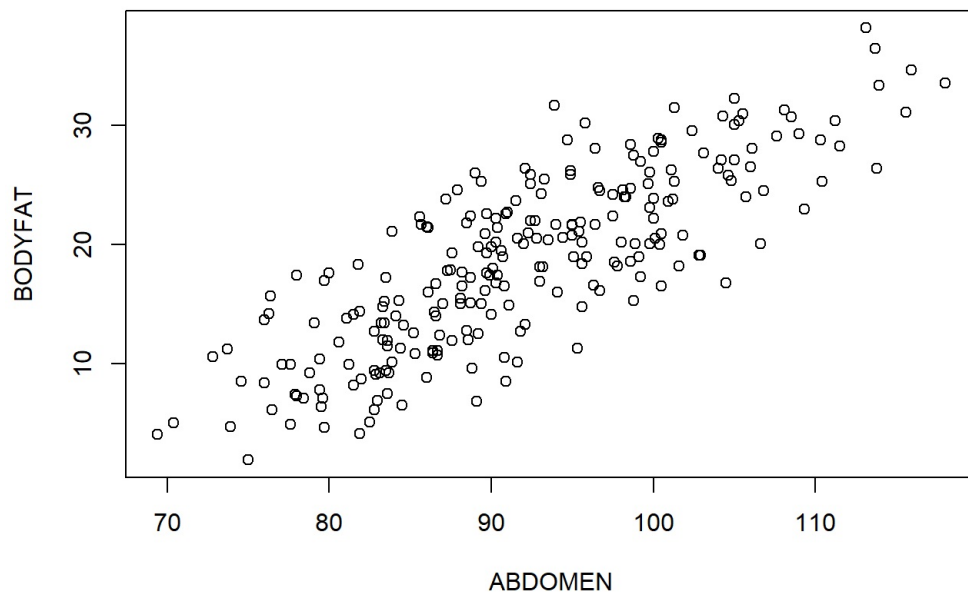
NECK vs BODYFAT



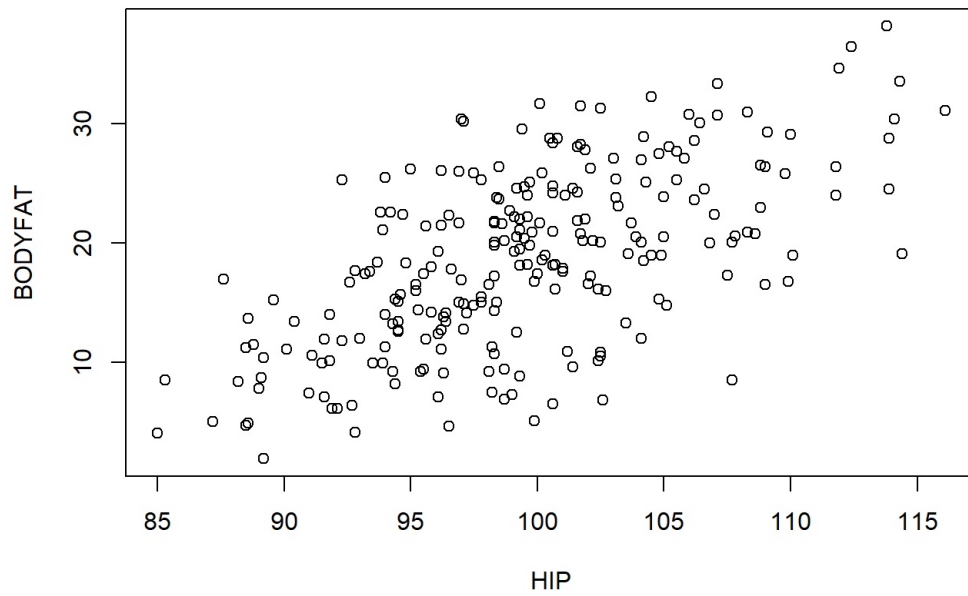
CHEST vs BODYFAT



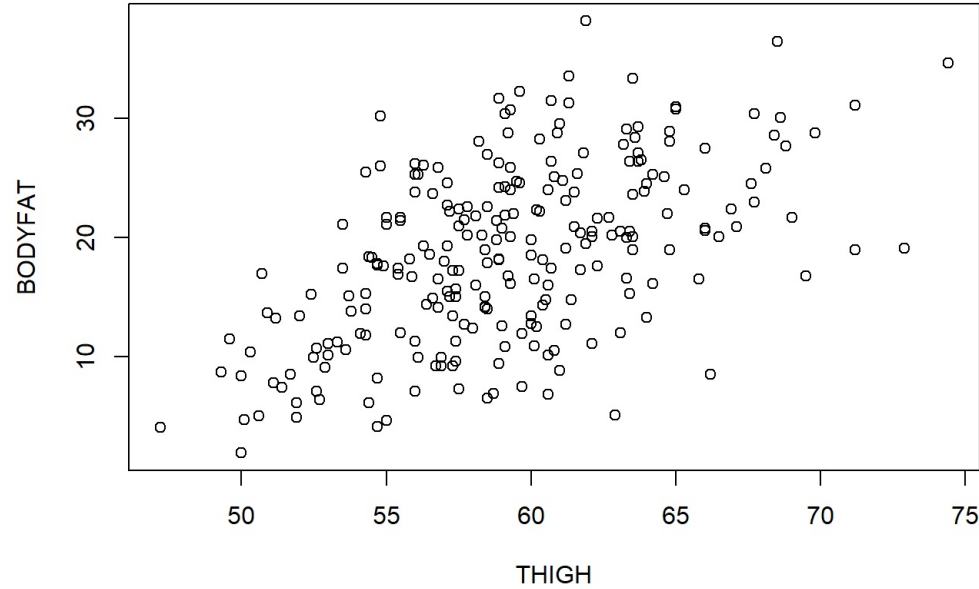
ABDOMEN vs BODYFAT



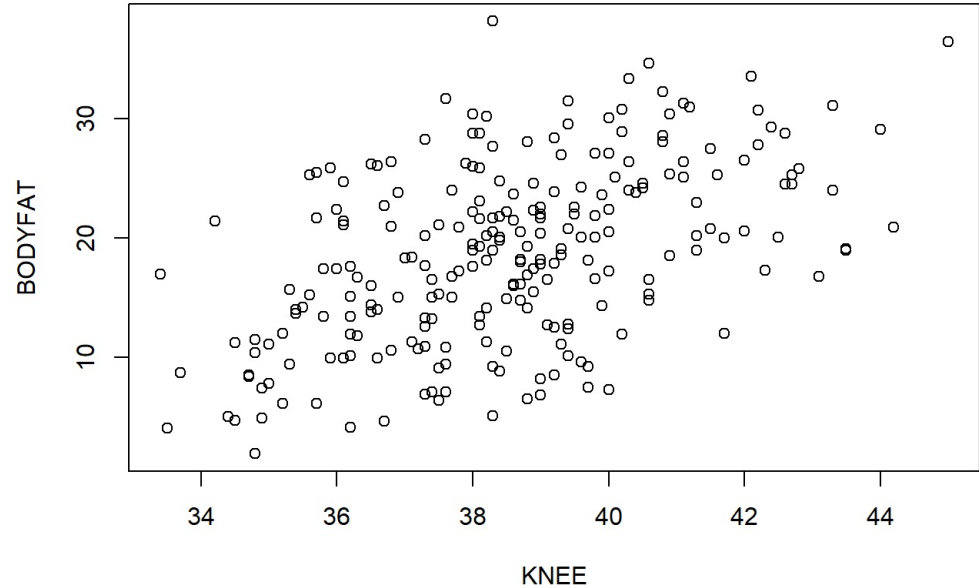
HIP vs BODYFAT



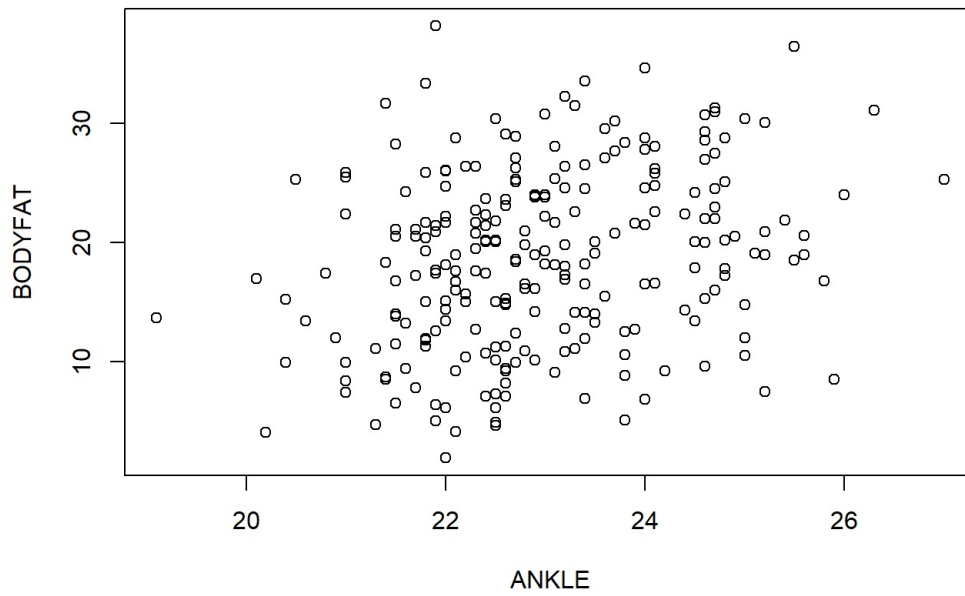
THIGH vs BODYFAT



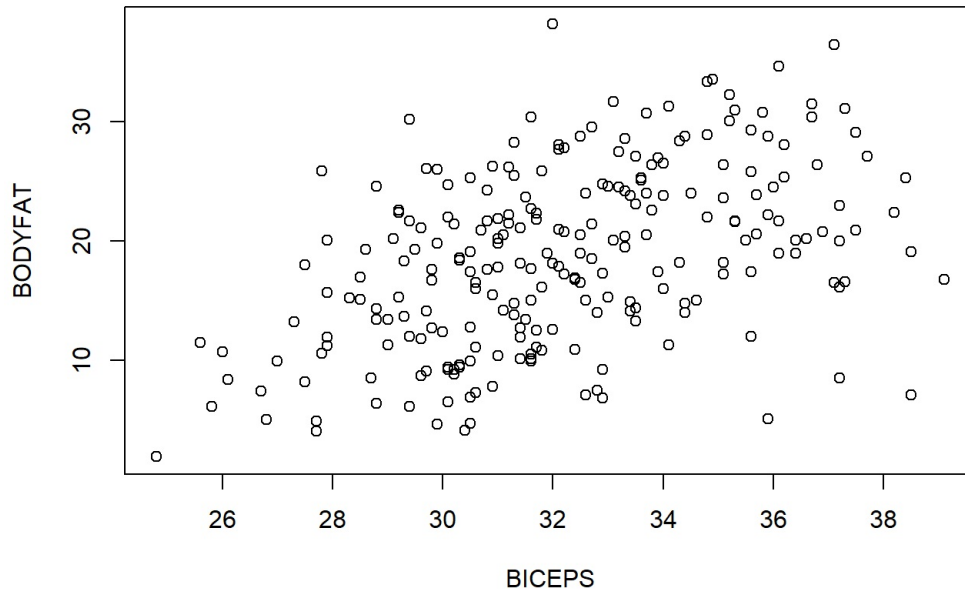
KNEE vs BODYFAT



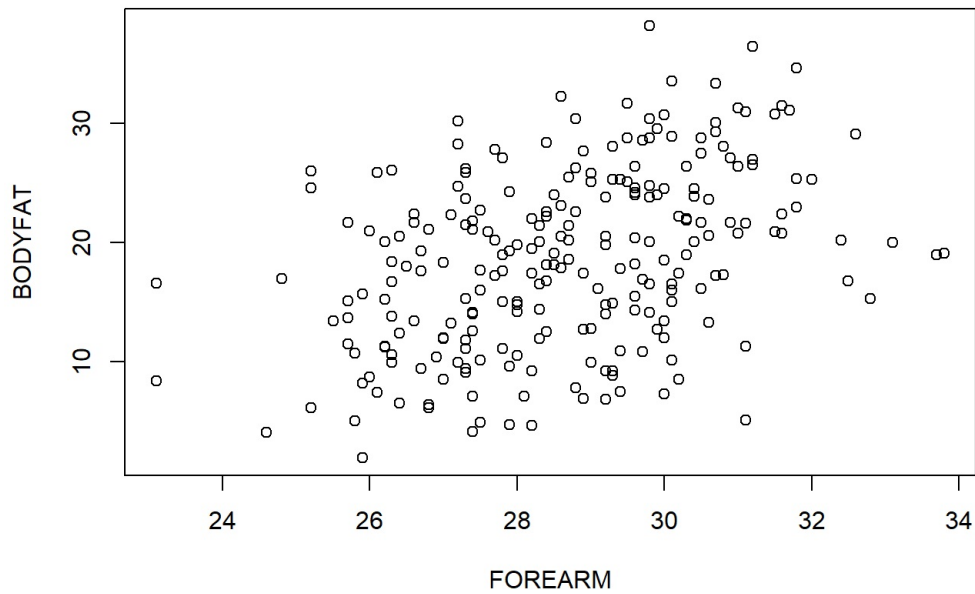
ANKLE vs BODYFAT



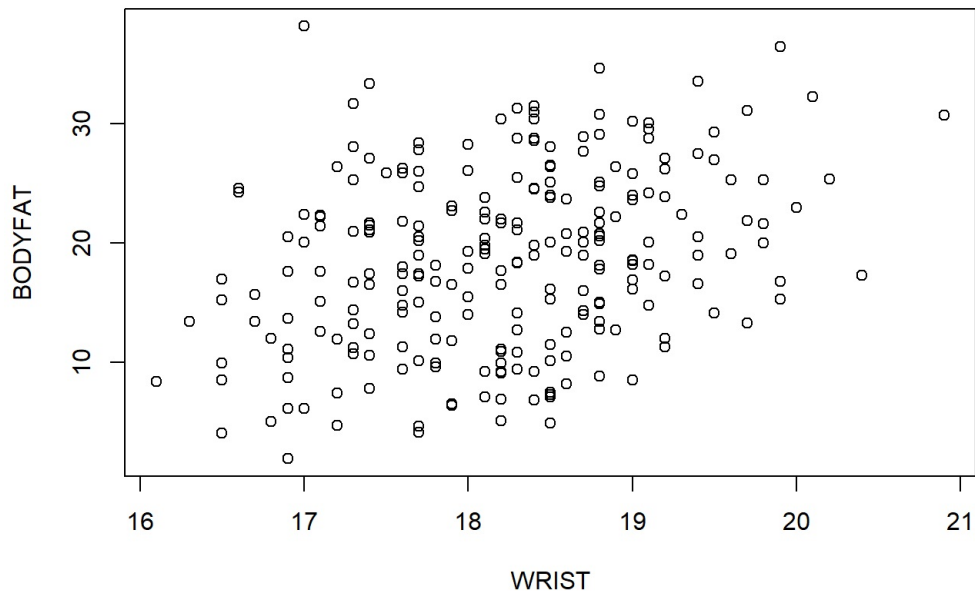
BICEPS vs BODYFAT



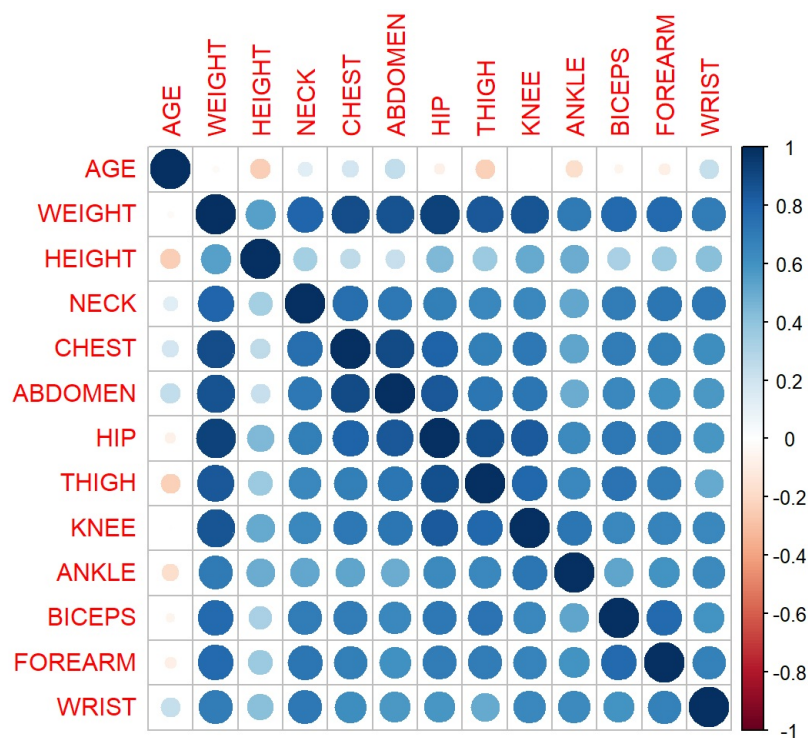
FOREARM vs BODYFAT



WRIST vs BODYFAT



```
# Calculate correlation matrix and plot it
correlationMatrix <- cor(selectedData)
corrplot(correlationMatrix, method="circle")
```



```
# Select features with correlation above a certain threshold
correlationThreshold <- 0.5 # Adjust this value as needed
highlyCorrelatedFeatures <- selectedColumns[apply(correlationMatrix, 1, function(x) any(abs(x) > correlationThreshold))]

# Extract highly correlated features
selectedData <- data[highlyCorrelatedFeatures]

# Configure k-fold cross-validation
numFolds <- 5
set.seed(123) # Set seed for reproducibility

# Initialize arrays to store MSE and R-squared values
mseValues <- numeric(numFolds)
rsquaredValues <- numeric(numFolds)

# Perform k-fold cross-validation
folds <- createFolds(target, k = numFolds, list = TRUE, returnTrain = FALSE)

for (fold in 1:numFolds) {
  testIndices <- unlist(folds[fold])
  trainData <- selectedData[-testIndices, ]
  trainTarget <- target[-testIndices]
  testData <- selectedData[testIndices, ]
  testTarget <- target[testIndices]

  # Build a decision tree model for this fold
  model <- rpart(trainTarget ~ ., data = trainData)

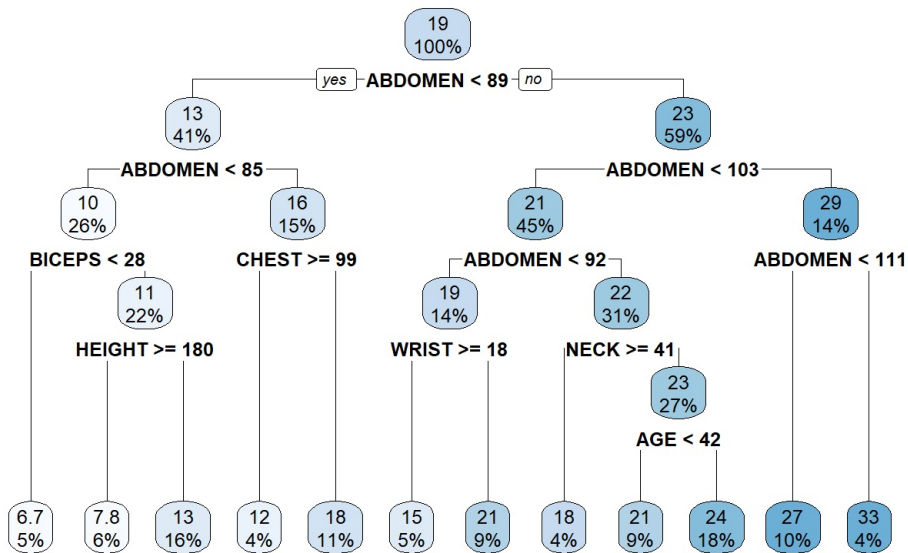
  # Display the decision tree plot for this fold
  rpart.plot(model, main=paste("Decision Tree for Fold", fold))

  # Make predictions
  predictions <- predict(model, newdata = data.frame(testData))

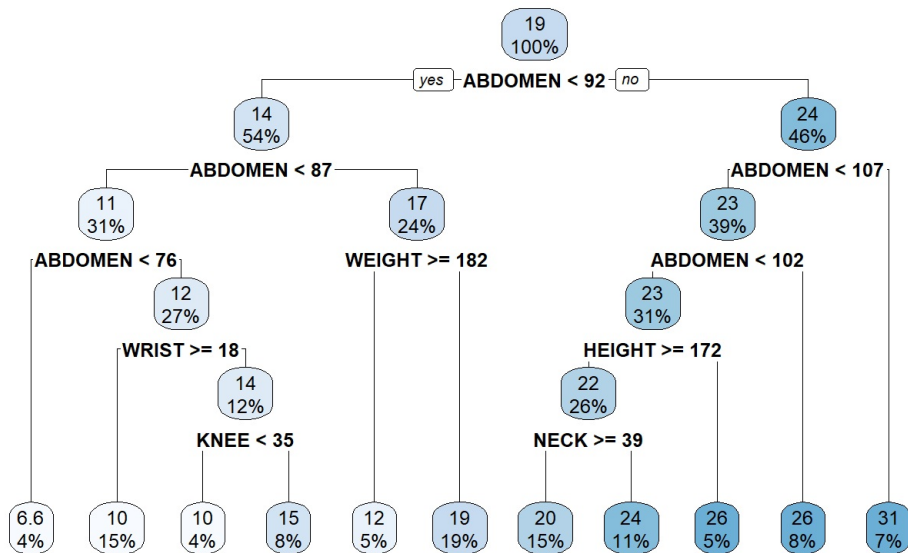
  # Calculate Mean Squared Error (MSE) for this fold
  mseValues[fold] <- mean((testTarget - predictions)^2)

  # Calculate R-squared for this fold
  sst <- sum((testTarget - mean(testTarget))^2) # Total sum of squares
  ssr <- sum((testTarget - predictions)^2) # Residual sum of squares
  rsquaredValues[fold] <- 1 - (ssr / sst)
}
```

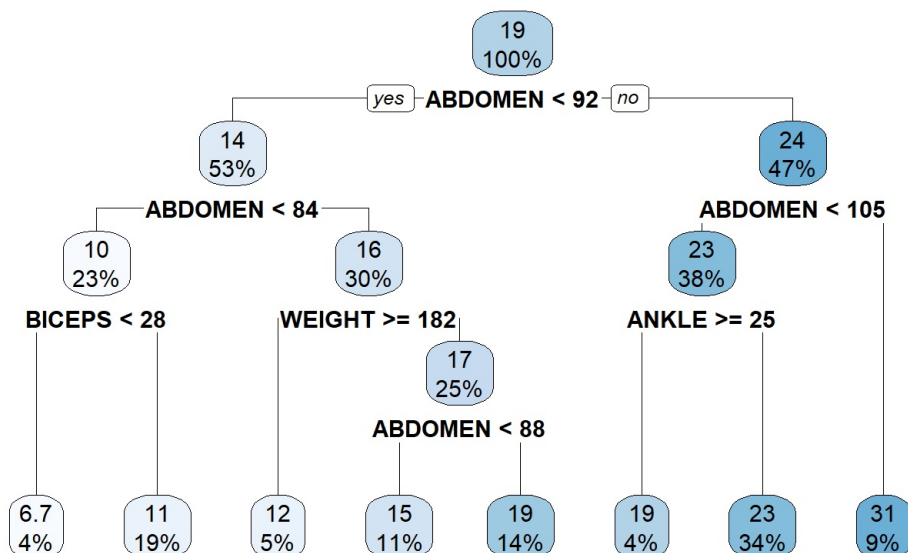

Decision Tree for Fold 1



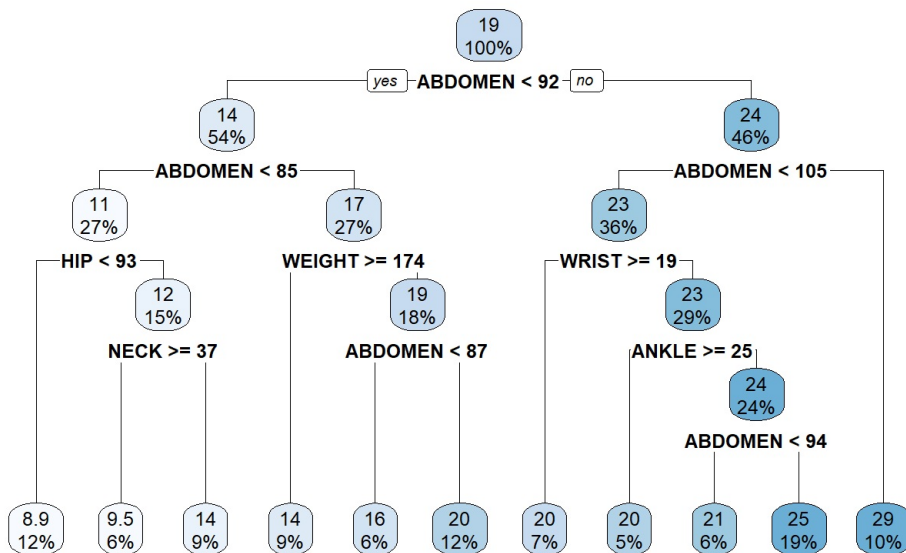
Decision Tree for Fold 2



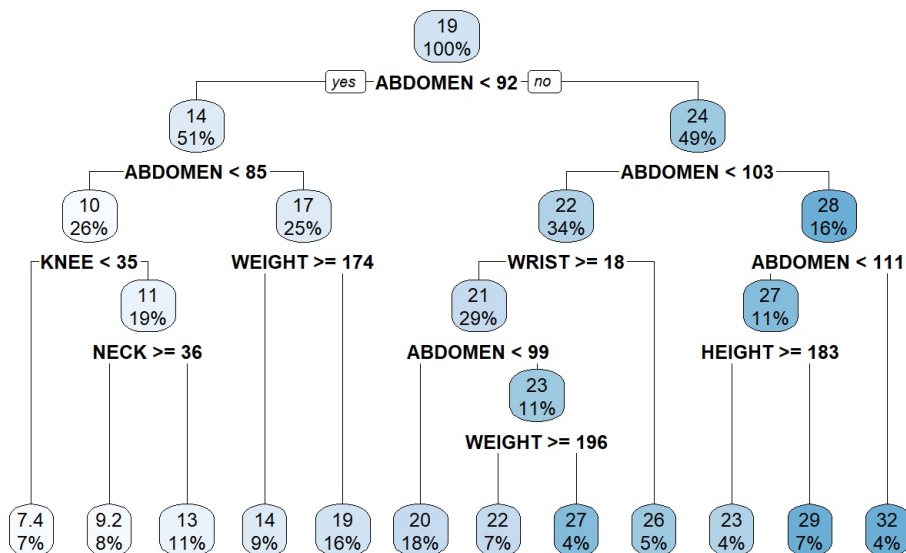
Decision Tree for Fold 3



Decision Tree for Fold 4



Decision Tree for Fold 5



```
# Calculate the average MSE and R-squared values across all folds
```

```
averageMSE <- mean(mseValues)
```

```
averageRSquared <- mean(rsquaredValues)
```

```
cat("Mean Squared Error (MSE) across", numFolds, "-fold cross-validation:", averageMSE, "\n")
```

```
## Mean Squared Error (MSE) across 5 -fold cross-validation: 22.77934
```

```
cat("R-squared across", numFolds, "-fold cross-validation:", averageRSquared, "\n")
```

```
## R-squared across 5 -fold cross-validation: 0.5843284
```