Project Report

ASSESSING THE SIGNIFICANT MORPHOLOGICAL FACTORS ASSOCIATED WITH DEVELOPMENT OF BREAST CANCER

Aruna Prasanna Simhambhatla
Parvathi Dandibhotla
SuneethMuthineni
Vignitha Maddipatla
YaznaPenmetsa

Professor - Dr.Zeyana Hamid

Project Report

Introduction to Problem statement

Breast cancer is a significant global health concern. Early and accurate detection is crucial for effective treatment and improved survival rates (Dua and Graff, 2019). This challenge often leads to unnecessary biopsies or missed diagnoses (Adorada et al., 2018).
The aim of our study is to analyze the association between morphological characteristics and the likelihood of developing malignant or benign breast cancer.
The research objective is to conduct a statistical analysis to evaluate the relationship between morphological characteristics of breast tumors and the probability of developing malignant or benign breast cancer, elucidating key factors contributing to diagnostic precision and clinical decision-making. The long-term objective is to improve patient care by improving breast cancer diagnostics, improving risk stratification, and customizing treatment using morphological analysis insights.

Research Question

Given the diagnostic challenges, our study focuses on the following question: "Do morphological characteristics of breast tumors observed in FNA samples have a significant association with the likelihood of the tumors being malignant or benign?" This question aims to uncover whether specific morphological features can serve as reliable indicators for breast cancer diagnosis.

Hypothesis

Null Hypothesis (H0): There is no significant association between the morphological characteristics of breast tumors and their likelihood of being malignant or benign.
Alternative Hypothesis (H1): There is a significant association between the morphological characteristics of breast tumors and their likelihood of being malignant or benign.

Our study is predicted to have a significant impact, with the potential to improve diagnostic techniques and lessen the need for invasive operations like biopsies. Our results might have a major impact on patient care by enabling more prompt and tailored treatment methods, which would eventually improve survival rates and patient quality of life. This would be achieved by increasing the accuracy of early screens.

Data description

•The Wisconsin Diagnostic Breast Cancer (WDBC) dataset contains 569 instances with 30 numerical features extracted from fine needle aspirate (FNA) images and longitudinal measurements.Data Set contains data related to morphological features and breast cancer as diagnosis.

•The data set contains no null values and duplicate values. It is clean and structured with mean values as variables.

Project Report

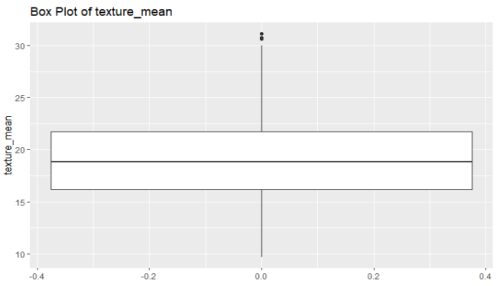
Variable type	Variable	Description
Numerical	radius_mean	Measurements related to the size and shape of the nucleus (radius).
Numerical	texture_mean	Measurements related to the texture of the nucleus (smoothness, uniformity of staining).
Numerical	perimeter_mean	Measurements related to the perimeter of the nucleus.
Numerical	area_mean	Measurements related to the area of the nucleus.
Numerical	smoothness_mean	Level of smoothness of the nucleus periphery.
Numerical	compactness_mean	Ratio of the nucleus area to its equivalent circle area.
Numerical	symmetry_mean	Degree of symmetry of the nucleus shape.
Numerical	fractal dimension_mean	Complexity of the nucleus shape.
Categorical	diagnosis	Indicates whether the tumor is malignant (M) or benign (B).
Categorical	tumour_size	We have created this variable on assumption of based on quartile ranges in existing numerical variable radius_mean.

Assumption: We created the "tumor_size" variable based on assumption, using quartile ranges from "radius_mean" for enhanced visualizations and analyses, given the absence of literature evidence and lack of units in our mean value variables.

Methodology: We have started data cleaning by function clean_names to remove empty rows and columns and then removed duplicated rows and columns with function remove_empty

Data cleaning: We have started data cleaning by function clean_names to remove empty rows and columns and then removed duplicated rows and columns with function remove_empty which revealed that our data doesn't have any null values.
We have dropped some columns in the data set and treated the outliers with 3 sigma rule.

Project Report




```r
# Ensure the ggplot2 package is installed and loaded
if (!require(ggplot2)) install.packages('ggplot2')

# List of variables
variables <- c("radius_mean", "texture_mean", "perimeter_mean", "area_mean",
               "smoothness_mean", "compactness_mean", "symmetry_mean", "fractal_dimension_mean")

# Outlier treatment function
treat_outliers <- function(x) {
  mean_x <- mean(x, na.rm = TRUE)
  sd_x <- sd(x, na.rm = TRUE)
  threshold <- mean_x + 3 * sd_x
  x[x > threshold] <- NA
  return(x)
}

# Apply outlier treatment and create box plots for each variable
plots <- lapply(variables, function(var) {
  Breastcancer_data2[[var]] <- treat_outliers(Breastcancer_data2[[var]]) # Apply outlier treatment
  ggplot(Breastcancer_data2, aes_string(y = var)) +
    geom_boxplot() +
    labs(title = paste("Box Plot of", var), y = var)
})

# Display the plots
plots
```

Descriptive statistics

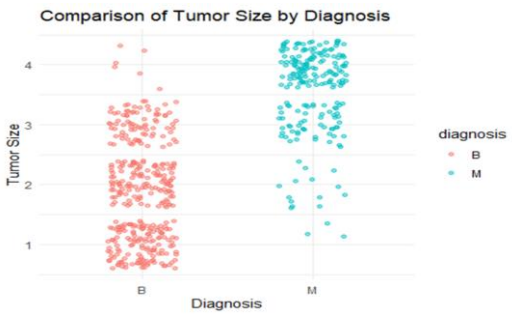
We performed descriptive statistics by using the function summary() to obtain quartiles, mean, median of each variable.

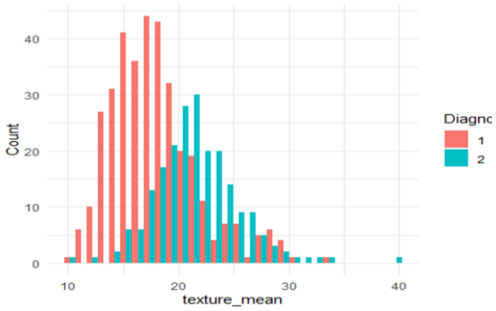
We conducted a Shapiro-Wilk test to assess normality, revealing that the variables are non-normally distributed.

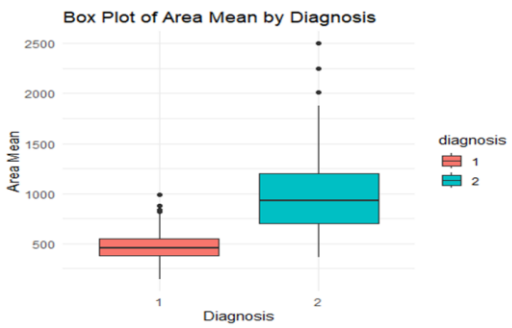
Then we plotted graphs for skewness which revealed that all our variables are right skewed.

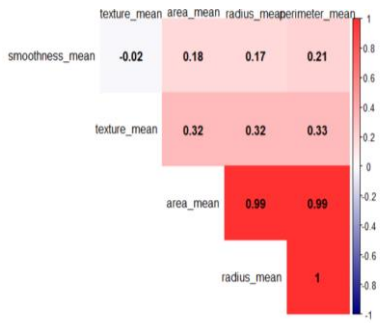
Exploratory Data Analysis and Visualizations

Here are some visualizations of our data.









Above scatter plot showing that malignant tumors are more influenced by tumor size compared to benign tumors.

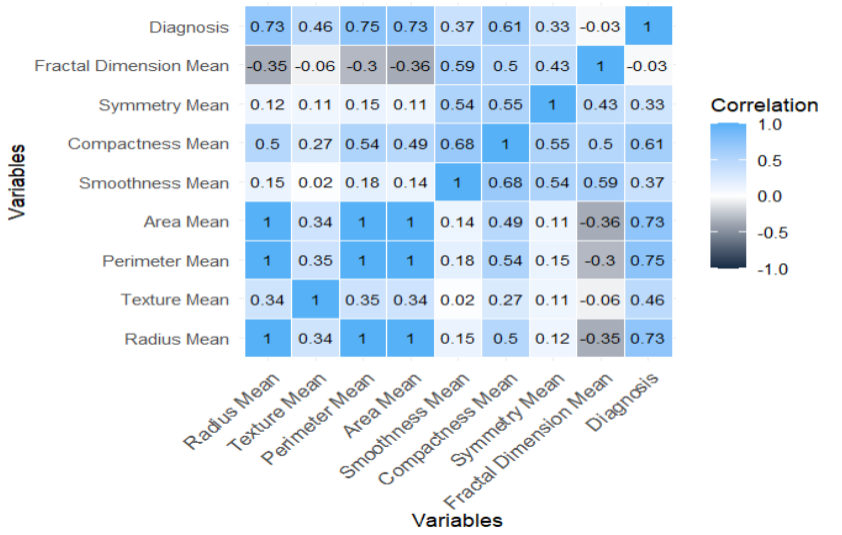
The box plot showing that malignant tumor has more area_mean than benign.

The heatmap displays the correlation coefficients among various morphological features such as 'radius_mean', 'area_mean', 'perimeter_mean', and others.

Project Report

Correlation Matrix

We used spearman correlation method and we found strong positive correlations between radius_mean, perimeter_mean, area_mean, compactness_mean and diagnosis.



Statistical Analysis

Mann-Whitney U Test: The non-parametric Mann-Whitney U test was conducted for all numerical variables to assess their association with the outcome variable 'diagnosis'. These tests revealed a of p value < 0.05 for all variables except fractal_dimension_mean stating significant association between predictor variables and outcome variable "diagnosis."

```
[1] "Mann-Whitney U test for smoothness_mean:"

        Wilcoxon rank sum test with continuity correction

data:  smoothness_mean by diagnosis
W = 21037, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for smoothness_mean: Significant association"
```

```
[1] "Mann-Whitney U test for fractal_dimension_mean:"

        Wilcoxon rank sum test with continuity correction

data:  fractal_dimension_mean by diagnosis
W = 39013, p-value = 0.5372
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for fractal_dimension_mean: No significant association"
```

Chi-square Test: A chi-square test was performed using the categorical variable 'tumor_size' and the outcome variable 'diagnosis' to examine the impact of tumor size on the likelihood of developing malignant or benign tumors. The obtained Malignant p-value < 2.2e-16 suggesting a significant association between tumor_size and malignant tumours. The obtained Benign p-value < 2.2e-16 suggesting significant association between tumor_size and benign tumours

```
# Create a contingency table for tumor_size and diagnosis
contingency_table_tumor_size <- table(Breastcancer_data2$tumor_size,
Breastcancer_data2$diagnosis)

# Perform chi-square test of independence for tumor_size and diagnosis
chi_square_tumor_size <- chisq.test(contingency_table_tumor_size)
print("Chi-square test for tumor_size and diagnosis:")
print(chi_square_tumor_size)

# Determine association result for tumor_size
association_result_tumor_size <- ifelse(chi_square_tumor_size$p.value < 0.0
"Significant association", "No significant association")
print(paste("Association result for tumor_size:",
association_result_tumor_size))
```

```
[1] "Chi-square test for 2 :"

        Chi-squared test for given probabilities

data:  cont_table
X-squared = 206.53, df = 3, p-value < 2.2e-16

[1] "Chi-square test for 1 :"

        Chi-squared test for given probabilities

data:  cont_table
X-squared = 123.77, df = 3, p-value < 2.2e-16
```

Project Report

Machine Learning Model

Logistic Regression Model Training:

Utilized the glm() function to construct a logistic regression model, with diagnosis as the dependent variable and all other variables as independent predictors. This model aimed to uncover the relationships between various features of the dataset and the likelihood of a breast tumor being malignant, based on training data. In the model summary some of the variables show significant p values. The AIC value of this model is 153.51.

```
Call:
glm(formula = diagnosis ~ ., family = binomial(), data = train_data)

Coefficients: (1 not defined because of singularities)
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          1.100e+01  1.383e+01   0.795 0.426644
id                  -4.446e-09  4.914e-09  -0.905 0.365551
radius_mean         -6.822e+00  3.781e+00  -1.804 0.071166 .
texture_mean         3.694e-01  6.820e-02   5.416 6.09e-08 ***
perimeter_mean       1.221e-01  5.360e-01   0.228 0.819790
area_mean            8.439e-02  2.153e-02   3.919 8.88e-05 ***
smoothness_mean      9.816e+01  2.632e+01   3.729 0.000192 ***
compactness_mean     2.320e+01  2.367e+01   0.980 0.326943
symmetry_mean        3.040e+01  1.341e+01   2.267 0.023377 *
fractal_dimension_mean -9.267e+01  9.473e+01  -0.978 0.327981
tumor_sizeSmall Tumors  2.383e+00  1.421e+00   1.677 0.093623 .
tumor_sizeMedium Tumors 2.596e+00  1.784e+00   1.455 0.145631
tumor_sizeLarge Tumors  6.610e-01  2.512e+00   0.263 0.792411
tumor_size_numerical           NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 602.31  on 455  degrees of freedom
Residual deviance: 127.51  on 443  degrees of freedom
AIC: 153.51

Number of Fisher Scoring iterations: 9
```

```
          Reference
Prediction  1  2
         1 69  2
         2  2 40
 Accuracy
0.9646018
Confusion Matrix and Statistics

          Reference
Prediction  1  2
         1 69  2
         2  2 40

               Accuracy : 0.9646
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9242

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9718
            Specificity : 0.9524
         Pos Pred Value : 0.9718
         Neg Pred Value : 0.9524
             Prevalence : 0.6283
         Detection Rate : 0.6106
   Detection Prevalence : 0.6283
      Balanced Accuracy : 0.9621

       'Positive' Class : 1
```

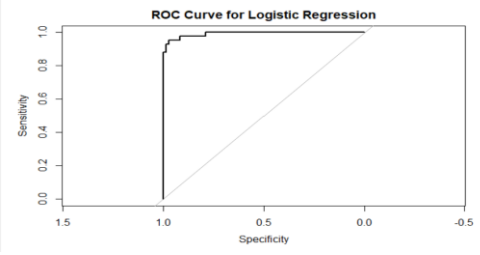
Model Evaluation:

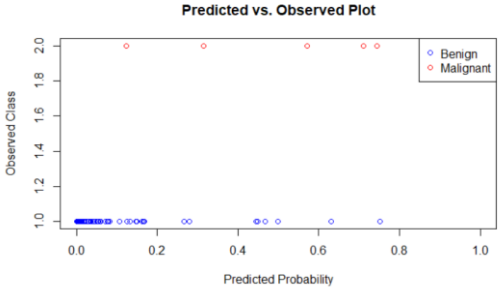
Probability Predictions: The model was then applied to a separate test dataset to estimate the probabilities that each case was malignant. This step is crucial for understanding how well the model can generalize its predictions to new, unseen data.

Class Predictions: These probabilities were converted into binary class predictions. A threshold of 0.5 was used, where probabilities above this value led to a diagnosis prediction of malignant, and those below indicated benign. This binary classification facilitates easier clinical decision-making.

Visualization of Model Predictions:

Predicted vs. Observed Plot: Developed a scatter plot comparing the predicted probabilities to the actual diagnoses in the test data. This visual representation is essential for evaluating the predictive performance of the model. Points in the plot are color-coded—red for benign and blue for malignant—to visually delineate the accuracy of predictions against true classifications. This plot not only confirms the model's effectiveness but also highlights any potential prediction errors or biases.





Project Report

Conclusion:

We have enough evidence to reject the null hypothesis as some variables show significant association with the outcome variable. Thus, we fail to reject alternate hypothesis that there is a significant association between morphological characteristics and the likelihood of developing malignant or benign breast cancer.

The significant associations observed from above analysis suggest that some morphological characteristics, such as radius, texture, perimeter, area, smoothness, compactness, and symmetry, play a crucial role in estimating association with the likelihood of developing malignant or benign breast cancer.

However, the fractal_dimension_mean did not exhibit a significant association with either malignant or benign tumors.

These findings provide evidence supporting the hypothesis that morphological characteristics have a significant association with the likelihood of developing malignant or benign breast cancer.

Limitations:

1. The small sample size significantly restricts the applicability of machine learning models, especially for conducting individualized analyses to association of malignant and benign tumors with predictor variables.
2.The broad nature of the research question adds complexity to the analysis process, potentially requiring more extensive data and computational resources.
3.The absence of units for variables in the dataset presents challenges in interpreting the numerical values accurately. Moreover, the lack of supported literature makes it difficult to validate findings and methodologies effectively.

Project Report

References

Adorada, A., Permatasari, R., Wirawan, P. W., Wibowo, A., & Sujiwo, A. (2018). Support vector machine-recursive feature elimination (svm-rfe) for selection of microrna expression features of breast cancer. 2018 2nd International Conference on Informatics and Computational Sciences (ICICoS).

UCI machine learning repository. (n.d.). Uci.edu. Retrieved May 1, 2024, from https://archive.ics.uci.edu/

Appendix:

```{r}
Data <- read.csv("C:\\Users\\sunee\\Desktop\\data.csv")
# Rename the data frame to "Breast Cancer Data Set"
colnames(Data) <- "Breast Cancer Data Set"                # renaming data set
```

```{r}
# Read the data and rename the data frame
`Breastcancerdata` <- read.csv("C:\\Users\\sunee\\Desktop\\data.csv")       #reading dataset

```

```{r}
library(janitor)
```

```{r}
Breastcancer_data2 <- clean_names(Breastcancerdata)

```

```{r}

Breastcancer_data2 <- remove_empty(Breastcancer_data2, which = c("rows", "cols"), quiet = FALSE) # removing duplicate columns
```

```
No empty rows to remove.
Removing 1 empty columns of 33 columns total (Removed: x).
```

```{r}
# Load the dplyr package
library(dplyr)

# Assuming your dataset is named "data3", remove duplicate rows
Breastcancer_data2<- distinct(Breastcancer_data2)


```

```{r}
# Get the dimensions (shape) of the dataset
dim(Breastcancer_data2)

# Get a summary description of the dataset
summary(Breastcancer_data2)
```

```
[1] 569  32
       id            diagnosis          radius_mean      texture_mean    perimeter_mean     area_mean      smoothness_mean   compactness_mean
 Min.   :     8670   Length:569         Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.   :0.01938
 1st Qu.:   869218   Class :character   1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492
 Median :   906024   Mode  :character   Median :13.370   Median :18.84   Median : 86.24   Median : 551.1   Median :0.09587   Median :0.09263
 Mean   : 30371831                      Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean   :0.10434
 3rd Qu.:  8813129                      3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040
 Max.   :911320502                      Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.   :0.34540
 concavity_mean    concave_points_mean symmetry_mean    fractal_dimension_mean   radius_se        texture_se      perimeter_se       area_se
 Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04996        Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
 1st Qu.:0.02956   1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850
 Median :0.06154   Median :0.03350     Median :0.1792   Median :0.06154        Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530
 Mean   :0.08880   Mean   :0.04892     Mean   :0.1812   Mean   :0.06280        Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337
 3rd Qu.:0.13070   3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190
 Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09744        Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200
 smoothness_se      compactness_se      concavity_se     concave_points_se   symmetry_se       fractal_dimension_se  radius_worst   texture_worst
 Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02
 1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08
 Median :0.006380   Median :0.020450   Median :0.02589   Median :0.010930   Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41
 Mean   :0.007041   Mean   :0.025478   Mean   :0.03189   Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68
 3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72
 Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54
 perimeter_worst    area_worst      smoothness_worst  compactness_worst concavity_worst  concave_points_worst symmetry_worst   fractal_dimension_worst
 Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000      Min.   :0.1565   Min.   :0.05504
 1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146
 Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993      Median :0.2822   Median :0.08004
 Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461      Mean   :0.2901   Mean   :0.08395
 3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208
 Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100      Max.   :0.6638   Max.   :0.20750
```

```{r}
# Columns to be dropped
columns_to_drop <- c("concavity_mean", "concave_points_mean", "radius_se", "texture_se", "perimeter_se", "area_se",
                     "smoothness_se", "compactness_se", "concavity_se",
                     "concave_points_se", "symmetry_se", "fractal_dimension_se",
                     "radius_worst", "texture_worst", "perimeter_worst",
                     "area_worst", "smoothness_worst", "compactness_worst",
                     "concavity_worst", "concave_points_worst", "symmetry_worst",
                     "fractal_dimension_worst")

# Drop the specified columns from the Breast_cancer dataset
Breastcancer_data2 <- Breastcancer_data2[, !names(Breastcancer_data2) %in% columns_to_drop]


```

```{r}

# Normality test for radius_mean variable using Shapiro-Wilk test
print("Normality test for radius_mean:")
shapiro_test_radius <- shapiro.test(Breastcancer_data2$radius_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_radius$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_radius$p.value > 0.05) {
  print("Data is normally distributed for radius_mean")
} else {
  print("Data is not normally distributed for radius_mean")
}
```

```
 [1] "Normality test for radius_mean:"
 [1] "Shapiro-Wilk test p-value: 3.10564350835627e-14"
 [1] "Data is not normally distributed for radius_mean"
```

```{r}
# Normality test for texture_mean variable using Shapiro-Wilk test
print("Normality test for texture_mean:")
shapiro_test_texture <- shapiro.test(Breastcancer_data2$texture_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_texture$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_texture$p.value > 0.05) {
  print("Data is normally distributed for texture_mean")
} else {
  print("Data is not normally distributed for texture_mean")
}
```

```
 [1] "Normality test for texture_mean:"
 [1] "Shapiro-Wilk test p-value: 7.2835810327422e-08"
 [1] "Data is not normally distributed for texture_mean"
```

```{r}
# Normality test for perimeter_mean variable using Shapiro-Wilk test
print("Normality test for perimeter_mean:")
shapiro_test_perimeter <- shapiro.test(Breastcancer_data2$perimeter_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_perimeter$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_perimeter$p.value > 0.05) {
  print("Data is normally distributed for perimeter_mean")
} else {
  print("Data is not normally distributed for perimeter_mean")
}
```

```
 [1] "Normality test for perimeter_mean:"
 [1] "Shapiro-Wilk test p-value: 7.01140152099188e-15"
 [1] "Data is not normally distributed for perimeter_mean"
```

```{r}
# Normality test for area_mean variable using Shapiro-Wilk test
print("Normality test for area_mean:")
shapiro_test_area <- shapiro.test(Breastcancer_data2$area_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_area$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_area$p.value > 0.05) {
  print("Data is normally distributed for area_mean")
} else {
  print("Data is not normally distributed for area_mean")
}

```

```
 [1] "Normality test for area_mean:"
 [1] "Shapiro-Wilk test p-value: 3.19626432303972e-22"
 [1] "Data is not normally distributed for area_mean"
```

```{r}
# Normality test for smoothness_mean variable using Shapiro-Wilk test
print("Normality test for smoothness_mean:")
shapiro_test_smoothness <- shapiro.test(Breastcancer_data2$smoothness_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_smoothness$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_smoothness$p.value > 0.05) {
  print("Data is normally distributed for smoothness_mean")
} else {
  print("Data is not normally distributed for smoothness_mean")
}
```

```
 [1] "Normality test for smoothness_mean:"
 [1] "Shapiro-Wilk test p-value: 8.60083255698697e-05"
 [1] "Data is not normally distributed for smoothness_mean"
```

```{r}
# Normality test for compactness_mean variable using Shapiro-Wilk test
print("Normality test for compactness_mean:")
shapiro_test_compactness <- shapiro.test(Breastcancer_data2$compactness_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_compactness$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_compactness$p.value > 0.05) {
  print("Data is normally distributed for compactness_mean")
} else {
  print("Data is not normally distributed for compactness_mean")
}
```

```
[1] "Normality test for compactness_mean:"
[1] "Shapiro-Wilk test p-value: 3.96720386388631e-17"
[1] "Data is not normally distributed for compactness_mean"
```

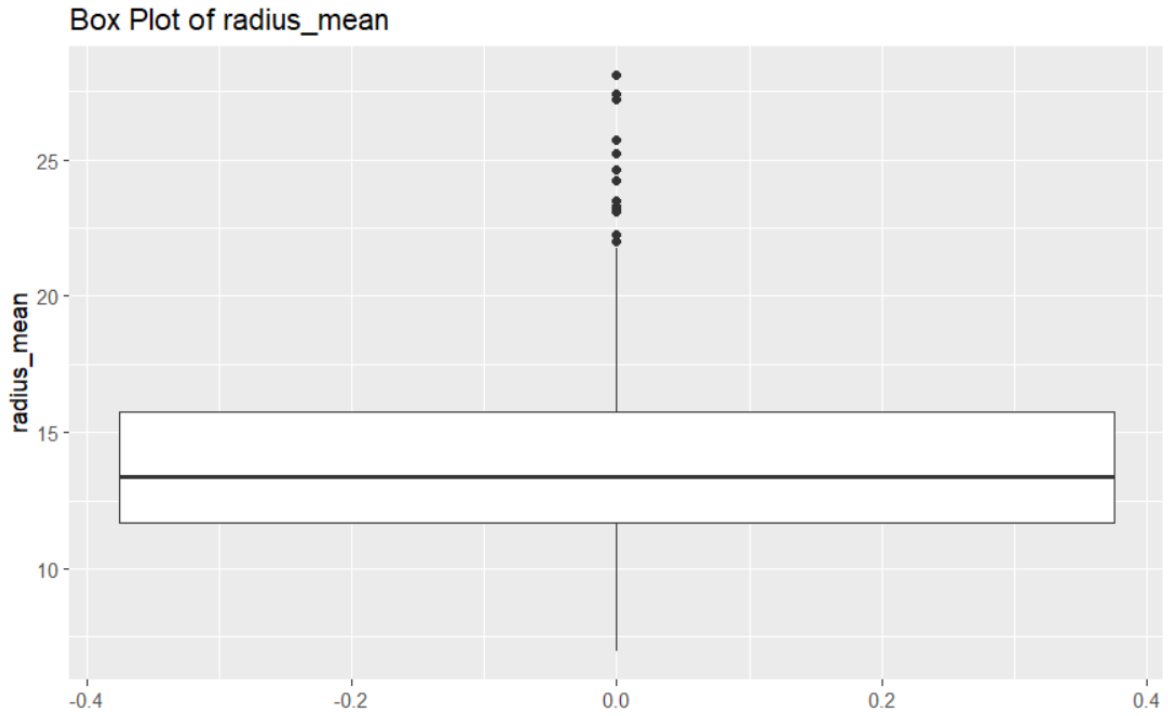
```{r}
# Normality test for symmetry_mean variable using Shapiro-Wilk test
print("Normality test for symmetry_mean:")
shapiro_test_symmetry <- shapiro.test(Breastcancer_data2$symmetry_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_symmetry$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_symmetry$p.value > 0.05) {
  print("Data is normally distributed for symmetry_mean")
} else {
  print("Data is not normally distributed for symmetry_mean")
}
```

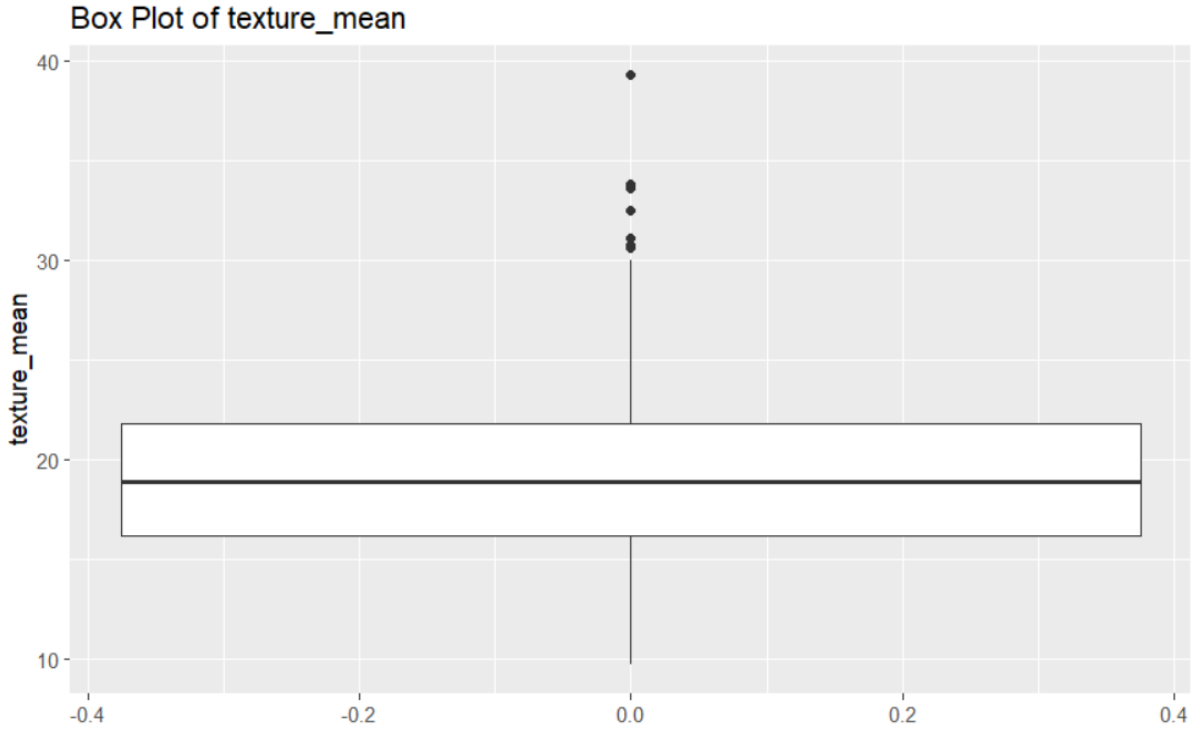
```
[1] "Normality test for symmetry_mean:"
[1] "Shapiro-Wilk test p-value: 7.8847728112006e-09"
[1] "Data is not normally distributed for symmetry_mean"
```

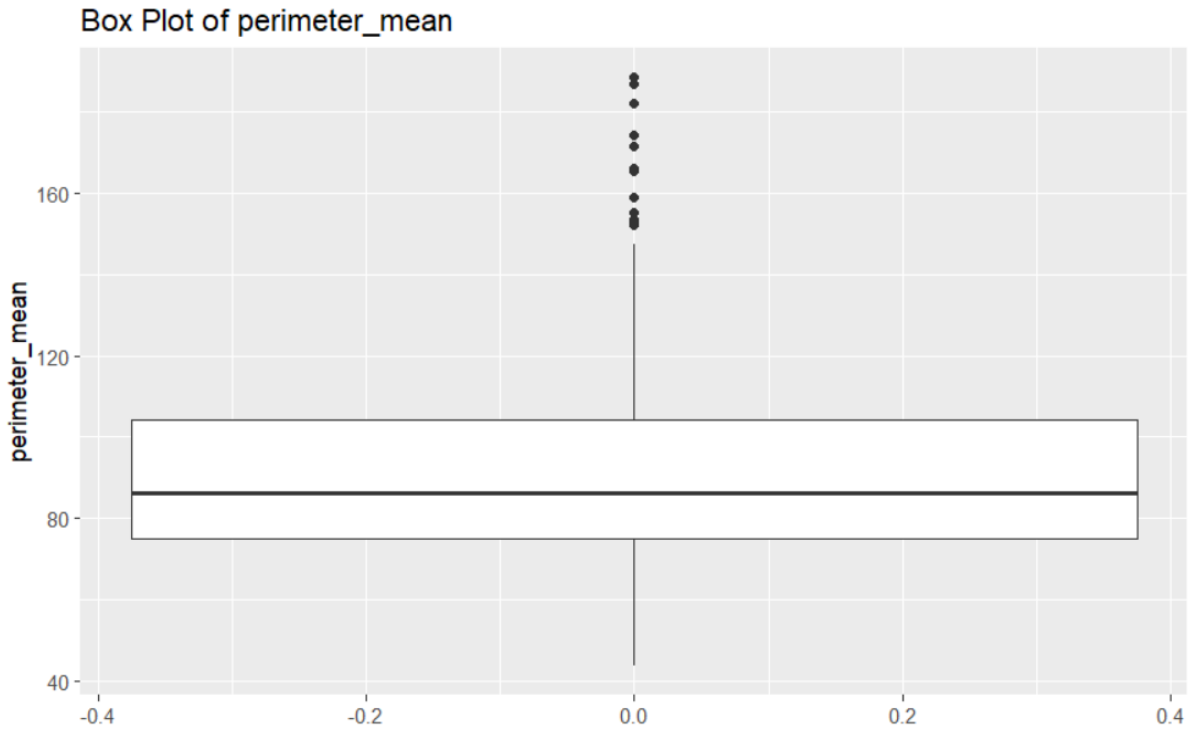
```{r}

# Normality test for fractal_dimension_mean variable using Shapiro-Wilk test
print("Normality test for fractal_dimension_mean:")
shapiro_test_fractal_dimension <- shapiro.test(Breastcancer_data2$fractal_dimension_mean)
print(paste("Shapiro-Wilk test p-value:", shapiro_test_fractal_dimension$p.value))

# Check if data is normal based on Shapiro-Wilk test
if (shapiro_test_fractal_dimension$p.value > 0.05) {
  print("Data is normally distributed for fractal_dimension_mean")
} else {
  print("Data is not normally distributed for fractal_dimension_mean")
}

```

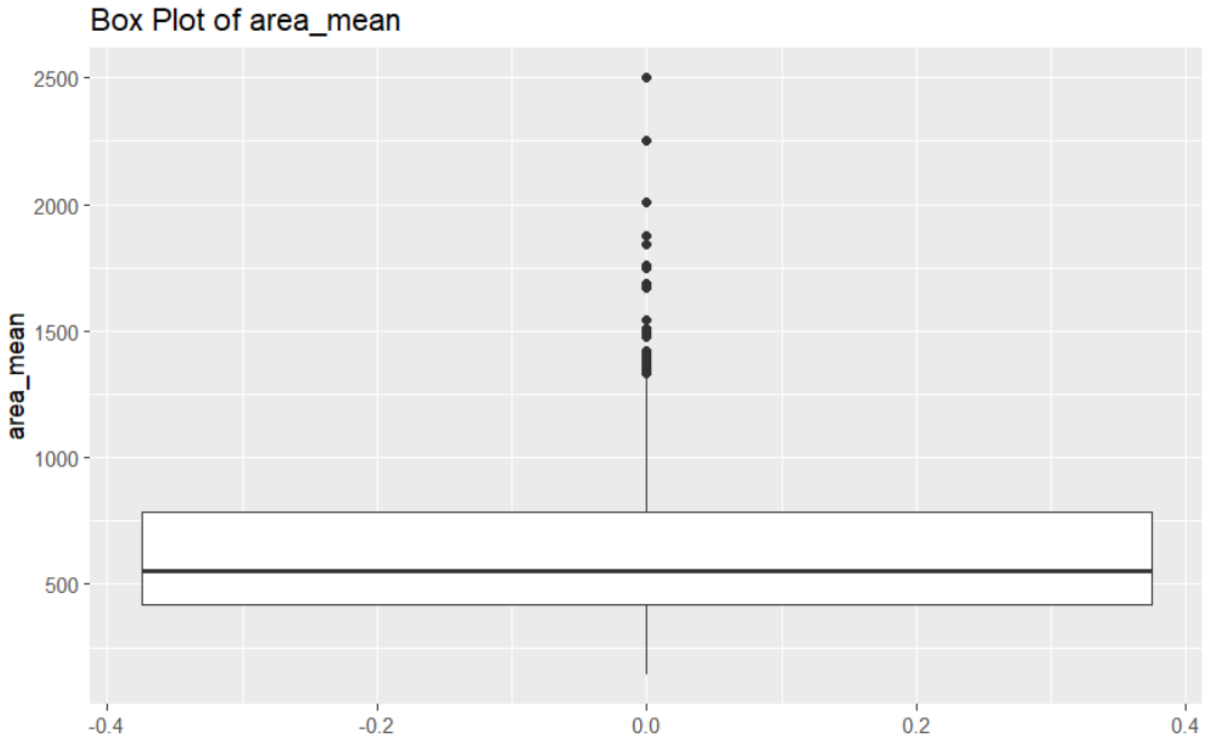
```
[1] "Normality test for fractal_dimension_mean:"
[1] "Shapiro-Wilk test p-value: 1.95657476081236e-16"
[1] "Data is not normally distributed for fractal_dimension_mean"
```

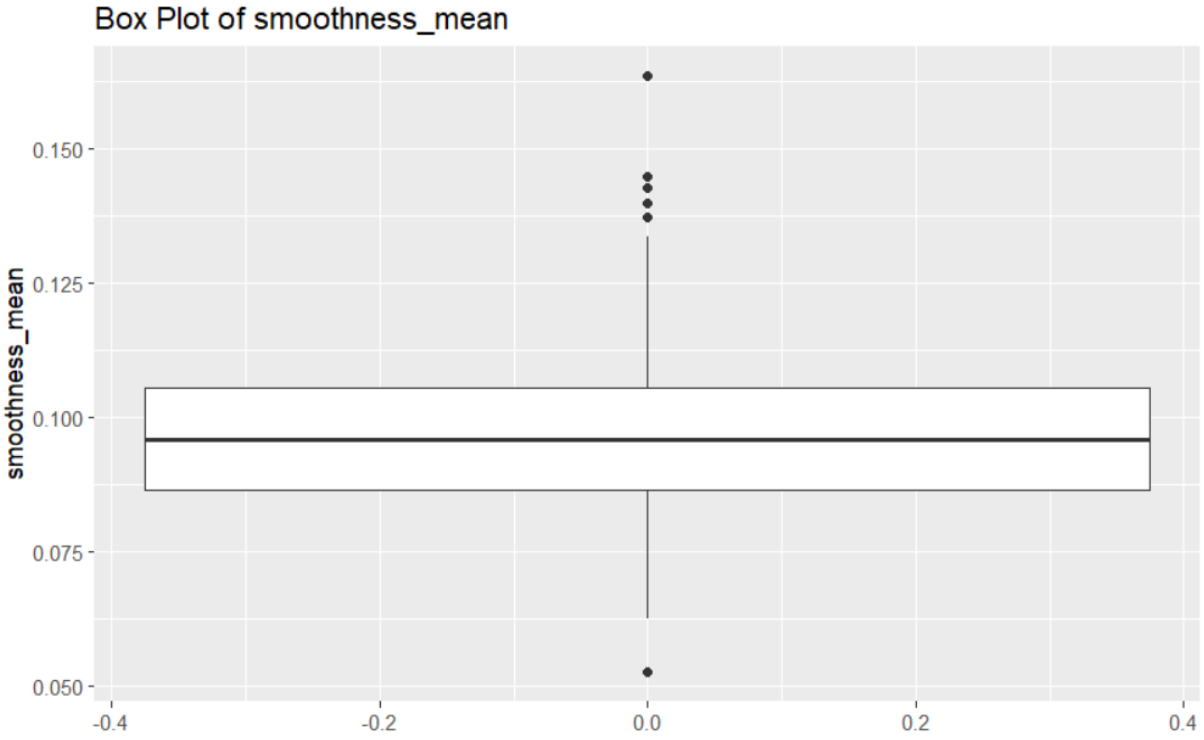
```{r}

# Ensure the ggplot2 package is installed and loaded
if (!require(ggplot2)) install.packages('ggplot2')

# List of variables
variables <- c("radius_mean", "texture_mean", "perimeter_mean", "area_mean",
               "smoothness_mean", "compactness_mean", "symmetry_mean", "fractal_dimension_mean")

# Create box plots for each variable
plots <- lapply(variables, function(var) {
  ggplot(Breastcancer_data2, aes_string(y = var)) +
    geom_boxplot() +
    labs(title = paste("Box Plot of", var), y = var)
})

# Display the plots
plots
```

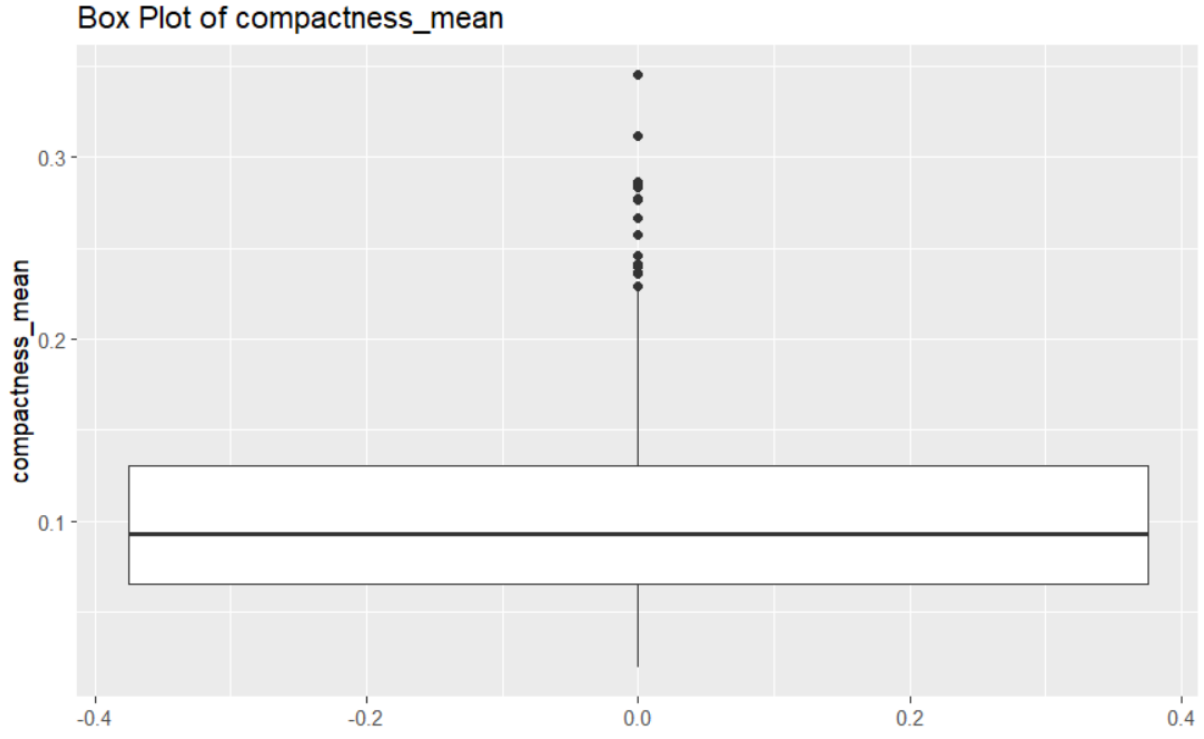

Box Plot of radius_mean
radius_mean
25
20
15
10
-0.4
-0.2
0.0
0.2
0.4

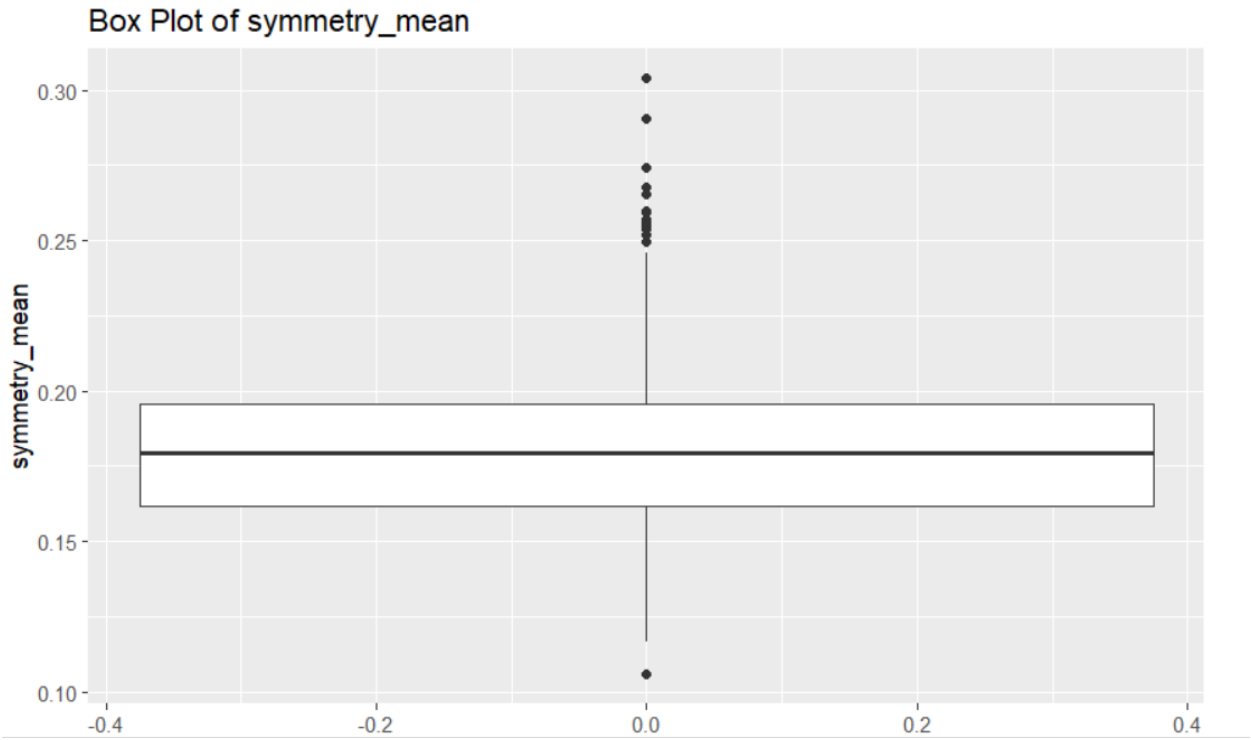

Box Plot of texture_mean
texture_mean
40
30
20
10
-0.4
-0.2
0.0
0.2
0.4

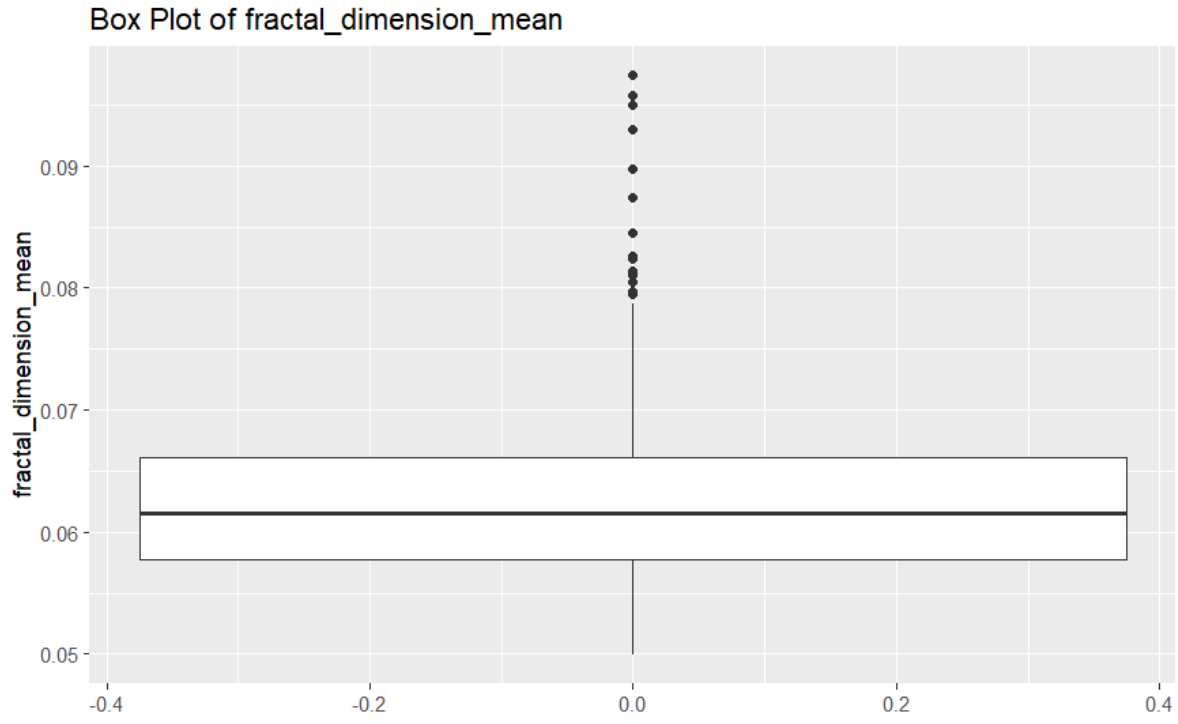

Box Plot of perimeter_mean
perimeter_mean
160
120
80
40
-0.4
-0.2
0.0
0.2
0.4


Box Plot of area_mean
area_mean
2500
2000
1500
1000
500
-0.4
-0.2
0.0
0.2
0.4

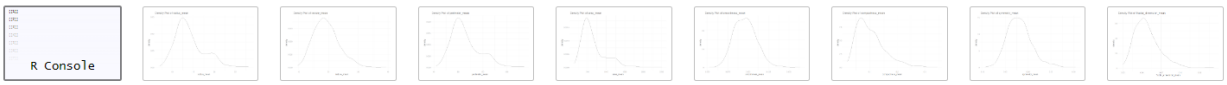

Box Plot of smoothness_mean
smoothness_mean
0.150
0.125
0.100
0.075
0.050
-0.4
-0.2
0.0
0.2
0.4


Box Plot of compactness_mean
compactness_mean
0.3
0.2
0.1
-0.4
-0.2
0.0
0.2
0.4

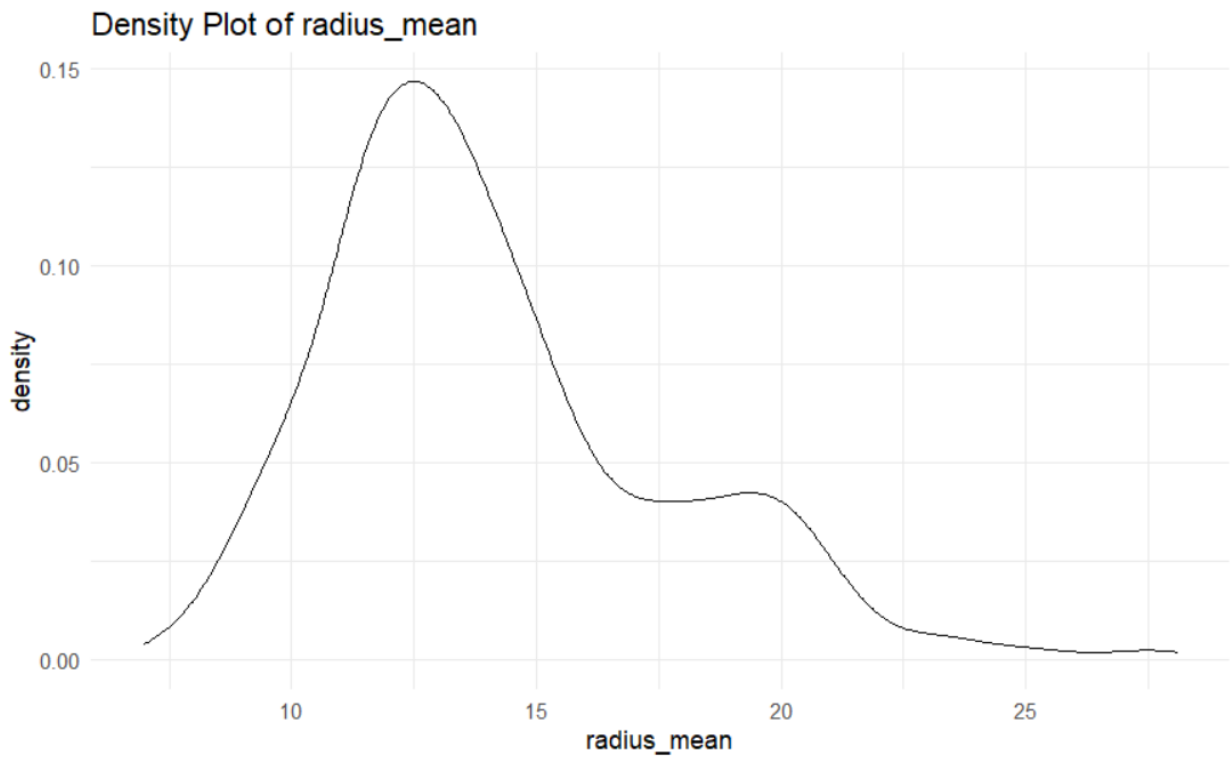

Box Plot of symmetry_mean
symmetry_mean
0.30
0.25
0.20
0.15
0.10
-0.4
-0.2
0.0
0.2
0.4

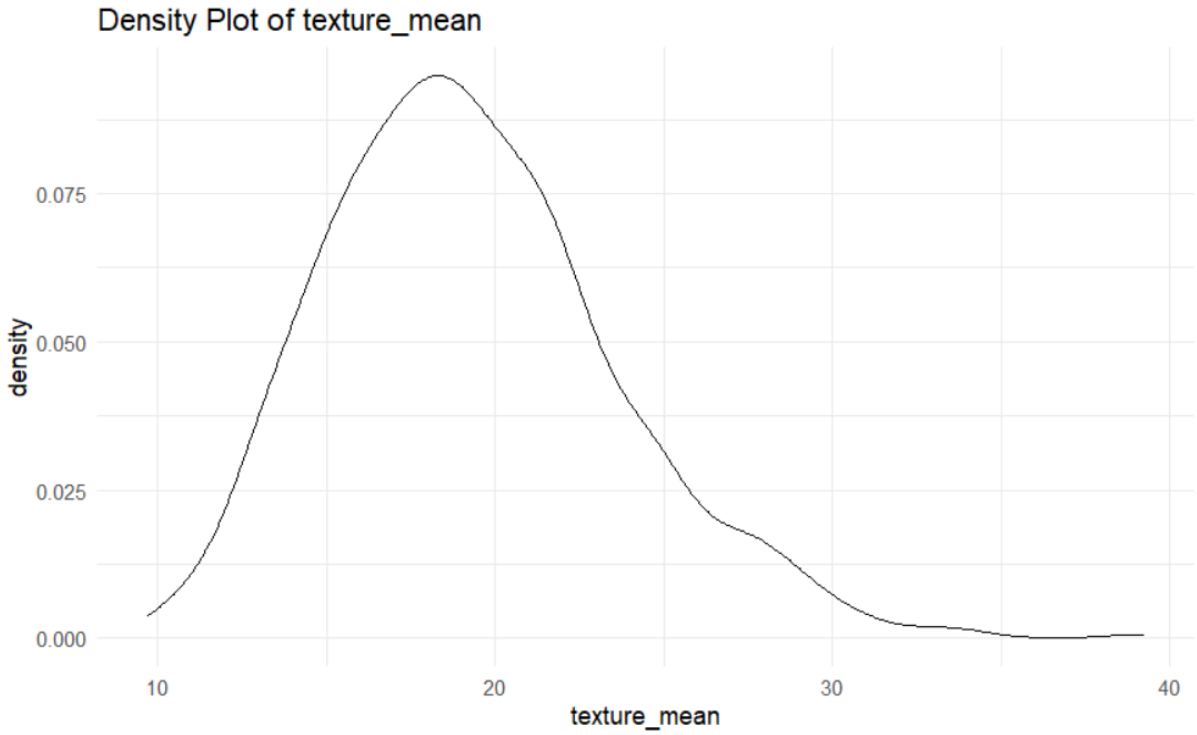

Box Plot of fractal_dimension_mean
fractal_dimension_mean
0.09
0.08
0.07
0.06
0.05
-0.4
-0.2
0.0
0.2
0.4

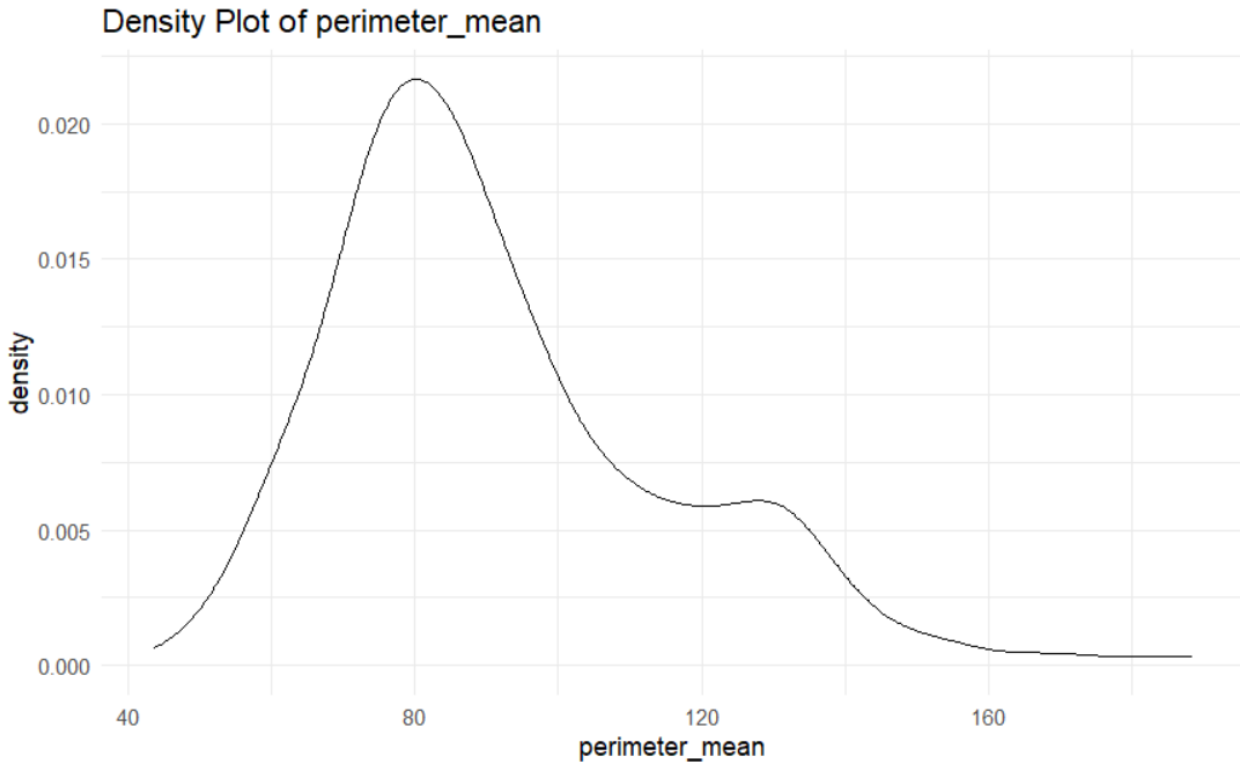
```r
# Ensure the ggplot2 package is installed and loaded
if (!require(ggplot2)) install.packages('ggplot2')

# Function to calculate skewness
skewness <- function(x) {
  mean((x - mean(x)) ^ 3) / (sd(x) ^ 3)
}

# List of variables
variables <- c("radius_mean", "texture_mean", "perimeter_mean", "area_mean",
               "smoothness_mean", "compactness_mean", "symmetry_mean", "fractal_dimension_mean")

# Calculate skewness for each variable
skewness_values <- sapply(Breastcancer_data2[variables], skewness)

# Create density plots for each variable
plots <- lapply(variables, function(var) {
  ggplot(Breastcancer_data2, aes_string(x = var)) +
    geom_density() +
    labs(title = paste("Density Plot of", var), x = var) +
    theme_minimal()
})

# Display the plots
plots

# Print skewness values
print(skewness_values)
```

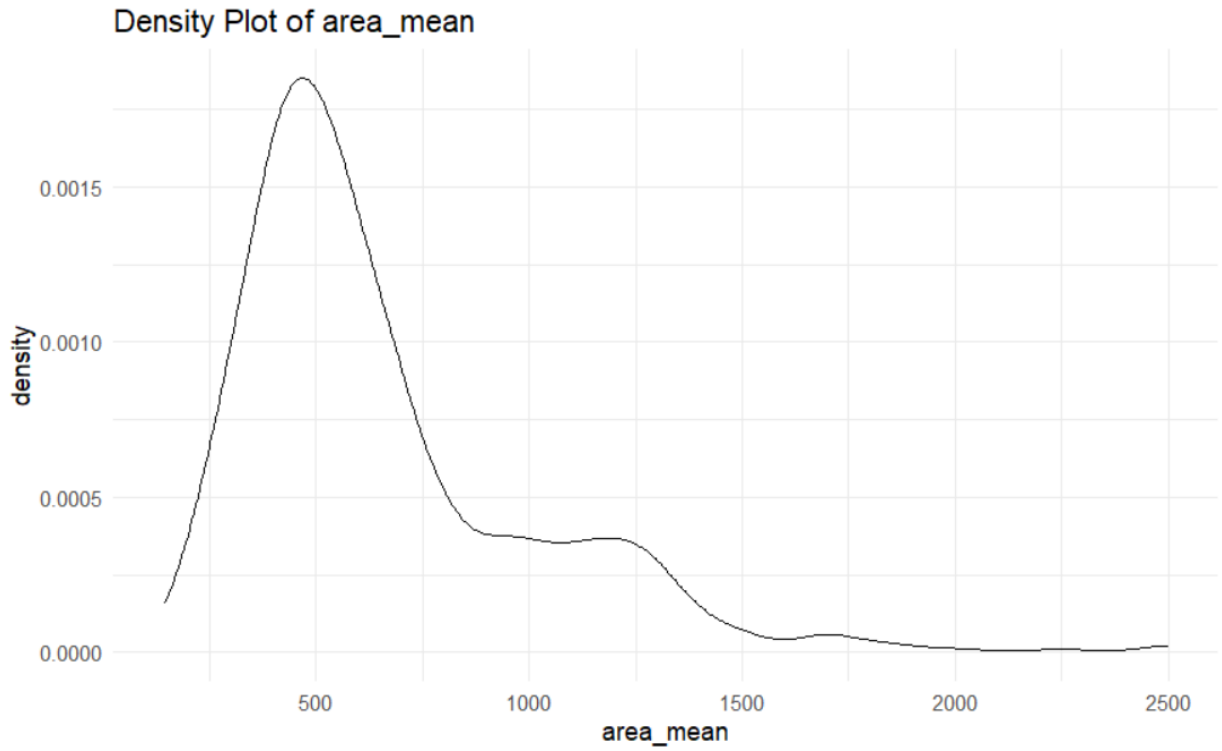


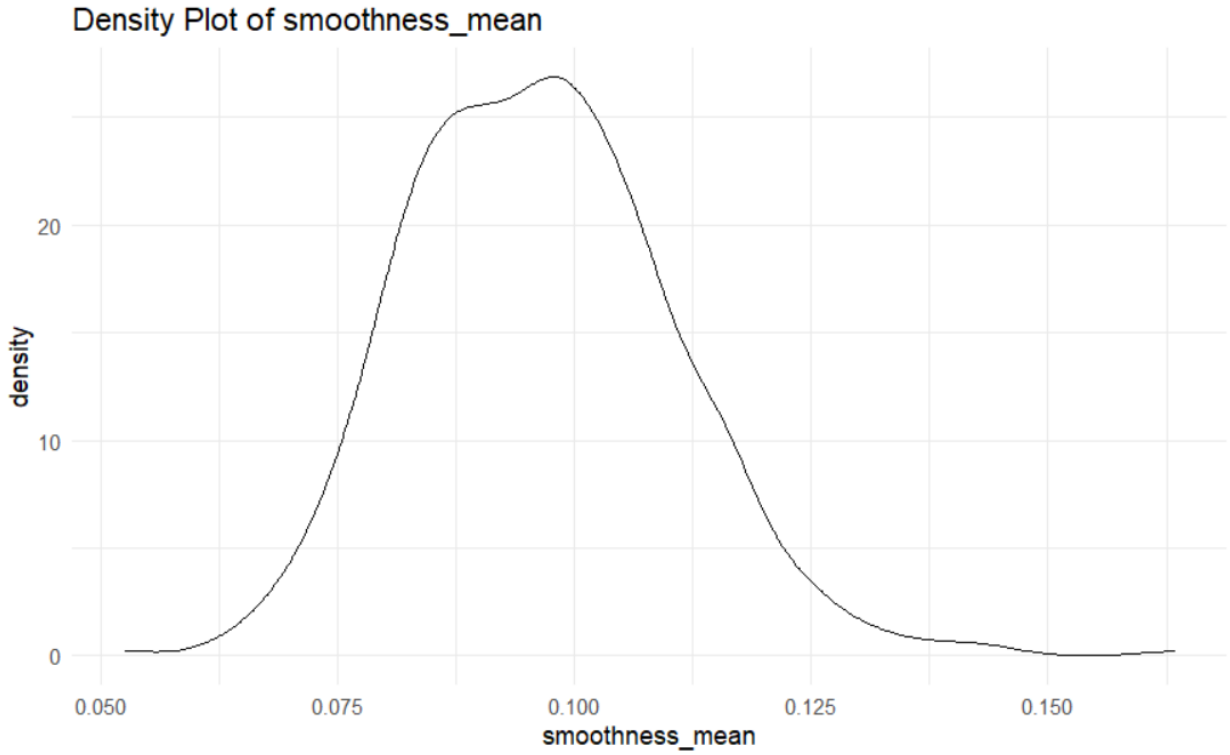

```
[[1]]

[[2]]

[[3]]

[[4]]

[[5]]

[[6]]

[[7]]

[[8]]

          radius_mean           texture_mean         perimeter_mean
area_mean        smoothness_mean       compactness_mean
            0.9374168             0.6470241             0.9854334
1.6370654             0.4539207             1.1838556
         symmetry_mean fractal_dimension_mean
            0.7217877             1.2976191
```

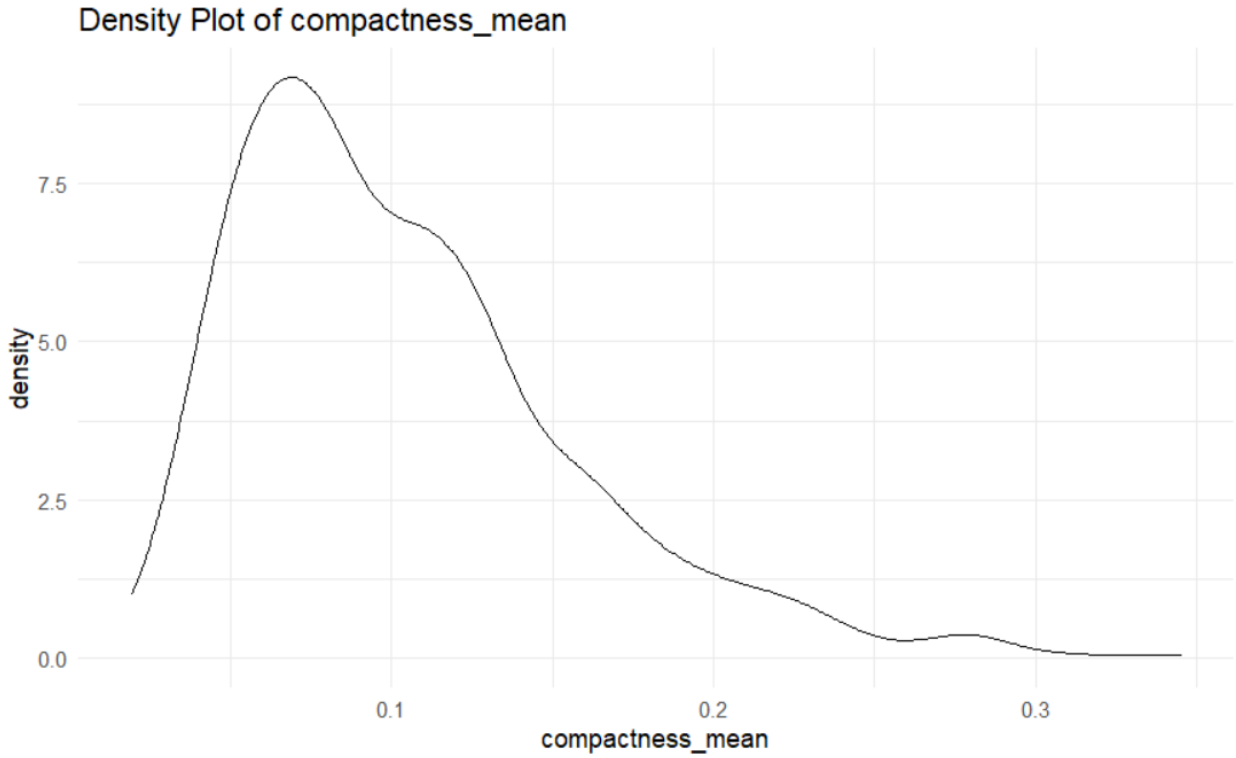


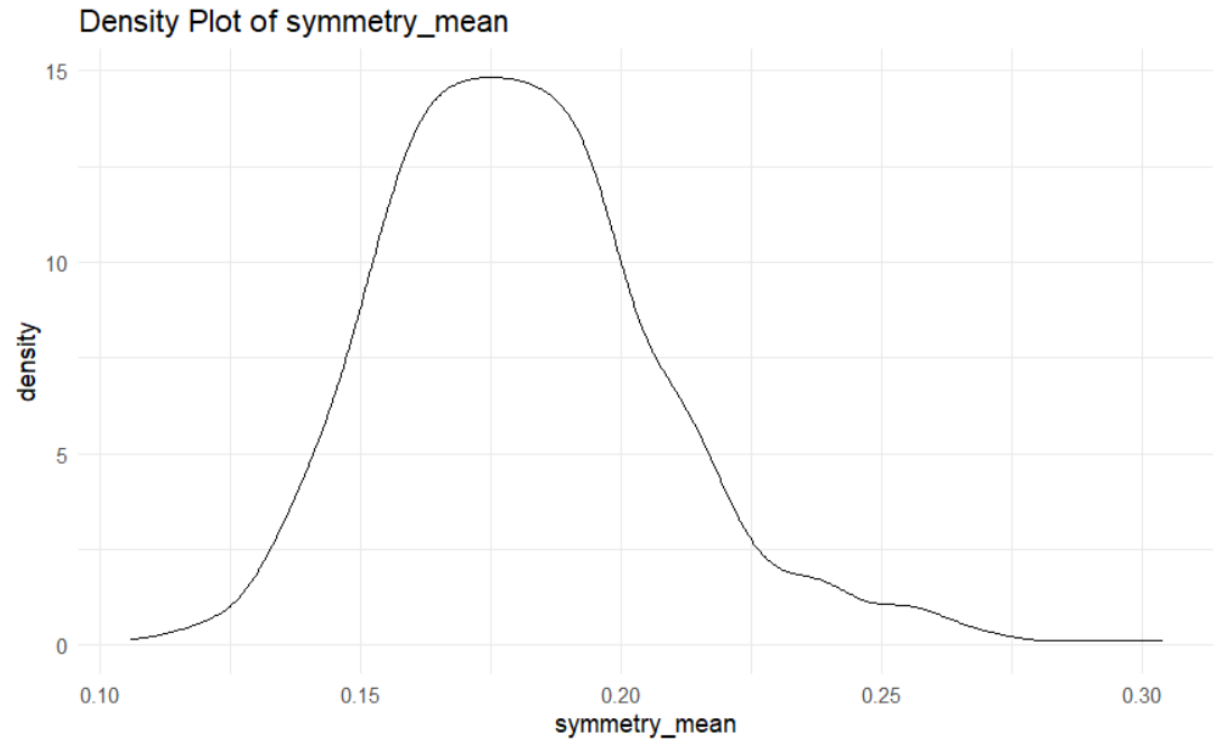

Density Plot of texture_mean
density
0.075
0.050
0.025
0.000
10
20
30
40
texture_mean

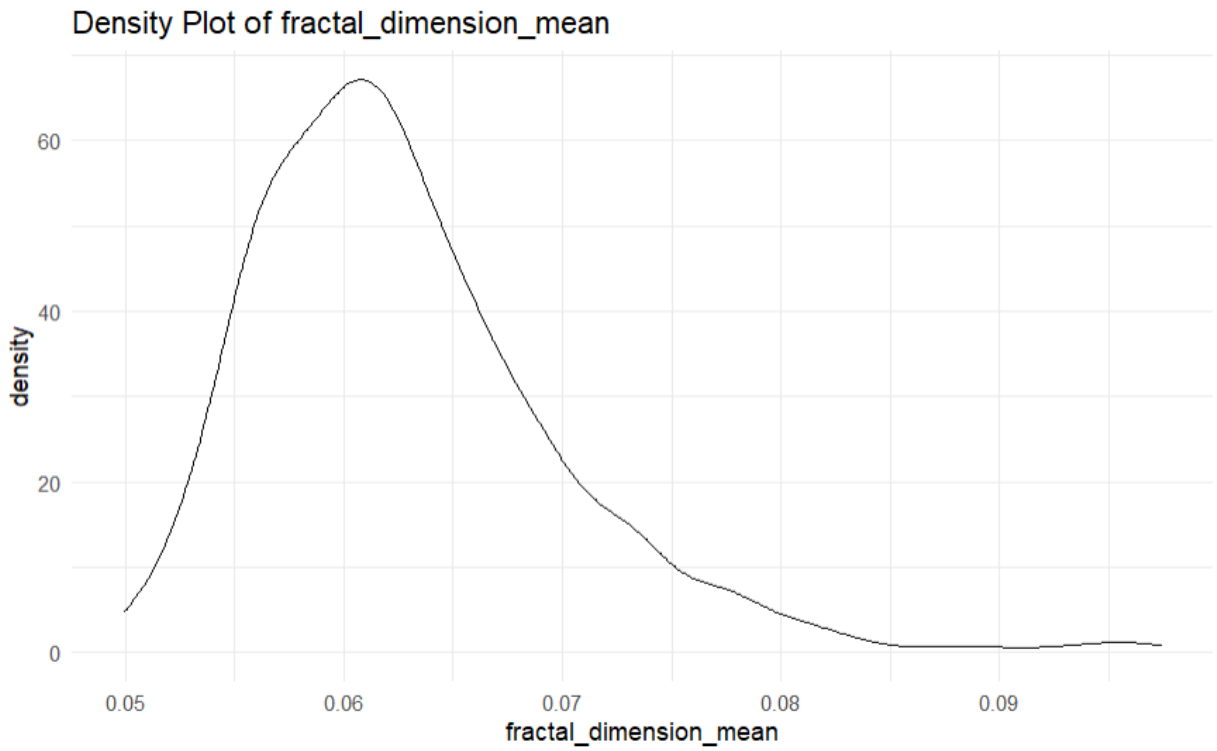

Density Plot of perimeter_mean
density
0.020
0.015
0.010
0.005
0.000
40
80
120
160
perimeter_mean


Density Plot of area_mean
density
0.0015
0.0010
0.0005
0.0000
500
1000
1500
2000
2500
area_mean

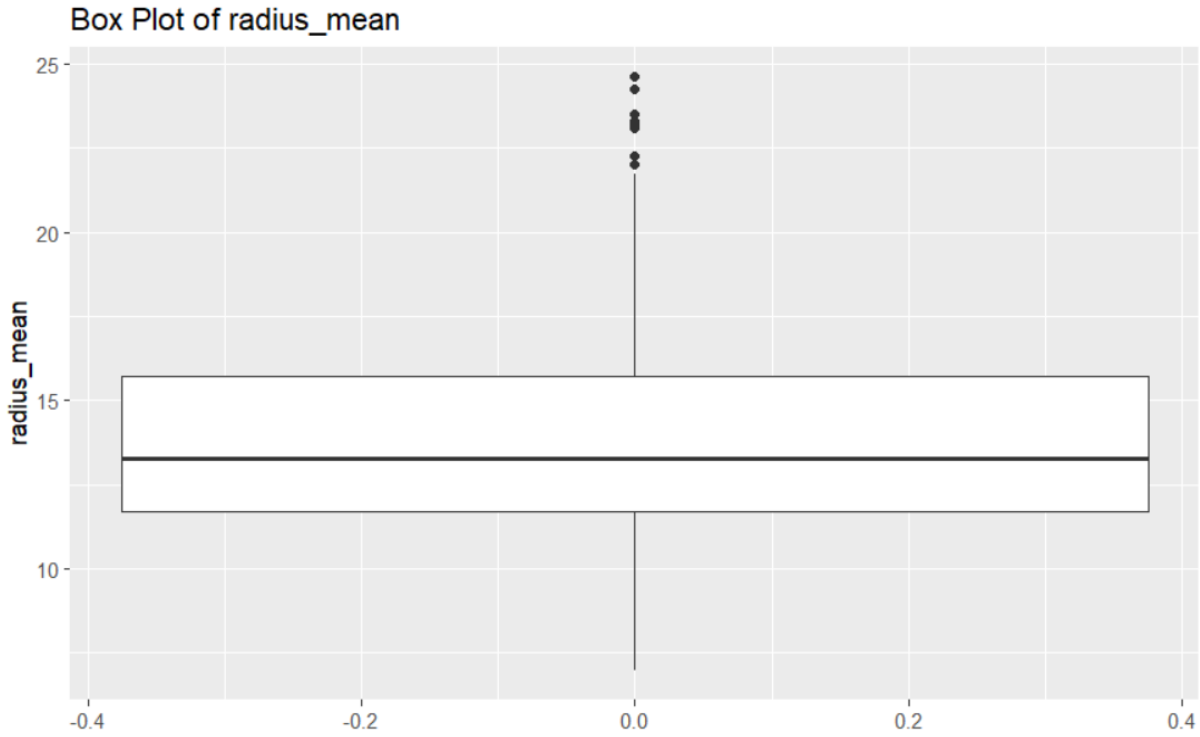

Density Plot of smoothness_mean
density
20
10
0
0.050
0.075
0.100
0.125
0.150
smoothness_mean

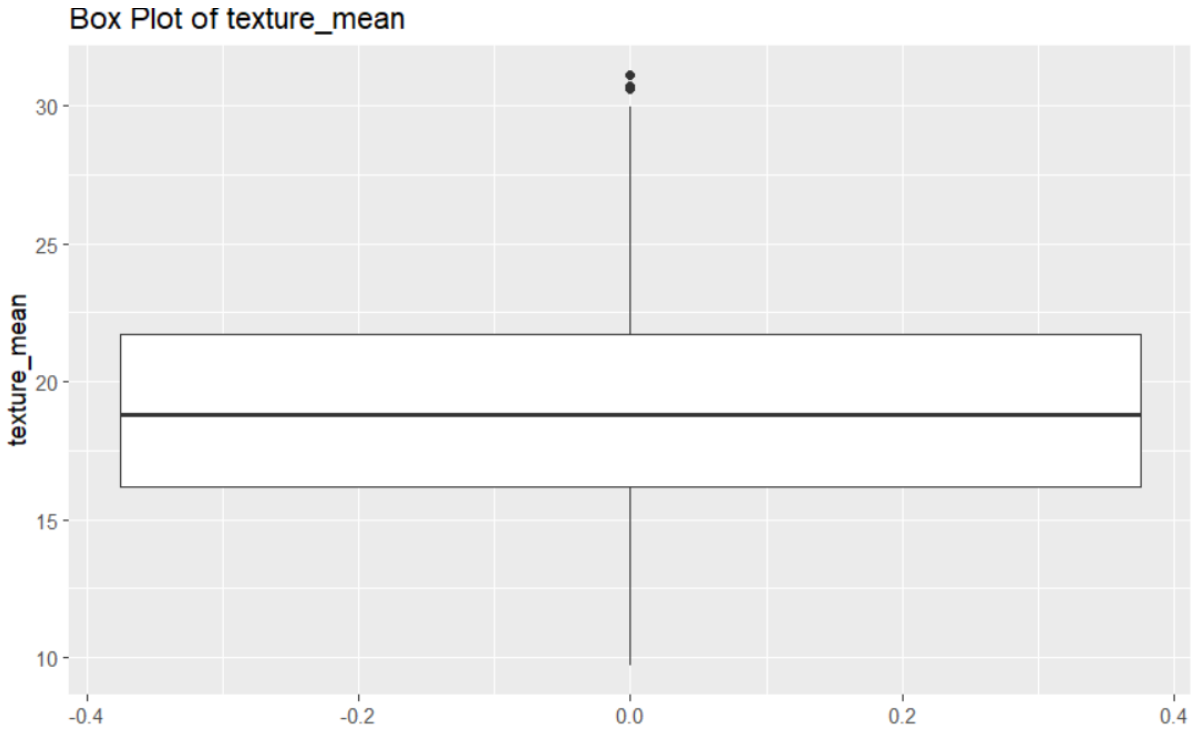

Density Plot of compactness_mean
density
7.5
5.0
2.5
0.0
0.1
0.2
0.3
compactness_mean

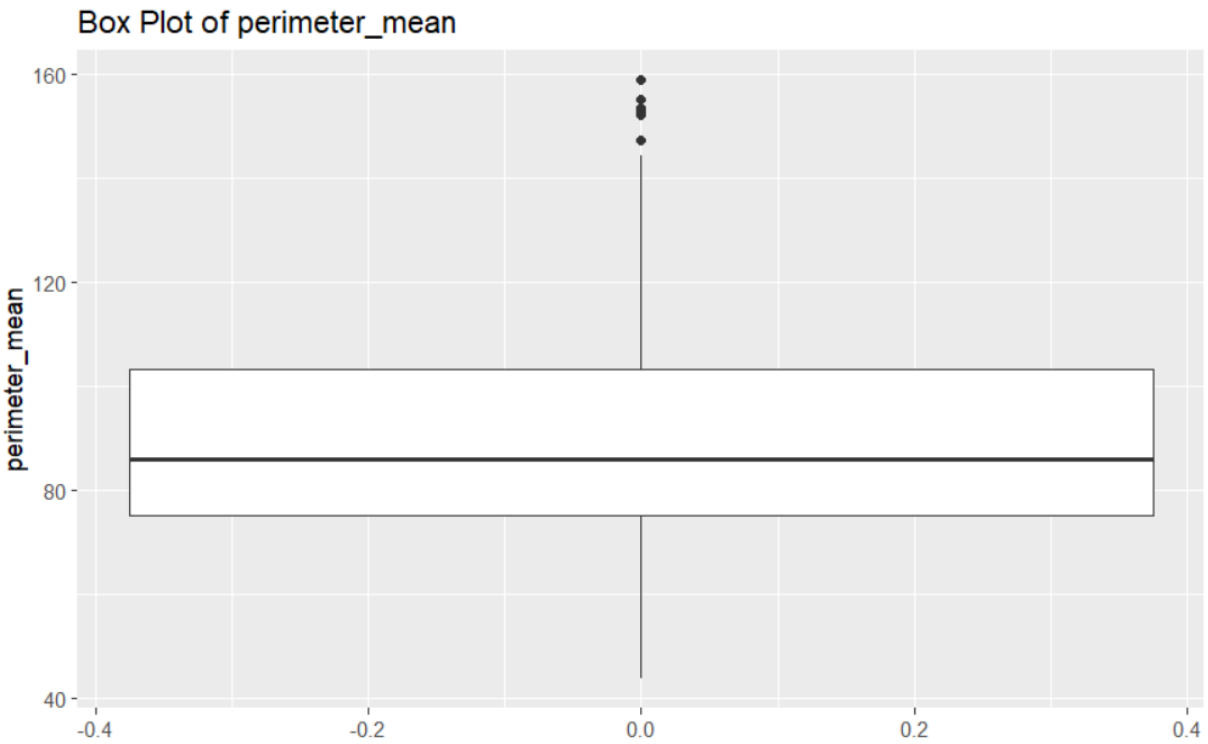

Density Plot of symmetry_mean
density
15
10
5
0
0.10
0.15
0.20
0.25
0.30
symmetry_mean

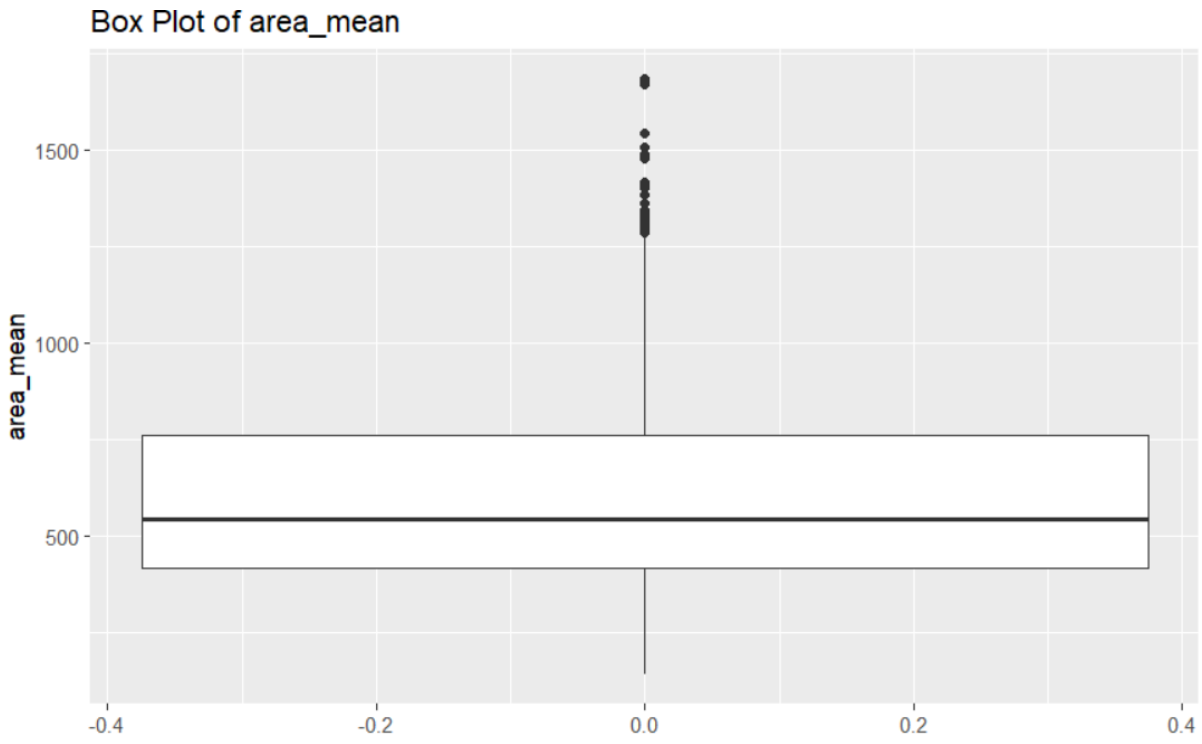


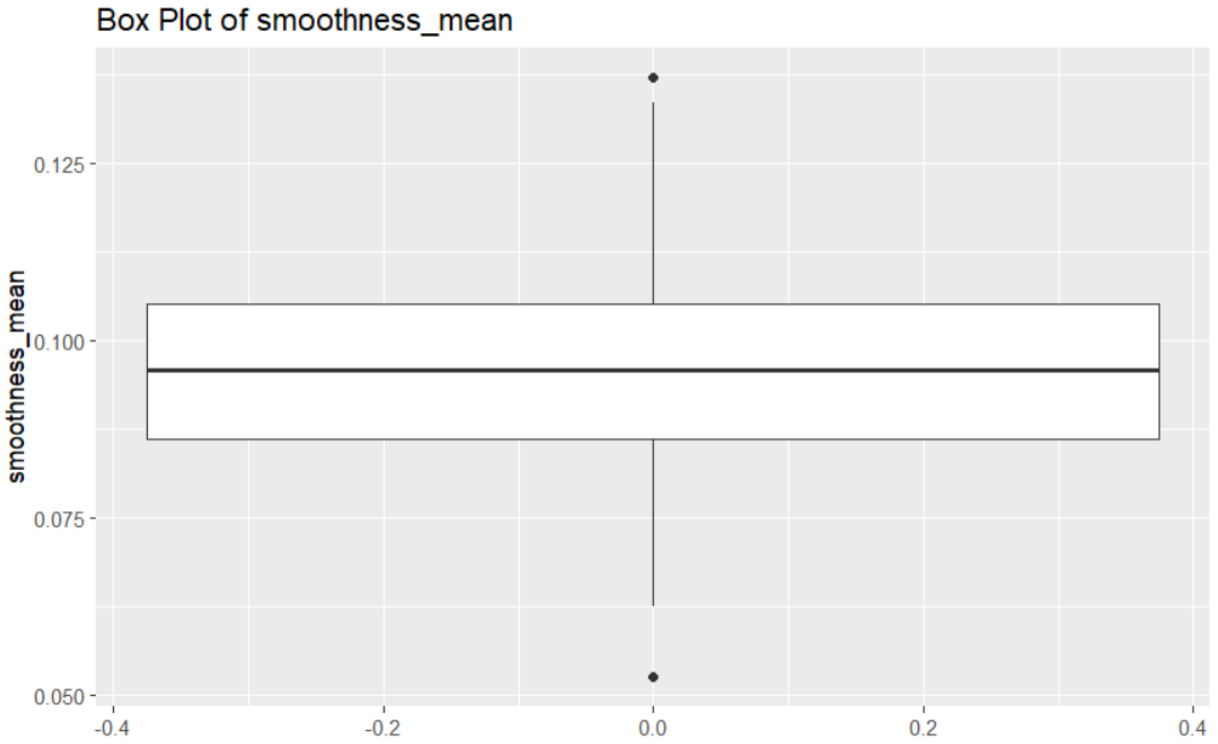

```r
# Ensure the ggplot2 package is installed and loaded
if (!require(ggplot2)) install.packages('ggplot2')

# List of variables
variables <- c("radius_mean", "texture_mean", "perimeter_mean", "area_mean",
               "smoothness_mean", "compactness_mean", "symmetry_mean", "fractal_dimension_mean")

# Outlier treatment function
treat_outliers <- function(x) {
  mean_x <- mean(x, na.rm = TRUE)
  sd_x <- sd(x, na.rm = TRUE)
  threshold <- mean_x + 3 * sd_x
  x[x > threshold] <- NA
  return(x)
}

# Apply outlier treatment and create box plots for each variable
plots <- lapply(variables, function(var) {
  Breastcancer_data2[[var]] <- treat_outliers(Breastcancer_data2[[var]]) # Apply outlier treatment
  ggplot(Breastcancer_data2, aes_string(y = var)) +
    geom_boxplot() +
    labs(title = paste("Box Plot of", var), y = var)
})

# Display the plots
plots
```

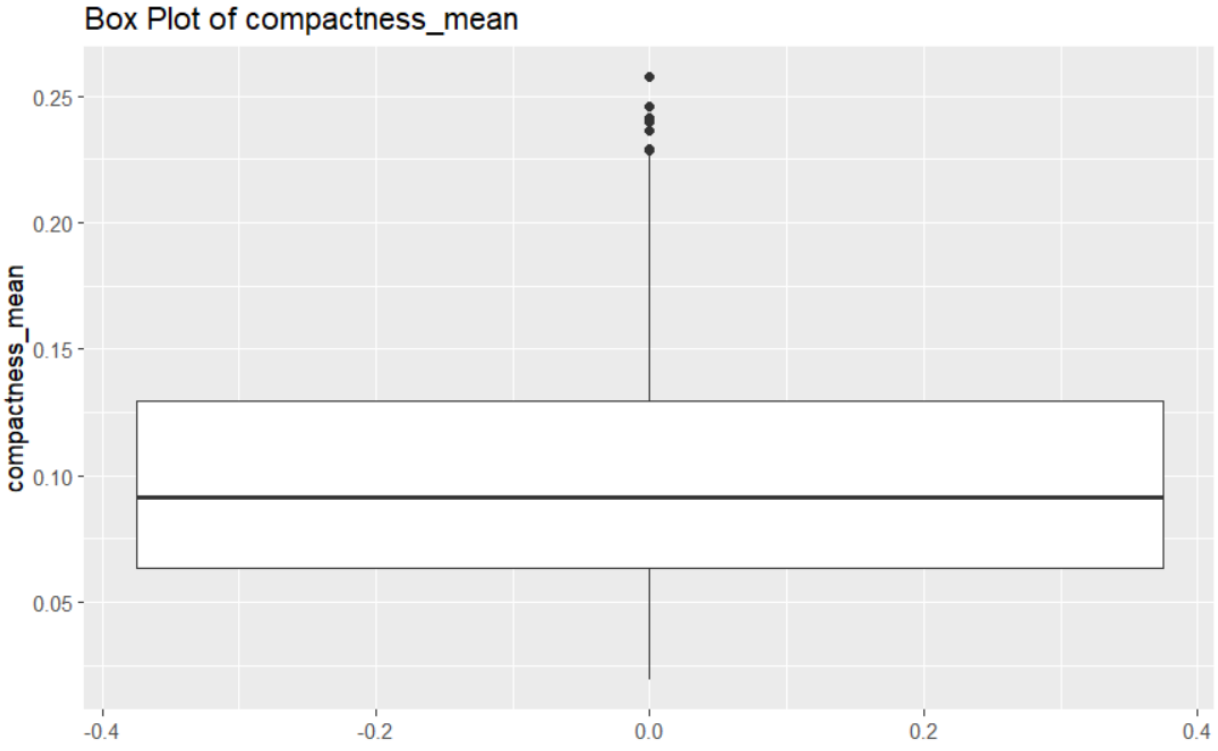

Box Plot of radius_mean
radius_mean
25
20
15
10
-0.4
-0.2
0.0
0.2
0.4

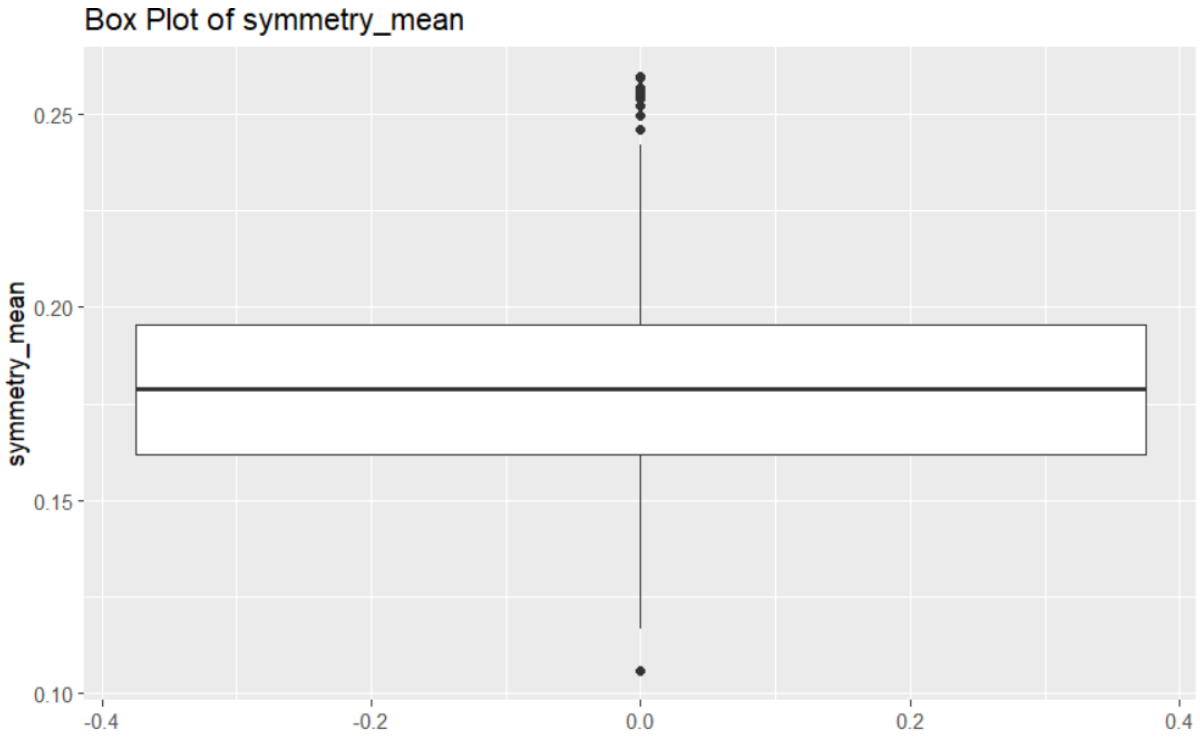

Box Plot of texture_mean
texture_mean
30
25
20
15
10
-0.4
-0.2
0.0
0.2
0.4

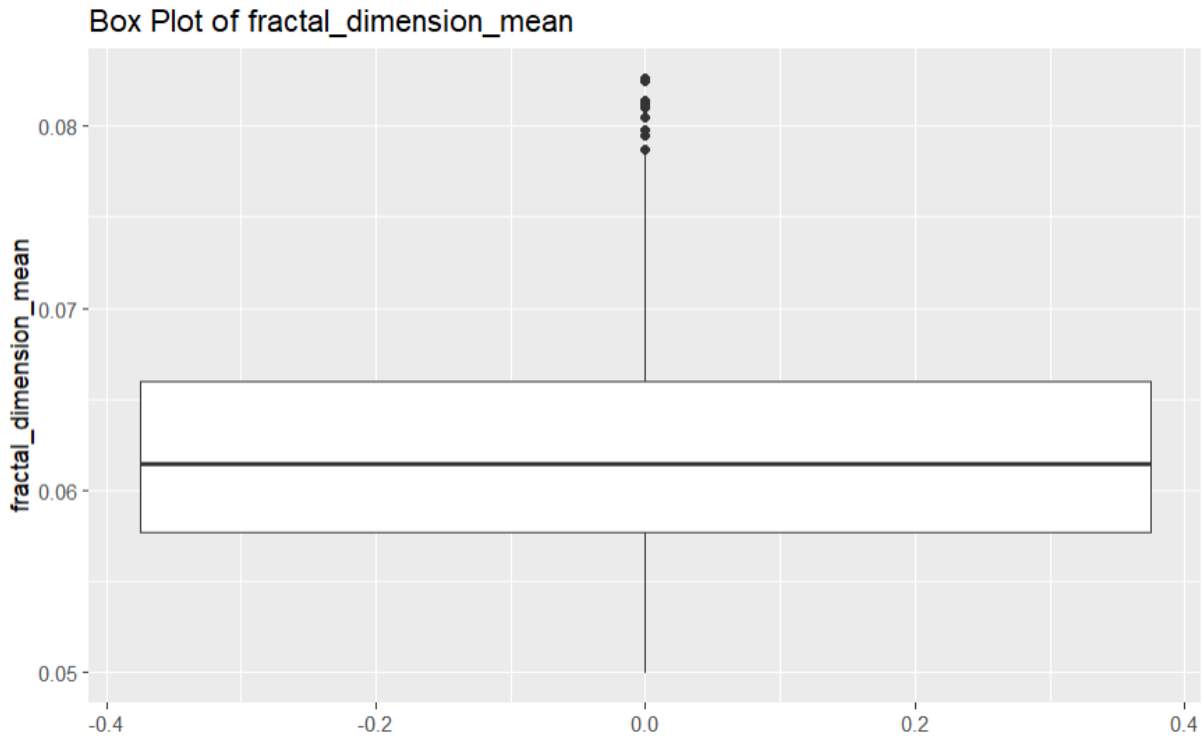

Box Plot of perimeter_mean
perimeter_mean
160
120
80
40
-0.4
-0.2
0.0
0.2
0.4


Box Plot of area_mean
area_mean
1500
1000
500
-0.4
-0.2
0.0
0.2
0.4


Box Plot of smoothness_mean
smoothness_mean
0.125
0.100
0.075
0.050
-0.4
-0.2
0.0
0.2
0.4


Box Plot of compactness_mean
compactness_mean
0.25
0.20
0.15
0.10
0.05
-0.4
-0.2
0.0
0.2
0.4


Box Plot of symmetry_mean
symmetry_mean
0.25
0.20
0.15
0.10
-0.4
-0.2
0.0
0.2
0.4


Box Plot of fractal_dimension_mean
fractal_dimension_mean
0.08
0.07
0.06
0.05
-0.4
-0.2
0.0
0.2
0.4

```{r}
## Calculate the quartiles of the radius_mean column
quartiles <- quantile(Breastcancer_data2$radius_mean, probs = c(0, 0.25, 0.5, 0.75, 1))

# Create a new column with the tumor size categories based on the quartile ranges
Breastcancer_data2$tumor_size <- cut(Breastcancer_data2$radius_mean,
                                     breaks = quartiles,
                                     labels = c("Very Small Tumors", "Small Tumors", "Medium Tumors", "Large Tumors"),
                                     include.lowest = TRUE)

# Print the updated data frame
head(Breastcancer_data2)

```

Description: df [6 × 11]

	id <int>	diagnosis <chr>	radius_mean <dbl>	texture_mean <dbl>	perimeter_mean <dbl>	area_mean <dbl>	smoothness_mean <dbl>	compactness_mean <dbl>	symmetry_mean <dbl>
1	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.2419
2	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.1812
3	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.2069
4	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2597
5	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1809
6	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.2087

6 rows | 1-10 of 11 columns

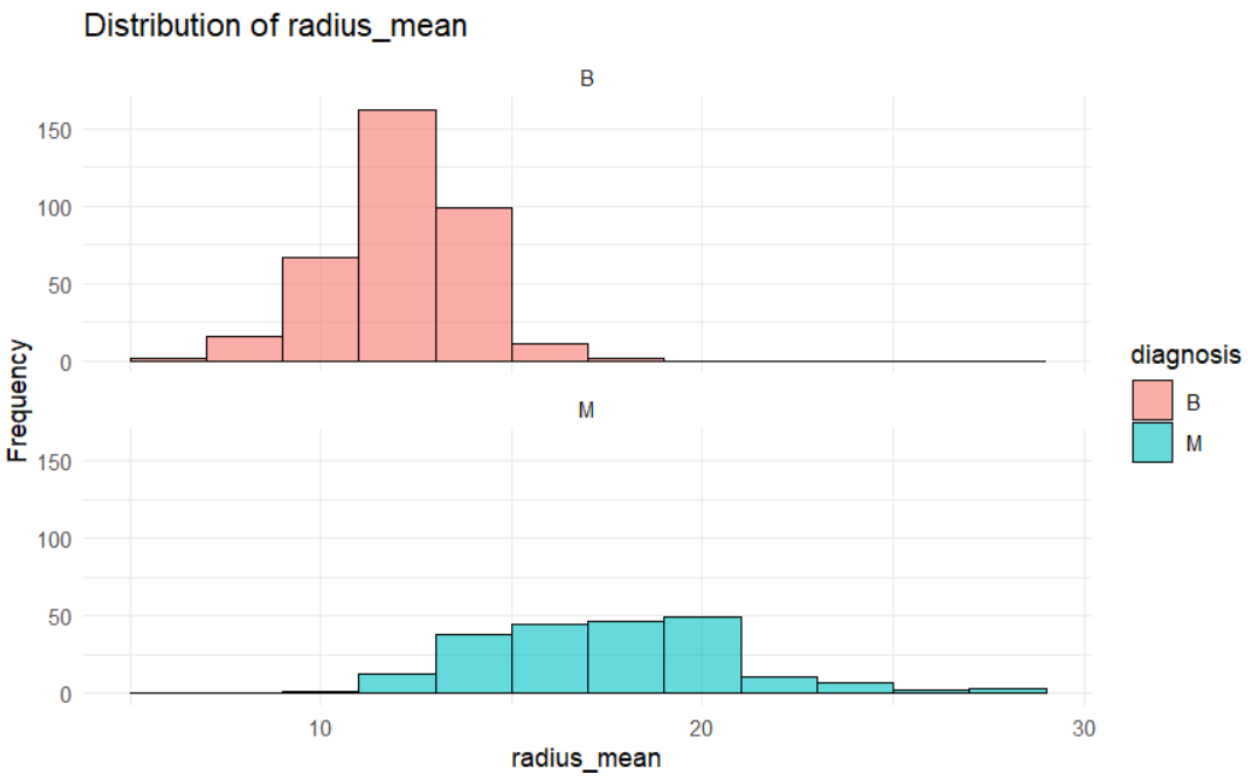
```{r}
# Calculate the quartiles of the radius_mean column
quartiles <- quantile(Breastcancer_data2$radius_mean, probs = c(0, 0.25, 0.5, 0.75, 1))

# Create a new column with the tumor size categories based on the quartile ranges
Breastcancer_data2$tumor_size <- cut(Breastcancer_data2$radius_mean,
                                     breaks = quartiles,
                                     labels = c("Very Small Tumors", "Small Tumors", "Medium Tumors", "Large Tumors"),
                                     include.lowest = TRUE)

# Create a new column with numerical values corresponding to the categories in the existing "tumor_size" column
Breastcancer_data2$tumor_size_numerical <- as.integer(as.factor(Breastcancer_data2$tumor_size))

# Print the updated data frame
head(Breastcancer_data2)

```

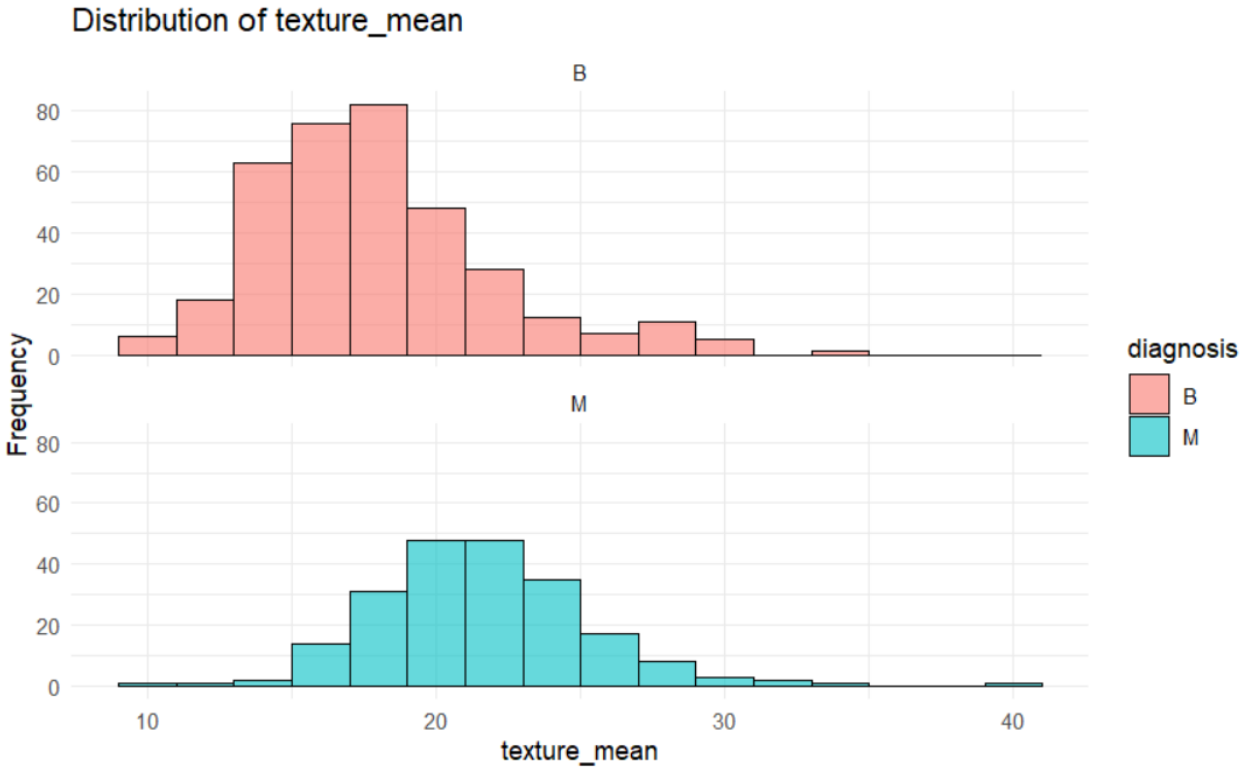
Description: df [6 × 12]

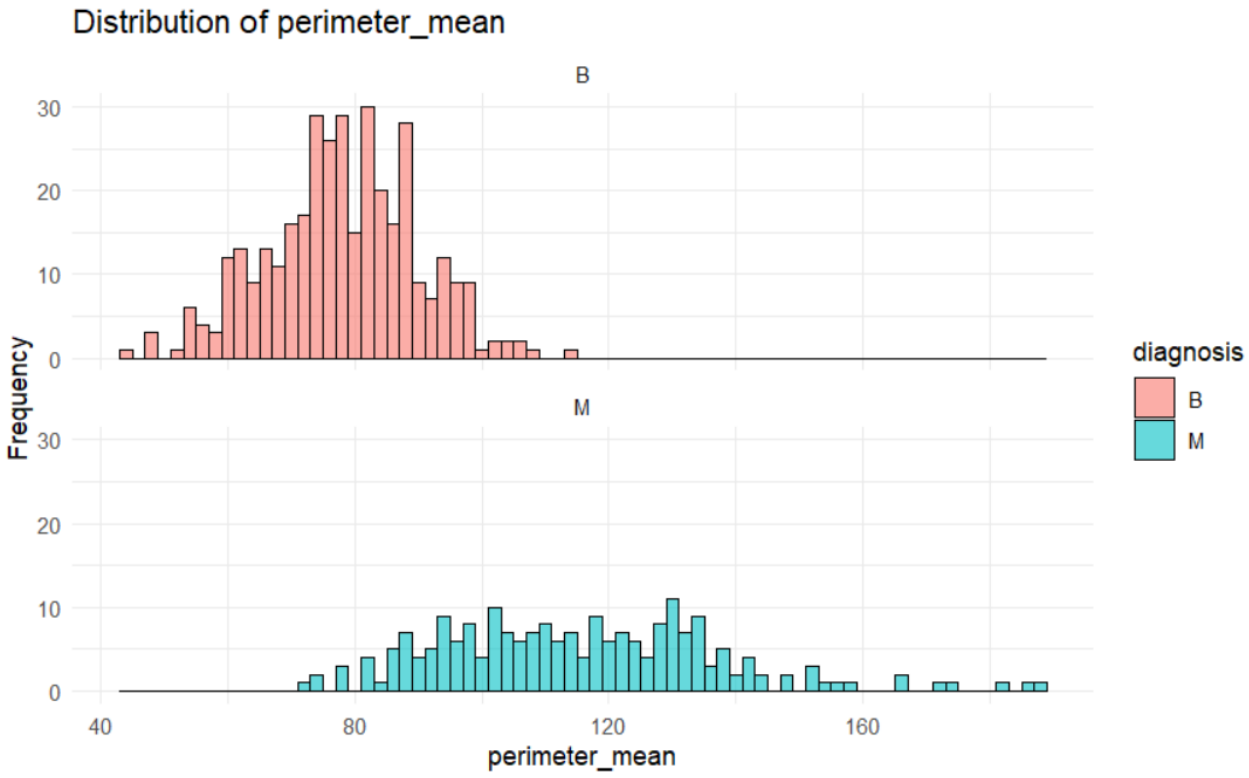
	id <int>	diagnosis <chr>	radius_mean <dbl>	texture_mean <dbl>	perimeter_mean <dbl>	area_mean <dbl>	smoothness_mean <dbl>	compactness_mean <dbl>	symmetry_mean <dbl>
1	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.2419
2	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.1812
3	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.2069
4	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2597
5	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1809
6	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.2087

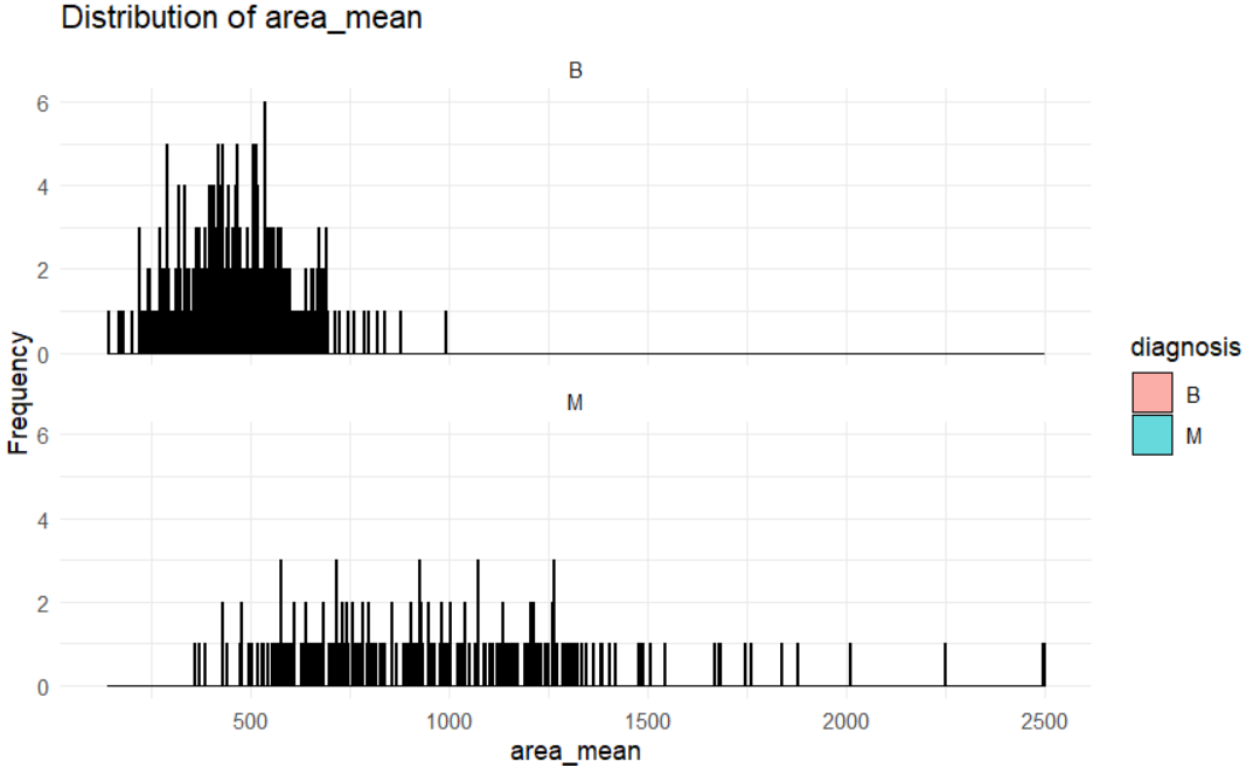
6 rows | 1-10 of 12 columns

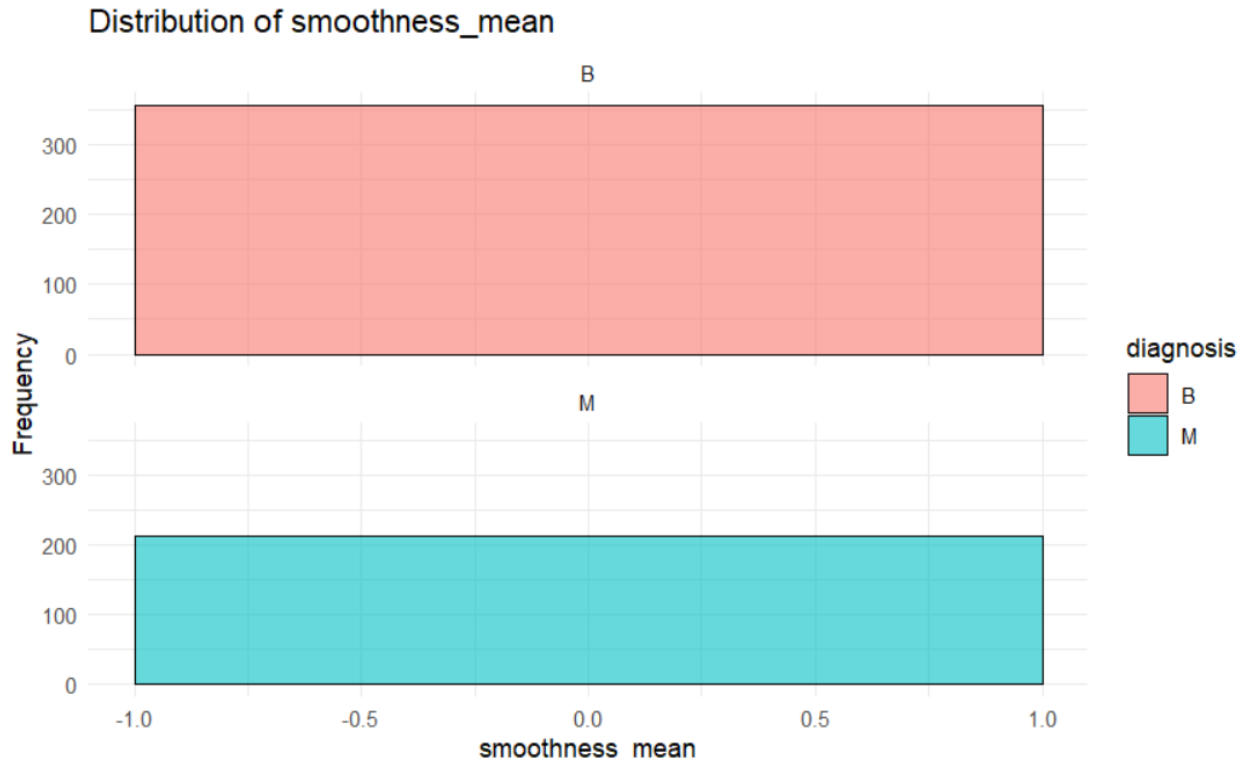
```{r}
# Load necessary libraries
library(ggplot2)

# Assuming the diagnosis variable is named "diagnosis" in your dataset

# Create histograms for all variables grouped by diagnosis
for (var in c("radius_mean", "texture_mean", "perimeter_mean", "area_mean",
              "smoothness_mean", "compactness_mean", "symmetry_mean")) {

  # Create histogram
  p <- ggplot(Breastcancer_data2, aes_string(x = var, fill = "diagnosis")) +
    geom_histogram(binwidth = 2, color = "black", alpha = 0.6) +
    labs(x = var, y = "Frequency", title = paste("Distribution of", var)) +
    theme_minimal() +
    facet_wrap(~ diagnosis, ncol = 1)

  # Print histogram
  print(p)
}


```

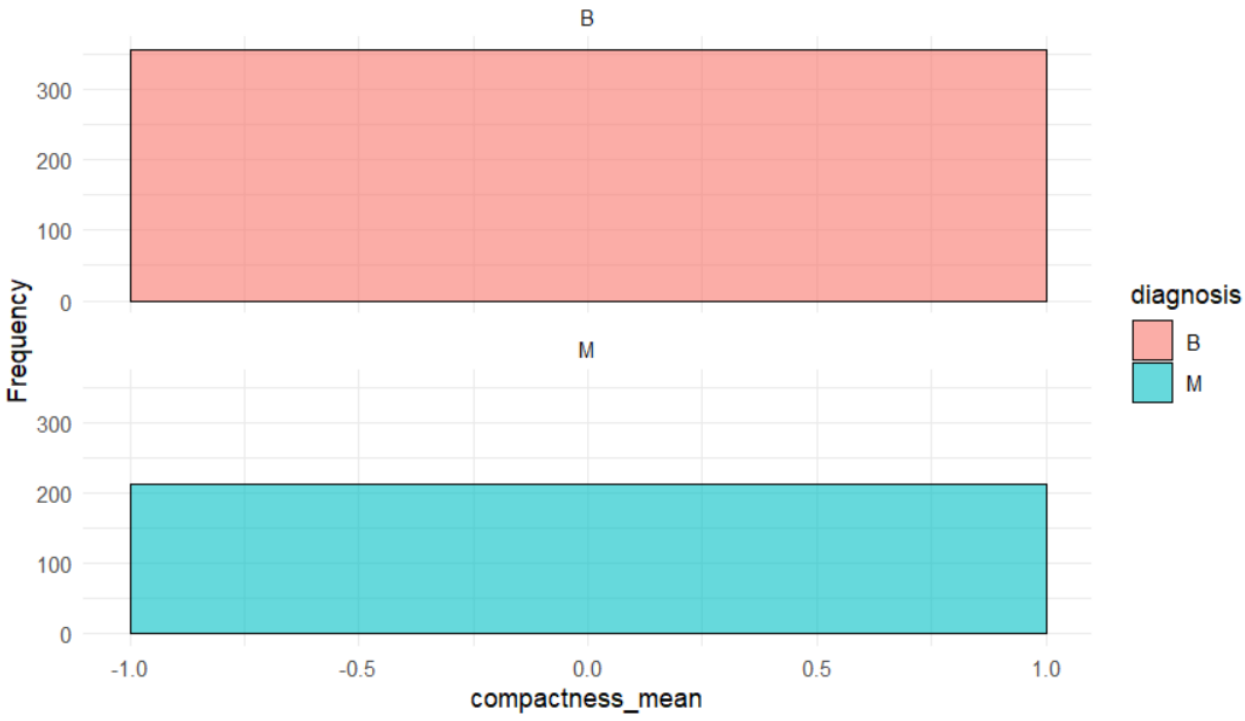

Distribution of radius_mean
B
M
Frequency
150
100
50
0
10
20
30
radius_mean
diagnosis
B
M

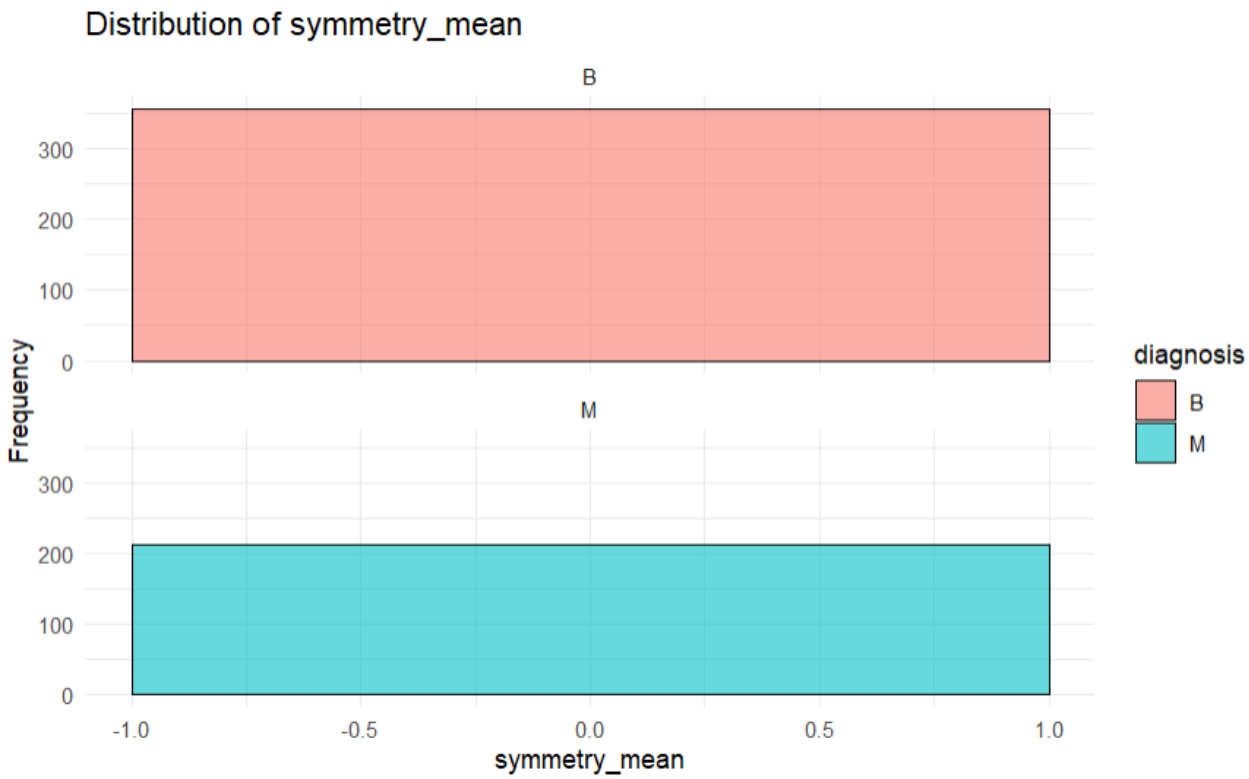

Distribution of texture_mean
B
M
Frequency
80
60
40
20
0
10
20
30
40
texture_mean
diagnosis
B
M


Distribution of perimeter_mean
B
M
Frequency
30
20
10
0
40
80
120
160
perimeter_mean
diagnosis
B
M

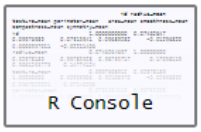

Distribution of area_mean
B
M
Frequency
6
4
2
0
500
1000
1500
2000
2500
area_mean
diagnosis
B
M


Distribution of smoothness_mean
B
M
Frequency
300
200
100
0
-1.0
-0.5
0.0
0.5
1.0
smoothness_mean
diagnosis
B
M

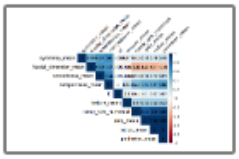

Distribution of compactness_mean
B
M
Frequency
300
200
100
0
-1.0
-0.5
0.0
0.5
1.0
compactness_mean
diagnosis
B
M

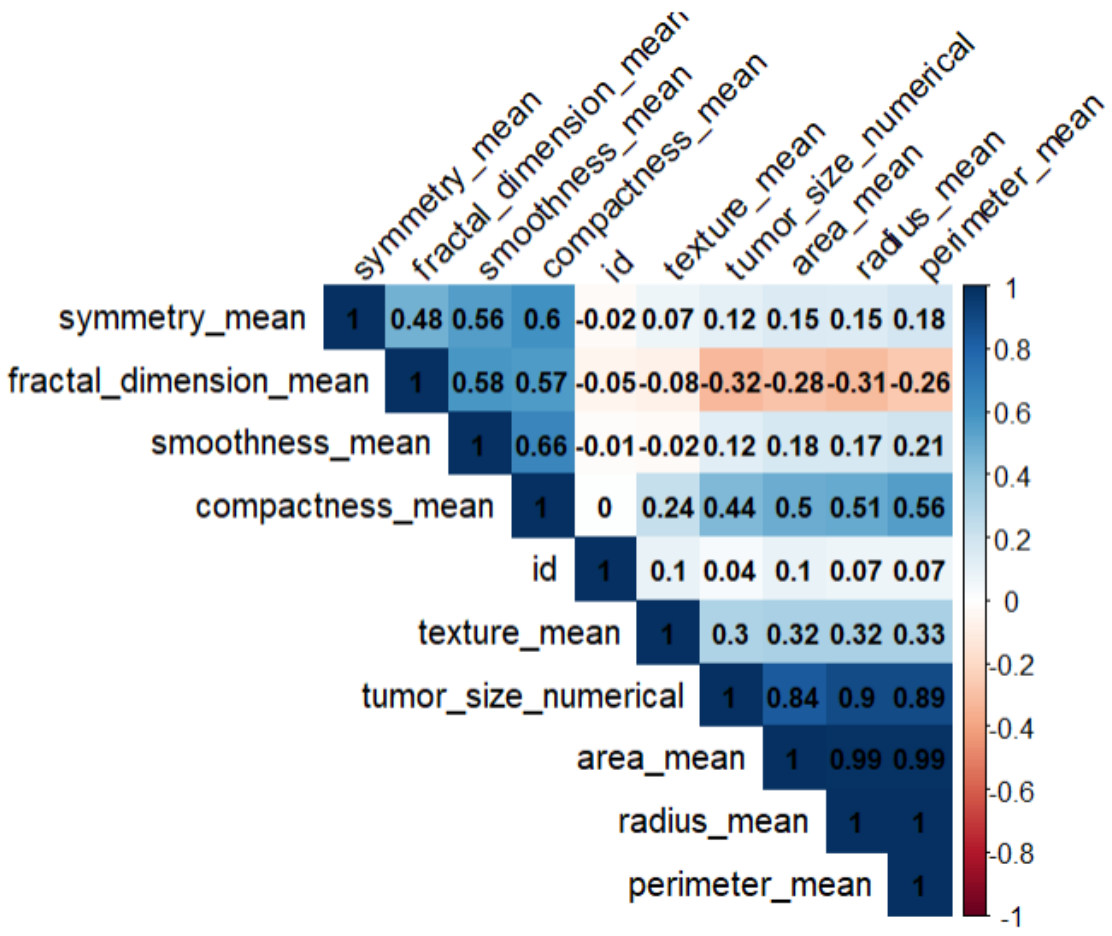



```r

# Select only numeric columns for correlation analysis
numeric_data <- Breastcancer_data2[sapply(Breastcancer_data2, is.numeric)]


# Compute the correlation matrix
cor_matrix <- cor(numeric_data)


# Print the correlation matrix
print(cor_matrix)


# Install and load corrplot package if not already installed
if (!requireNamespace("corrplot", quietly = TRUE)) {
    install.packages("corrplot")
}
library(corrplot)

# Visualize the correlation matrix with adjusted plot size and label size
corrplot(cor_matrix, method = "color", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45, addCoef.col = "black",
         number.cex = 0.8, tl.cex = 1.0)

# The correlation coefficients provided in the matrix indicate the strength and direction of the linear relationship between different features and tumor diagnosis. For instance, the radius_mean, perimeter_mean, and area_mean show strong positive correlations with tumor diagnosis, with coefficients of approximately 0.74, 0.73, and 0.74, respectively, indicating that larger values of these features tend to be associated with malignant tumors. Similarly, compactness_mean and smoothness_mean also show moderate positive correlations with tumor diagnosis, suggesting that higher compactness and smoothness values are associated with malignant tumors. Conversely, fractal_dimension_mean shows a moderate negative correlation with tumor diagnosis, implying that lower values of fractal dimension tend to be associated with malignant tumors.
```

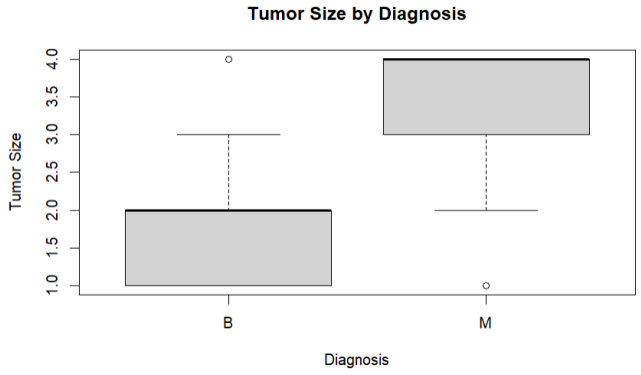


```
texture_mean             0.0997698912  0.32378189    1.00000000     0.32953306
0.32108570     -0.02338852     0.2367022221    0.07140098
perimeter_mean           0.0731594119  0.99785528    0.32953306     1.00000000
0.98650680      0.20727816     0.5569362109    0.18302721
area_mean                0.0968928233  0.98735717    0.32108570     0.98650680
1.00000000      0.17702838     0.4985016822    0.15129308
smoothness_mean         -0.0129681975  0.17058119   -0.02338852     0.20727816
0.17702838      1.00000000     0.6591232152    0.55777479
compactness_mean         0.0000957011  0.50612358    0.23670222     0.55693621
0.49850168      0.65912322     1.0000000000    0.60264105
symmetry_mean           -0.0221140609  0.14774124    0.07140098     0.18302721
0.15129308      0.55777479     0.6026410484    1.00000000
fractal_dimension_mean -0.0525114476 -0.31163083   -0.07643718    -0.26147691
-0.28310981      0.58479200     0.5653686634    0.47992133
tumor_size_numerical     0.0360597917  0.89547746    0.30348407     0.89040132
0.83918195      0.12175585     0.4444671476    0.11946552
                      fractal_dimension_mean tumor_size_numerical
id                                -0.05251145           0.03605979
radius_mean                       -0.31163083           0.89547746
texture_mean                      -0.07643718           0.30348407
perimeter_mean                    -0.26147691           0.89040132
area_mean                         -0.28310981           0.83918195
smoothness_mean                    0.58479200           0.12175585
compactness_mean                   0.56536866           0.44446715
symmetry_mean                      0.47992133           0.11946552
fractal_dimension_mean             1.00000000          -0.32403775
tumor_size_numerical              -0.32403775           1.00000000
```

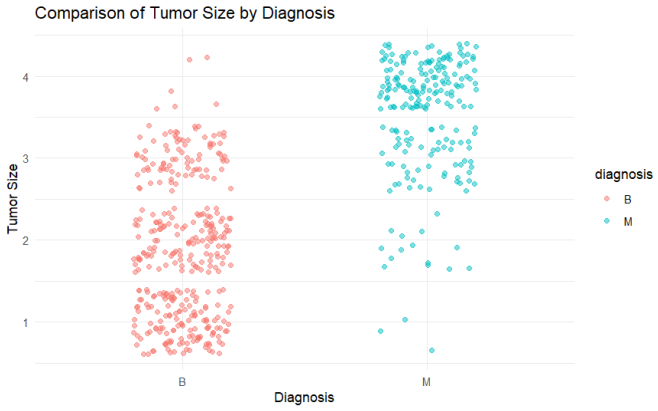




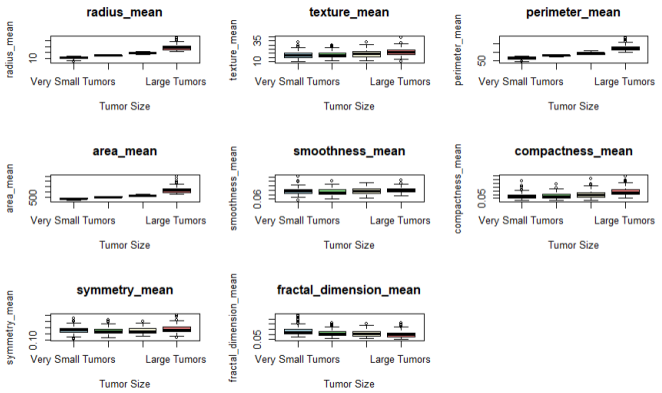

```{r}
# Comparison of tumor size by diagnosis using box plot
boxplot(tumor_size_numerical ~ diagnosis, data = Breastcancer_data2,
        main = "Tumor Size by Diagnosis", xlab = "Diagnosis", ylab = "Tumor Size")

```



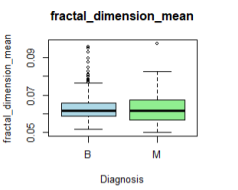
```{r}

# Create a scatter plot
ggplot(Breastcancer_data2, aes(x = diagnosis, y = tumor_size_numerical, color = diagnosis)) +
  geom_point(position = position_jitter(width = 0.2), alpha = 0.5) +
  labs(x = "Diagnosis", y = "Tumor Size", title = "Comparison of Tumor Size by Diagnosis") +
  theme_minimal()
```

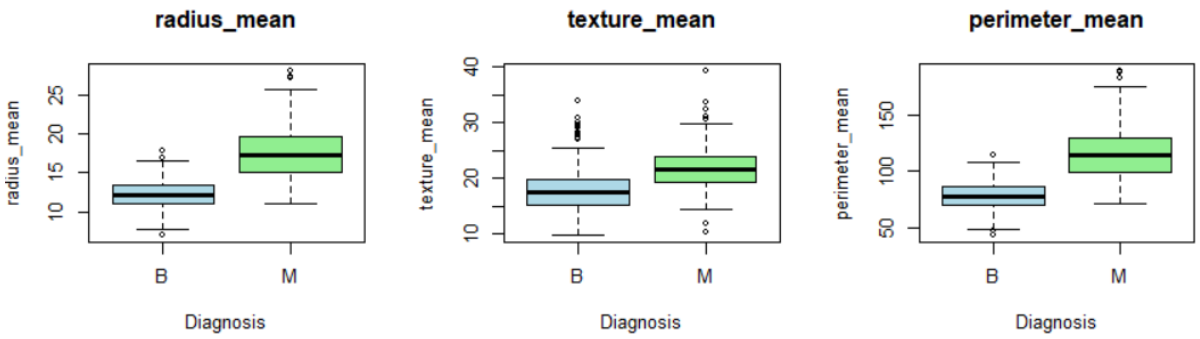


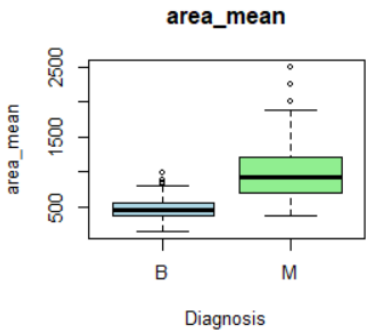

```{r}
# Comparison of morphological characteristics by tumor size using grouped box plots
par(mfrow=c(3, 3))  # Setting up the layout for multiple plots
for (var in c("radius_mean", "texture_mean", "perimeter_mean", "area_mean",
              "smoothness_mean", "compactness_mean", "symmetry_mean",
              "fractal_dimension_mean")) {
  boxplot(get(var) ~ tumor_size, data = Breastcancer_data2, main = var,
          xlab = "Tumor Size", ylab = var, col = c("lightblue", "lightgreen", "lightyellow", "lightcoral"))
}
```

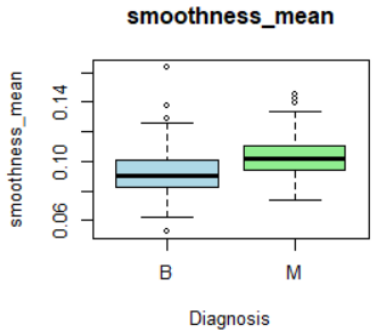


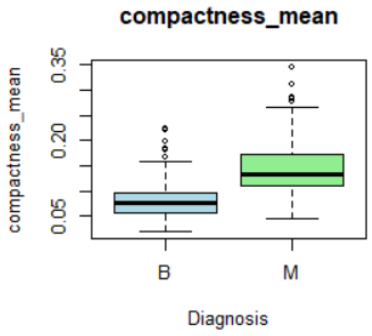

```{r}
# Comparison of multiple variables by diagnosis using grouped box plots
par(mfrow=c(2, 3))
for (var in c("radius_mean", "texture_mean", "perimeter_mean", "area_mean",
              "smoothness_mean", "compactness_mean","fractal_dimension_mean")) {
  boxplot(get(var) ~ diagnosis, data = Breastcancer_data2, main = var,
          xlab = "Diagnosis", ylab = var, col = c("lightblue", "lightgreen"))
}
```












```{r}


# Replace 'M' with 1 and 'D' with 0
Breastcancer_data2$diagnosis <- ifelse(Breastcancer_data2$diagnosis == "M", 1,
                                       ifelse(Breastcancer_data2$diagnosis == "B", 0, Breastcancer_data2$diagnosis))

# If you want to convert the column to numeric type
Breastcancer_data2$diagnosis <- as.numeric(Breastcancer_data2$diagnosis)


```

```{r}
# Mann-Whitney U test for radius_mean
mann_whitney_radius <- wilcox.test(radius_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for radius_mean:")
print(mann_whitney_radius)

# Association result for radius_mean
association_radius <- ifelse(mann_whitney_radius$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for radius_mean:", association_radius))
```

```
[1] "Mann-Whitney U test for radius_mean:"

        Wilcoxon rank sum test with continuity correction

data:  radius_mean by diagnosis
W = 4729, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for radius_mean: Significant association"
```

```{r}
# Mann-Whitney U test for texture_mean
mann_whitney_texture <- wilcox.test(texture_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for texture_mean:")
print(mann_whitney_texture)

# Association result for texture_mean
association_texture <- ifelse(mann_whitney_texture$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for texture_mean:", association_texture))
```

```
[1] "Mann-Whitney U test for texture_mean:"

        Wilcoxon rank sum test with continuity correction

data:  texture_mean by diagnosis
W = 16967, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for texture_mean: Significant association"
```

```{r}
# Mann-Whitney U test for perimeter_mean
mann_whitney_perimeter <- wilcox.test(perimeter_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for perimeter_mean:")
print(mann_whitney_perimeter)

# Association result for perimeter_mean
association_perimeter <- ifelse(mann_whitney_perimeter$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for perimeter_mean:", association_perimeter))
```

```
[1] "Mann-Whitney U test for perimeter_mean:"

        Wilcoxon rank sum test with continuity correction

data:  perimeter_mean by diagnosis
W = 4019, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for perimeter_mean: Significant association"
```

```{r}
# Mann-Whitney U test for area_mean
mann_whitney_area <- wilcox.test(area_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for area_mean:")
print(mann_whitney_area)

# Association result for area_mean
association_area <- ifelse(mann_whitney_area$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for area_mean:", association_area))
```

```
[1] "Mann-Whitney U test for area_mean:"

        Wilcoxon rank sum test with continuity correction

data:  area_mean by diagnosis
W = 4668.5, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for area_mean: Significant association"
```

```{r}
# Mann-Whitney U test for smoothness_mean
mann_whitney_smoothness <- wilcox.test(smoothness_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for smoothness_mean:")
print(mann_whitney_smoothness)

# Association result for smoothness_mean
association_smoothness <- ifelse(mann_whitney_smoothness$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for smoothness_mean:", association_smoothness))
```

```
[1] "Mann-Whitney U test for smoothness_mean:"

        Wilcoxon rank sum test with continuity correction

data:  smoothness_mean by diagnosis
W = 21037, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for smoothness_mean: Significant association"
```

```{r}
# Mann-Whitney U test for compactness_mean
mann_whitney_compactness <- wilcox.test(compactness_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for compactness_mean:")
print(mann_whitney_compactness)

# Association result for compactness_mean
association_compactness <- ifelse(mann_whitney_compactness$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for compactness_mean:", association_compactness))
```

```
[1] "Mann-Whitney U test for compactness_mean:"

        Wilcoxon rank sum test with continuity correction

data:  compactness_mean by diagnosis
W = 10310, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for compactness_mean: Significant association"
```

```r
# Mann-Whitney U test for symmetry_mean
mann_whitney_symmetry <- wilcox.test(symmetry_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for symmetry_mean:")
print(mann_whitney_symmetry)

# Association result for symmetry_mean
association_symmetry <- ifelse(mann_whitney_symmetry$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for symmetry_mean:", association_symmetry))
```

```
[1] "Mann-Whitney U test for symmetry_mean:"

        Wilcoxon rank sum test with continuity correction

data:  symmetry_mean by diagnosis
W = 22814, p-value = 2.268e-15
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for symmetry_mean: Significant association"
```

```r
# Mann-Whitney U test for fractal_dimension_mean
mann_whitney_fractal_dimension <- wilcox.test(fractal_dimension_mean ~ diagnosis, data = Breastcancer_data2)
print("Mann-Whitney U test for fractal_dimension_mean:")
print(mann_whitney_fractal_dimension)

# Association result for fractal_dimension_mean
association_fractal_dimension <- ifelse(mann_whitney_fractal_dimension$p.value < 0.05, "Significant association", "No significant association")
print(paste("Association result for fractal_dimension_mean:", association_fractal_dimension))
```

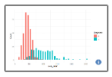
```
[1] "Mann-Whitney U test for fractal_dimension_mean:"

        Wilcoxon rank sum test with continuity correction

data:  fractal_dimension_mean by diagnosis
W = 39013, p-value = 0.5372
alternative hypothesis: true location shift is not equal to 0

[1] "Association result for fractal_dimension_mean: No significant association"
```

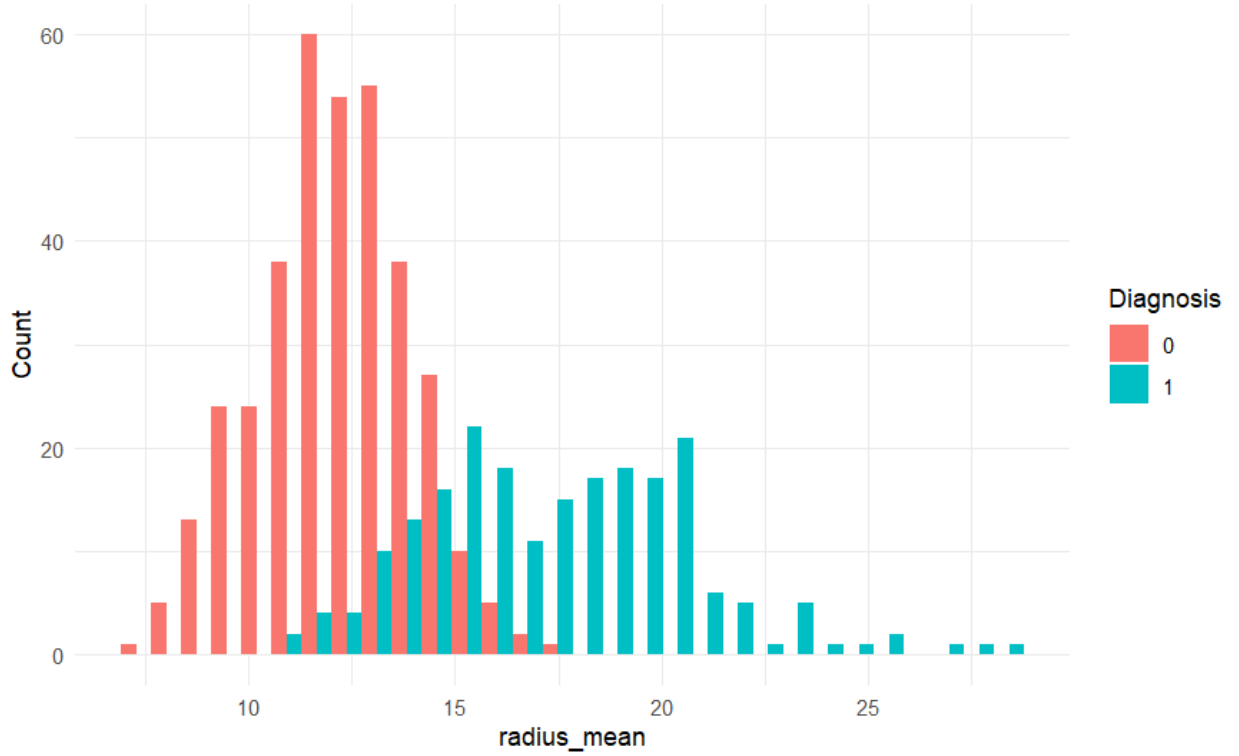
```r
# Get unique values in the diagnosis column
diagnosis_levels <- unique(Breastcancer_data2$diagnosis)

# Perform chi-square test for each diagnosis variable
for (level in diagnosis_levels) {
  # Subset the data for the current diagnosis level
  subset_data <- Breastcancer_data2[Breastcancer_data2$diagnosis == level, ]

  # Create a contingency table for tumor_size and the current diagnosis level
  cont_table <- table(subset_data$tumor_size)

  # Perform chi-square test
  chi_sq_test <- chisq.test(cont_table)

  # Print the results
  print(paste("Chi-square test for", level, ":", sep = " "))
  print(chi_sq_test)
}
```

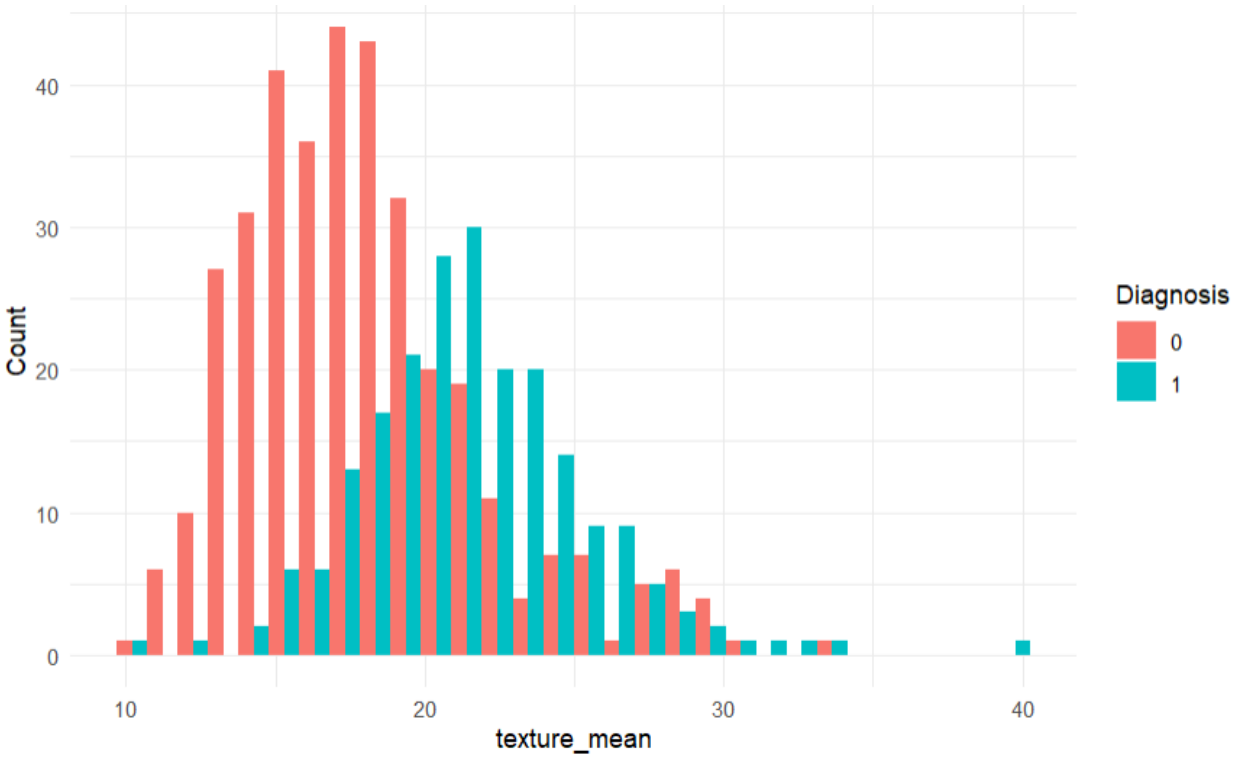
```
[1] "Chi-square test for 1 :"

        Chi-squared test for given probabilities

data:  cont_table
X-squared = 206.53, df = 3, p-value < 2.2e-16

[1] "Chi-square test for 0 :"

        Chi-squared test for given probabilities

data:  cont_table
X-squared = 123.77, df = 3, p-value < 2.2e-16
```

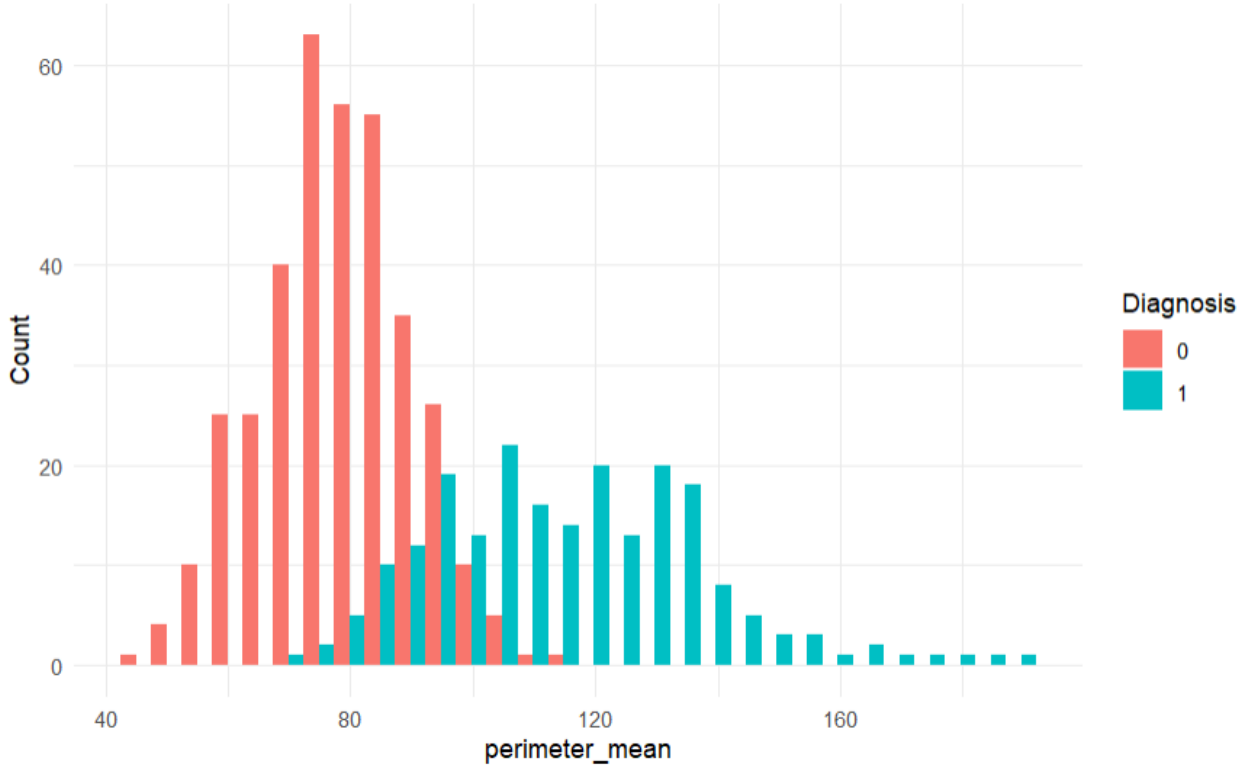
```r
# Load the necessary libraries
library(ggplot2)
library(dplyr)

# Prepare the data
Breastcancer_data2$diagnosis <- as.factor(Breastcancer_data2$diagnosis)

# Create a function to plot the association between diagnosis and a given feature
plot_feature_association <- function(feature_name) {
  ggplot(Breastcancer_data2, aes(x = !!sym(feature_name), fill = diagnosis)) +
    geom_histogram(position = "dodge") +
    labs(x = feature_name, y = "Count", fill = "Diagnosis") +
    theme_minimal()
}

# Plot the association for each feature
plot_feature_association("radius_mean")
plot_feature_association("texture_mean")
plot_feature_association("perimeter_mean")
plot_feature_association("area_mean")
```

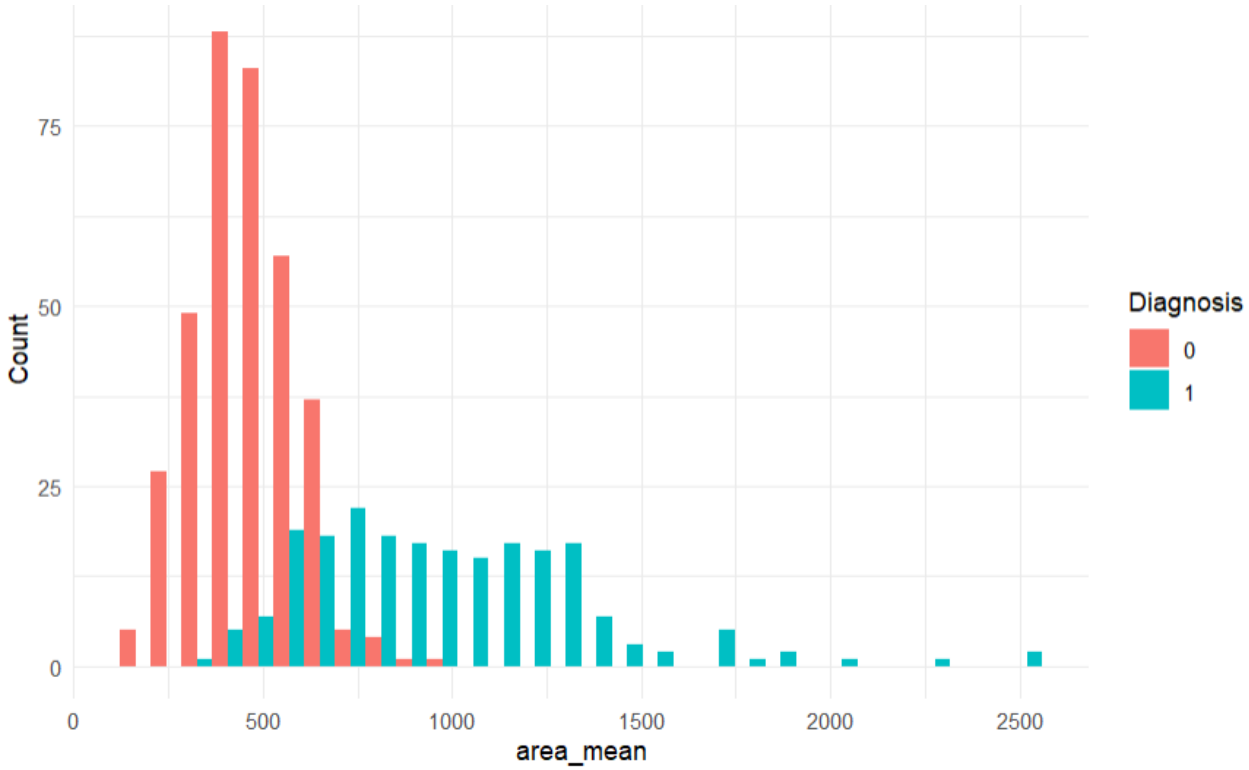


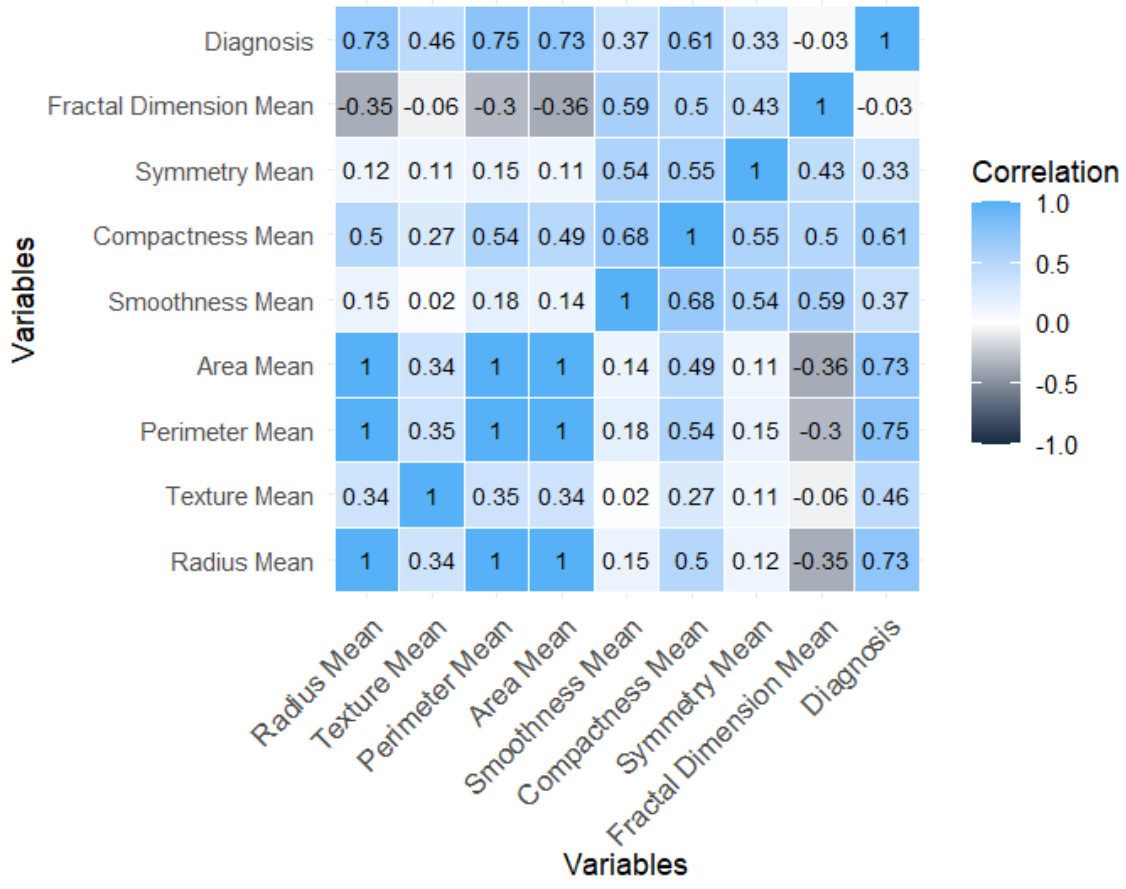

60
40
20
0
Count
10
15
20
25
radius_mean
Diagnosis
0
1


40
30
20
10
0
Count
10
20
30
40
texture_mean
Diagnosis
0
1


60
40
20
0
Count
40
80
120
160
perimeter_mean
Diagnosis
0
1


75
50
25
0
Count
0
500
1000
1500
2000
2500
area_mean
Diagnosis
0
1

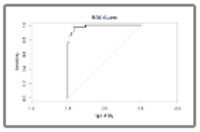
```r
# Convert "diagnosis" variable to numeric
Breastcancer_data2$diagnosis <- as.numeric(as.factor(Breastcancer_data2$diagnosis))

# Specify variables for correlation analysis (including "diagnosis")
correlation_variables <- c("radius_mean", "texture_mean", "perimeter_mean",
                           "area_mean", "smoothness_mean", "compactness_mean",
                           "symmetry_mean", "fractal_dimension_mean", "diagnosis")

# Calculate Spearman correlation matrix
correlation_matrix_spearman <- cor(Breastcancer_data2[, correlation_variables], method = "spearman")

# Convert correlation matrix to data frame
correlation_df_spearman <- as.data.frame(as.table(correlation_matrix_spearman))

# Add variable names
correlation_df_spearman$Var1 <- factor(correlation_df_spearman$Var1, levels = correlation_variables)
correlation_df_spearman$Var2 <- factor(correlation_df_spearman$Var2, levels = correlation_variables)

# Plot heatmap with Spearman correlation values displayed
library(ggplot2)

ggplot(correlation_df_spearman, aes(Var1, Var2, fill = Freq, label = round(Freq, 2))) +
  geom_tile(color = "white") +
  geom_text(color = "black", size = 3) +
  scale_fill_gradient2(low = "#132B43", high = "#56B1F7", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 10, hjust = 1)) +
  coord_fixed() +
  labs(x = "Variables", y = "Variables") +  # Adjust axis labels
  scale_x_discrete(labels = c("Radius Mean", "Texture Mean", "Perimeter Mean",
                              "Area Mean", "Smoothness Mean", "Compactness Mean",
                              "Symmetry Mean", "Fractal Dimension Mean", "Diagnosis")) +  # Include variable names
 scale_y_discrete(labels = c("Radius Mean", "Texture Mean", "Perimeter Mean",
                              "Area Mean", "Smoothness Mean", "Compactness Mean",
                              "Symmetry Mean", "Fractal Dimension Mean", "Diagnosis"))  # Include variable names
```



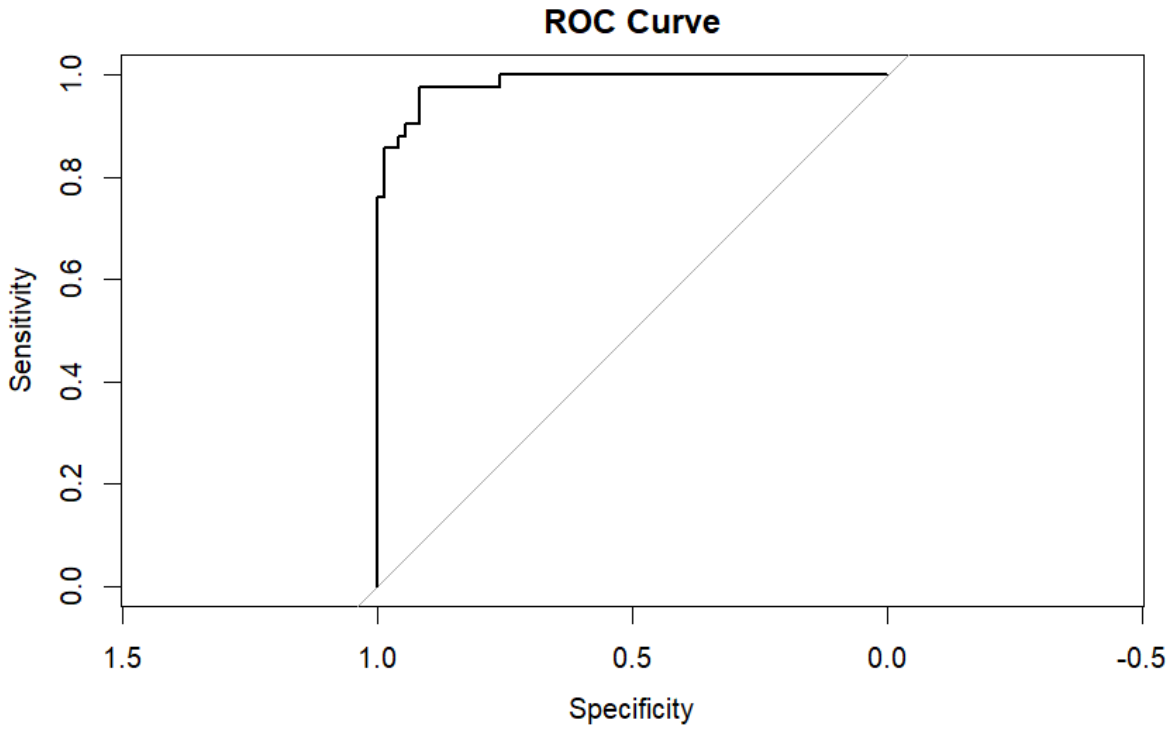
```r
# Load necessary libraries
library(ggplot2)
library(caret)
library(dplyr)
library(e1071)

# Prepare the data
Breastcancer_data2$diagnosis <- as.factor(Breastcancer_data2$diagnosis)

# Split the data into training and test sets
set.seed(123)
train_index <- createDataPartition(Breastcancer_data2$diagnosis, p = 0.8, list = FALSE)
train_data <- Breastcancer_data2[train_index, ]
test_data <- Breastcancer_data2[-train_index, ]

# Train the SVM model for classification
svm_model <- svm(diagnosis ~ ., data = train_data, type = "C-classification", kernel = "radial")

# Evaluate the model on the test set
predictions <- predict(svm_model, newdata = test_data)
accuracy <- sum(predictions == test_data$diagnosis) / nrow(test_data)
print(paste("Accuracy:", accuracy))

# Plot the association between diagnosis and each feature
for (feature in names(Breastcancer_data2)[2:ncol(Breastcancer_data2)]) {
  plot_feature_association(feature)
}

# Function to plot the association between diagnosis and a feature
plot_feature_association <- function(feature_name) {
  ggplot(Breastcancer_data2, aes_string(x = feature_name, fill = "diagnosis")) +
    geom_histogram(position = "dodge", binwidth = 1) +
    labs(x = feature_name, y = "Count", fill = "Diagnosis") +
    theme_minimal()
}
```

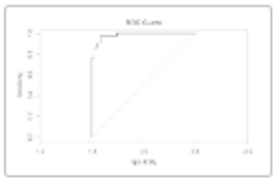
```
[1] "Accuracy: 0.920353982300885"
```

```r

library(pROC)
library(caret)
library(ggplot2)
library(e1071)

# Train the SVM model for classification with probability estimates
svm_model <- svm(diagnosis ~ ., data = train_data, type = "C-classification", kernel = "radial", probability = TRUE)

# Predict probabilities
probabilities <- predict(svm_model, newdata = test_data, probability = TRUE)

# Extract probabilities for the positive class
# Assuming the positive class is the second factor level, adjust if necessary
probs <- attr(probabilities, "probabilities")[,2]

# ROC curve using the pROC package
roc_obj <- roc(test_data$diagnosis, probs)
plot(roc_obj, main = "ROC Curve")
print(paste("AUC:", auc(roc_obj)))


predictions <- predict(svm_model, newdata = test_data)

# Generate the confusion matrix
conf_matrix <- confusionMatrix(as.factor(predictions), as.factor(test_data$diagnosis))

# Print the confusion matrix
print(conf_matrix$table)

# print the summary of the confusion matrix
print(conf_matrix)
```




R Console


ROC Curve
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity




```
Confusion Matrix and Statistics

          Reference
Prediction  1  2
         1 66  4
         2  5 38

               Accuracy : 0.9204
                 95% CI : (0.8542, 0.9629)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : 9.656e-13

                  Kappa : 0.8303

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9296
            Specificity : 0.9048
         Pos Pred Value : 0.9429
         Neg Pred Value : 0.8837
             Prevalence : 0.6283
         Detection Rate : 0.5841
   Detection Prevalence : 0.6195
      Balanced Accuracy : 0.9172

       'Positive' Class : 1
```

```r
# Convert diagnosis to a factor if it's not already
Breastcancer_data2$diagnosis <- as.factor(Breastcancer_data2$diagnosis)

# Split the data into training and test sets
set.seed(123)  # for reproducibility
train_index <- createDataPartition(Breastcancer_data2$diagnosis, p = 0.8, list = FALSE)
train_data <- Breastcancer_data2[train_index, ]
test_data <- Breastcancer_data2[-train_index, ]


logistic_model <- glm(diagnosis ~ ., data = train_data, family = binomial())


# Predicting probabilities
probabilities <- predict(logistic_model, newdata = test_data, type = "response")

# Predicting class labels
predicted_classes <- ifelse(probabilities > 0.5, levels(train_data$diagnosis)[2], levels(train_data$diagnosis)[1])
```

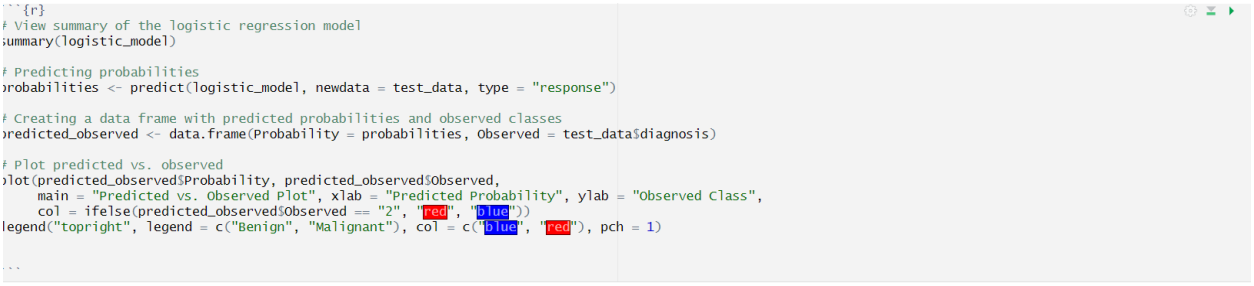
```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```{r}
conf_matrix <- confusionMatrix(as.factor(predicted_classes), test_data$diagnosis)
print(conf_matrix$table)
print(conf_matrix$overall['Accuracy'])

# print the summary of the confusion matrix
print(conf_matrix)
```

```
          Reference
Prediction  1  2
         1 69  2
         2  2 40
 Accuracy 
0.9646018 
Confusion Matrix and Statistics

          Reference
Prediction  1  2
         1 69  2
         2  2 40

               Accuracy : 0.9646          
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.6283          
    P-Value [Acc > NIR] : <2e-16          

                  Kappa : 0.9242          

 Mcnemar's Test P-Value : 1               

            Sensitivity : 0.9718          
            Specificity : 0.9524          
         Pos Pred Value : 0.9718          
         Neg Pred Value : 0.9524          
             Prevalence : 0.6283          
         Detection Rate : 0.6106          
   Detection Prevalence : 0.6283          
      Balanced Accuracy : 0.9621          

       'Positive' Class : 1               
```

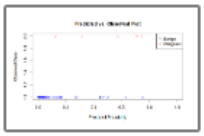
```{r}
# Get summary of the logistic regression model
summary_glm <- summary(logistic_model)
print(summary_glm)

#Interpretation of the coefficients suggests that texture_mean, area_mean, and smoothness_mean are statistically significant predictors of tumor diagnosis, as their p-values are less than the conventional threshold of 0.05. This implies that changes in these features are associated with changes in the likelihood of tumor diagnosis. On the other hand, radius_mean and symmetry_mean also show some significance with p-values close to the threshold, indicating they might have some predictive value but require further investigation. However, other variables such as compactness_mean, perimeter_mean, and fractal_dimension_mean do not appear to be statistically significant predictors in this model.
```

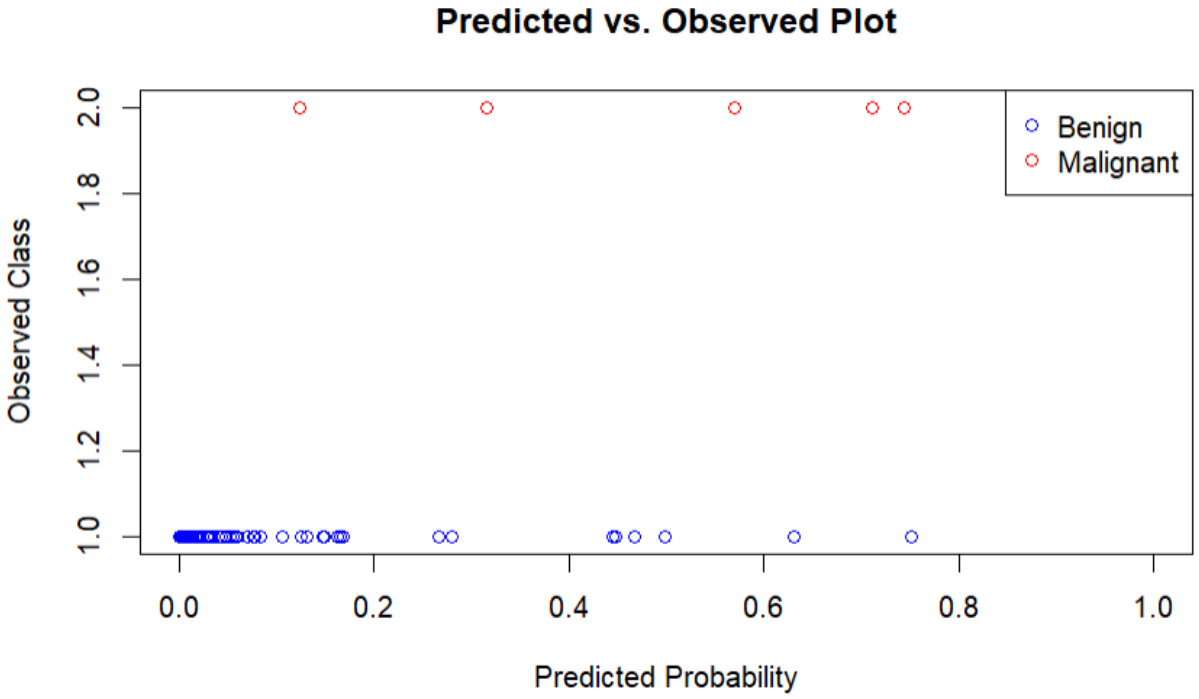
```
Call:
glm(formula = diagnosis ~ ., family = binomial(), data = train_data)

Coefficients: (1 not defined because of singularities)
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)               1.100e+01  1.383e+01   0.795 0.426644
id                       -4.446e-09  4.914e-09  -0.905 0.365551
radius_mean              -6.822e+00  3.781e+00  -1.804 0.071166 .
texture_mean              3.694e-01  6.820e-02   5.416 6.09e-08 ***
perimeter_mean            1.221e-01  5.360e-01   0.228 0.819790
area_mean                 8.439e-02  2.153e-02   3.919 8.88e-05 ***
smoothness_mean           9.816e+01  2.632e+01   3.729 0.000192 ***
compactness_mean          2.320e+01  2.367e+01   0.980 0.326943
symmetry_mean             3.040e+01  1.341e+01   2.267 0.023377 *
fractal_dimension_mean   -9.267e+01  9.473e+01  -0.978 0.327981
tumor_sizeSmall Tumors    2.383e+00  1.421e+00   1.677 0.093623 .
tumor_sizeMedium Tumors   2.596e+00  1.784e+00   1.455 0.145631
tumor_sizeLarge Tumors    6.610e-01  2.512e+00   0.263 0.792411
tumor_size_numerical             NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 602.31  on 455  degrees of freedom
Residual deviance: 127.51  on 443  degrees of freedom
AIC: 153.51

Number of Fisher Scoring iterations: 9
```

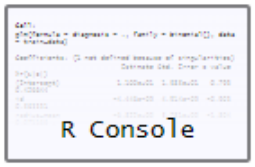


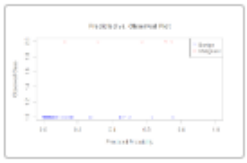
```r
# View summary of the logistic regression model
summary(logistic_model)

# Predicting probabilities
probabilities <- predict(logistic_model, newdata = test_data, type = "response")

# Creating a data frame with predicted probabilities and observed classes
predicted_observed <- data.frame(Probability = probabilities, Observed = test_data$diagnosis)

# Plot predicted vs. observed
plot(predicted_observed$Probability, predicted_observed$Observed,
     main = "Predicted vs. Observed Plot", xlab = "Predicted Probability", ylab = "Observed Class",
     col = ifelse(predicted_observed$Observed == "2", "red", "blue"))
legend("topright", legend = c("Benign", "Malignant"), col = c("blue", "red"), pch = 1)
```


R Console



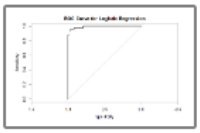

Predicted vs. Observed Plot
Observed Class
1.0
1.2
1.4
1.6
1.8
2.0
0.0
0.2
0.4
0.6
0.8
1.0
Predicted Probability
Benign
Malignant

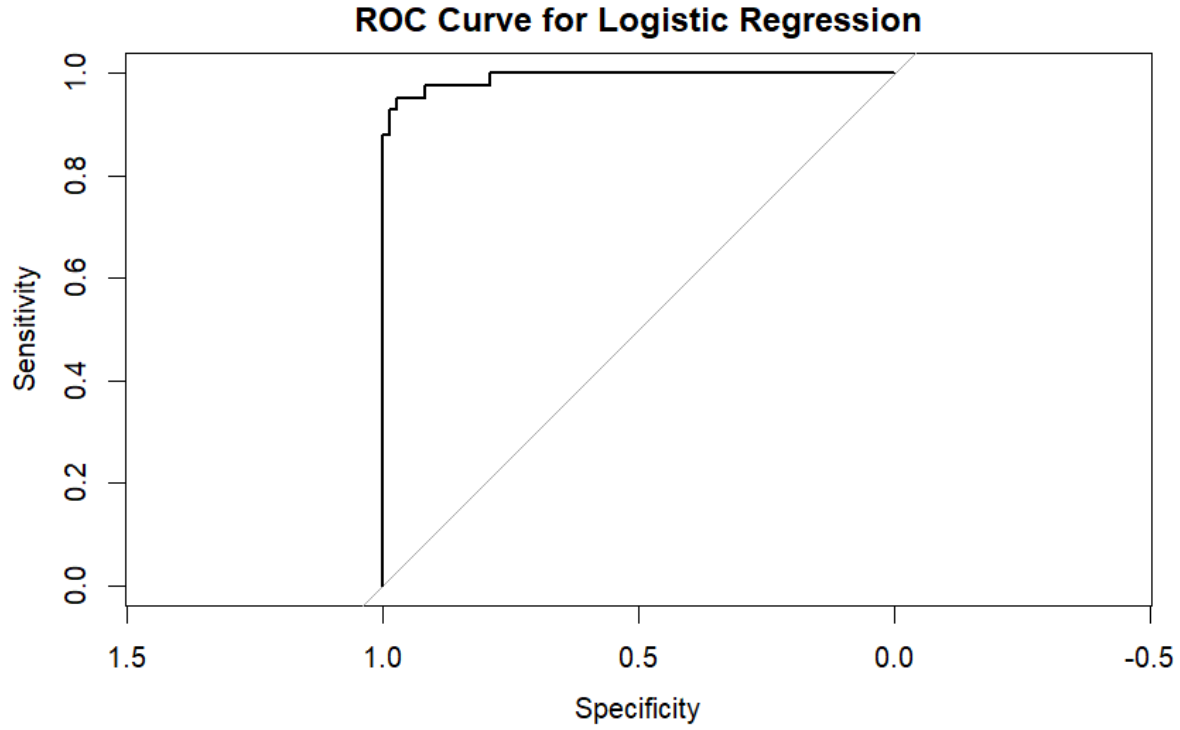


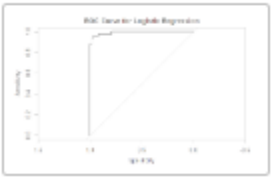


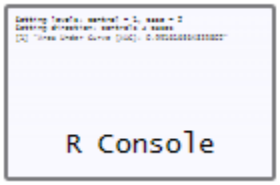
```
Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              1.100e+01  1.383e+01   0.795 0.426644
id                      -4.446e-09  4.914e-09  -0.905 0.365551
radius_mean             -6.822e+00  3.781e+00  -1.804 0.071166 .
texture_mean             3.694e-01  6.820e-02   5.416 6.09e-08 ***
perimeter_mean           1.221e-01  5.360e-01   0.228 0.819790
area_mean                8.439e-02  2.153e-02   3.919 8.88e-05 ***
smoothness_mean          9.816e+01  2.632e+01   3.729 0.000192 ***
compactness_mean         2.320e+01  2.367e+01   0.980 0.326943
symmetry_mean            3.040e+01  1.341e+01   2.267 0.023377 *
fractal_dimension_mean  -9.267e+01  9.473e+01  -0.978 0.327981
tumor_sizeSmall Tumors   2.383e+00  1.421e+00   1.677 0.093623 .
tumor_sizeMedium Tumors  2.596e+00  1.784e+00   1.455 0.145631
tumor_sizeLarge Tumors   6.610e-01  2.512e+00   0.263 0.792411
tumor_size_numerical            NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 602.31  on 455  degrees of freedom
Residual deviance: 127.51  on 443  degrees of freedom
AIC: 153.51

Number of Fisher Scoring iterations: 9
```

```r
# Load necessary library
library(pROC)

# Create ROC curve object
roc_obj <- roc(test_data$diagnosis, probabilities)

# Plot ROC curve
plot(roc_obj, main = "ROC Curve for Logistic Regression")

# Print AUC
print(paste("Area Under Curve (AUC):", auc(roc_obj)))
```










```
Setting levels: control = 1, case = 2
Setting direction: controls < cases
[1] "Area Under Curve (AUC): 0.991616364855802"
```

```{r}
conf_matrix <- confusionMatrix(as.factor(predicted_classes), test_data$diagnosis)
print(conf_matrix$table)
print(conf_matrix$overall['Accuracy'])

# print the summary of the confusion matrix
print(conf_matrix)


```

```
          Reference
Prediction  1  2
         1 69  2
         2  2 40
 Accuracy
0.9646018
Confusion Matrix and Statistics

          Reference
Prediction  1  2
         1 69  2
         2  2 40

               Accuracy : 0.9646
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9242

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9718
            Specificity : 0.9524
         Pos Pred Value : 0.9718
         Neg Pred Value : 0.9524
             Prevalence : 0.6283
         Detection Rate : 0.6106
   Detection Prevalence : 0.6283
      Balanced Accuracy : 0.9621

       'Positive' Class : 1
```

```{r}

# Set seed for reproducibility
set.seed(123)

# Sample 10 random observations from the test set
sample_test_data <- test_data[sample(nrow(test_data), 10), ]

# Predict with Logistic Regression
logistic_predictions <- predict(logistic_model, newdata = sample_test_data, type = "response")
logistic_predicted_classes <- ifelse(logistic_predictions > 0.5, levels(train_data$diagnosis)[2], levels(train_data$diagnosis)[1])

# Combine actual and predicted into a data frame for easy comparison
comparison_data <- data.frame(
  Actual = sample_test_data$diagnosis,
  Predicted_Logistic = logistic_predicted_classes
)

# Print results
print("Model Predictions Comparison:")
print(comparison_data)


```

R Console

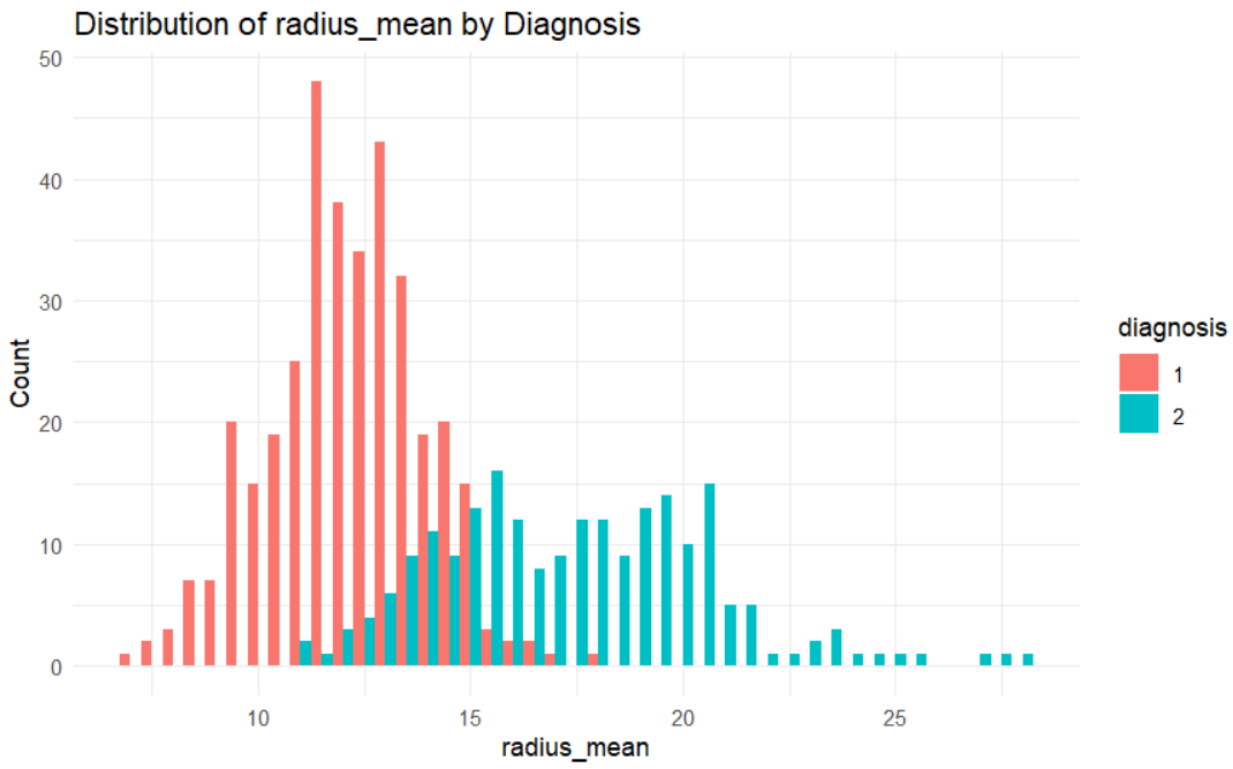
data.frame
10 × 2

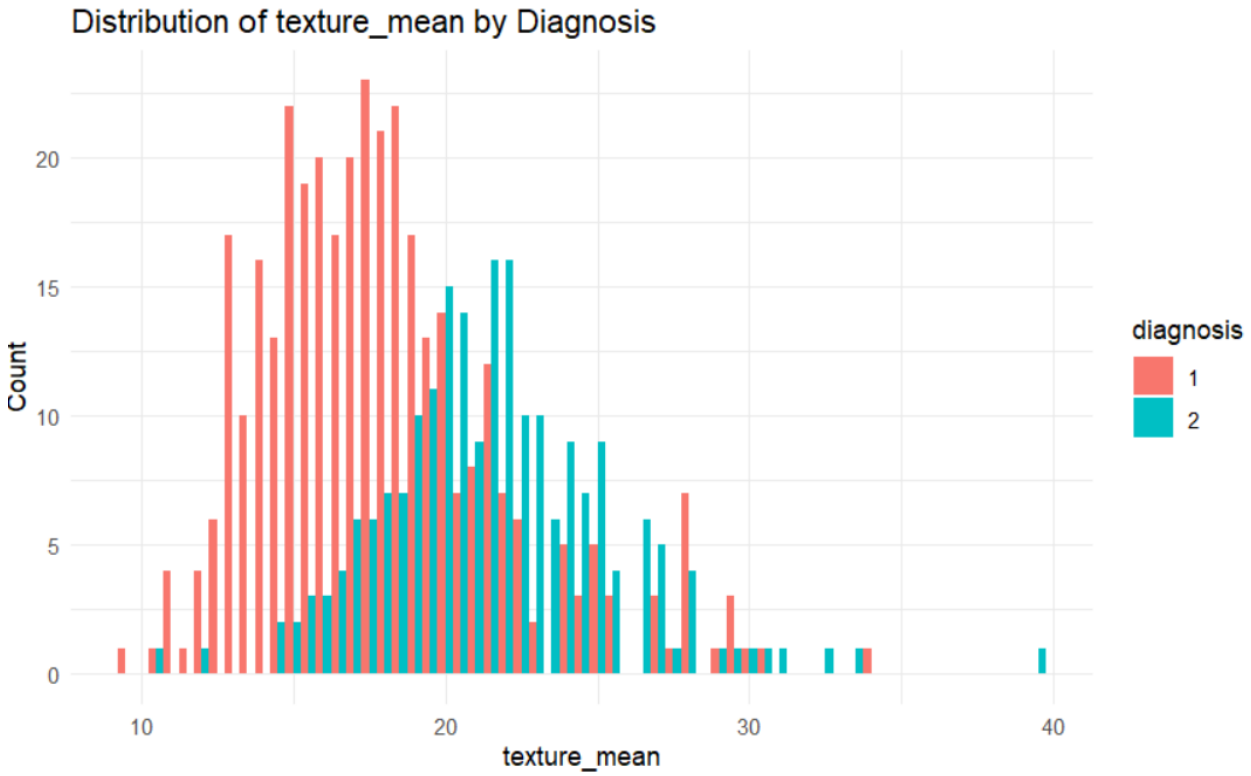
Description: df [10 × 2]

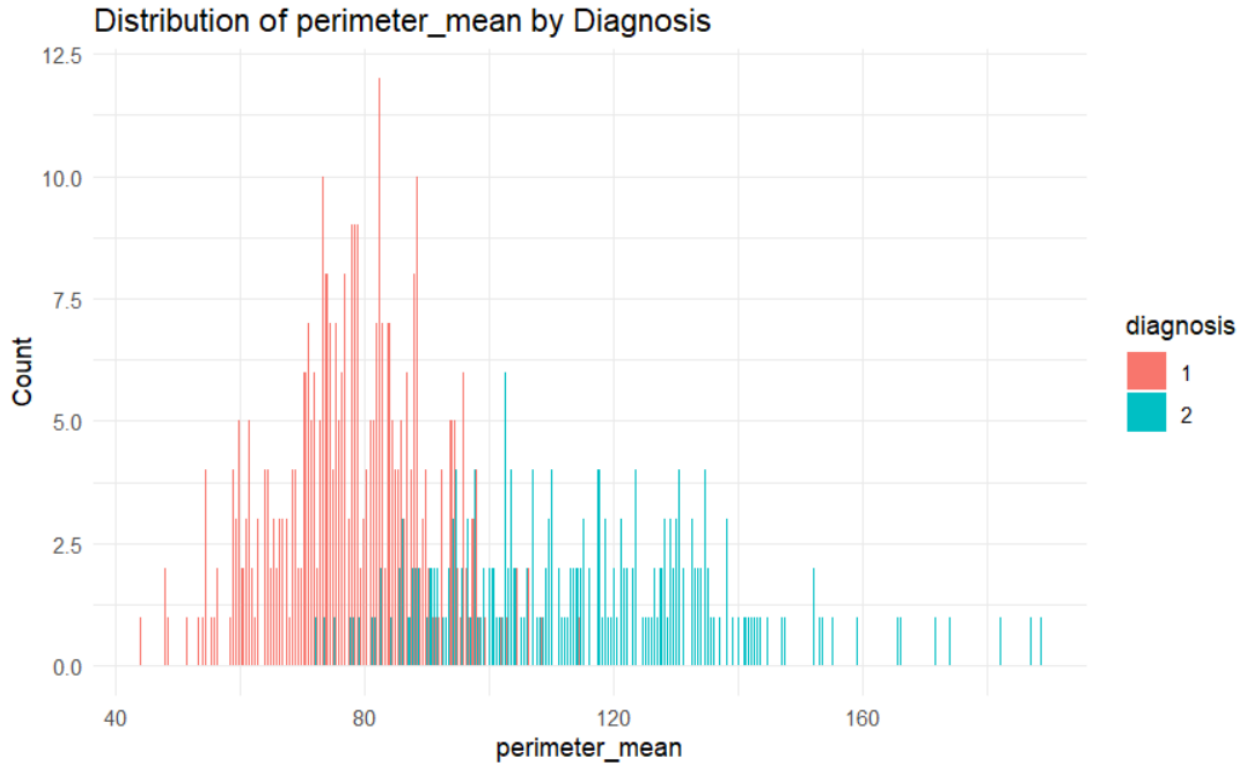
	Actual <fctr>	Predicted_Logistic <chr>
151	1	1
395	1	1
269	1	1
69	1	1
336	2	2
233	1	1
267	1	1
240	2	2
510	2	2
551	1	1

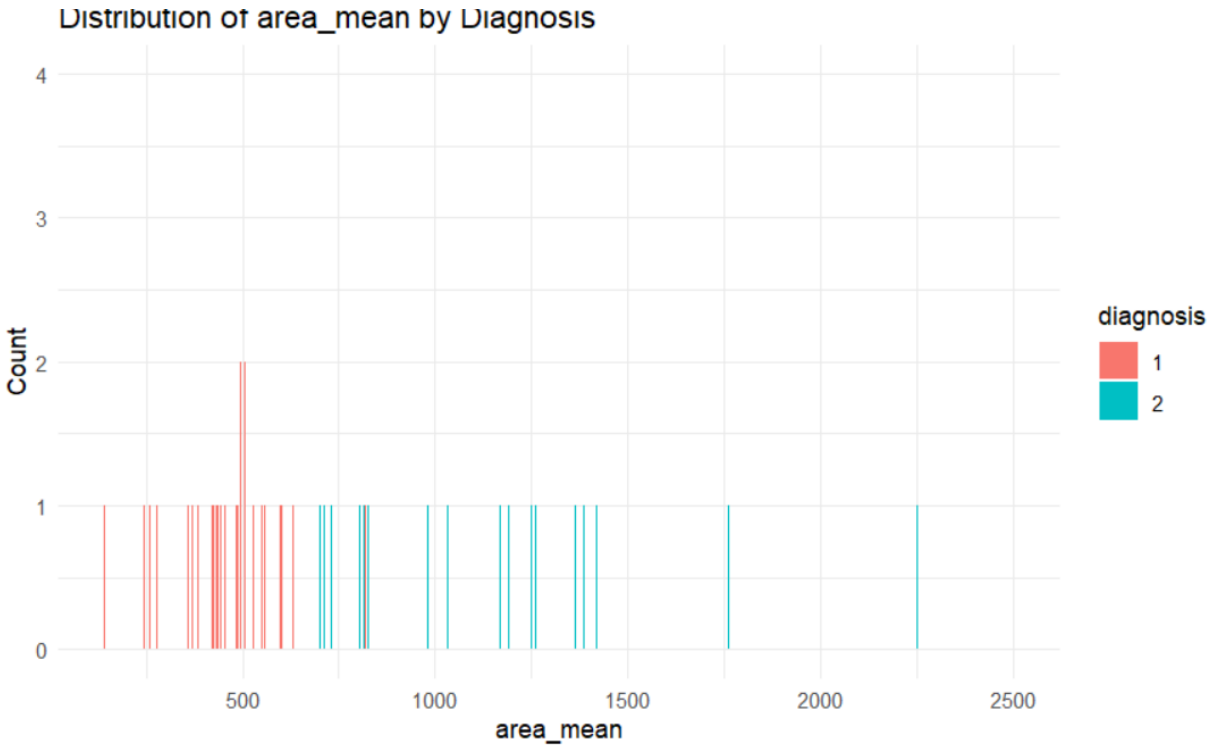
1-10 of 10 rows

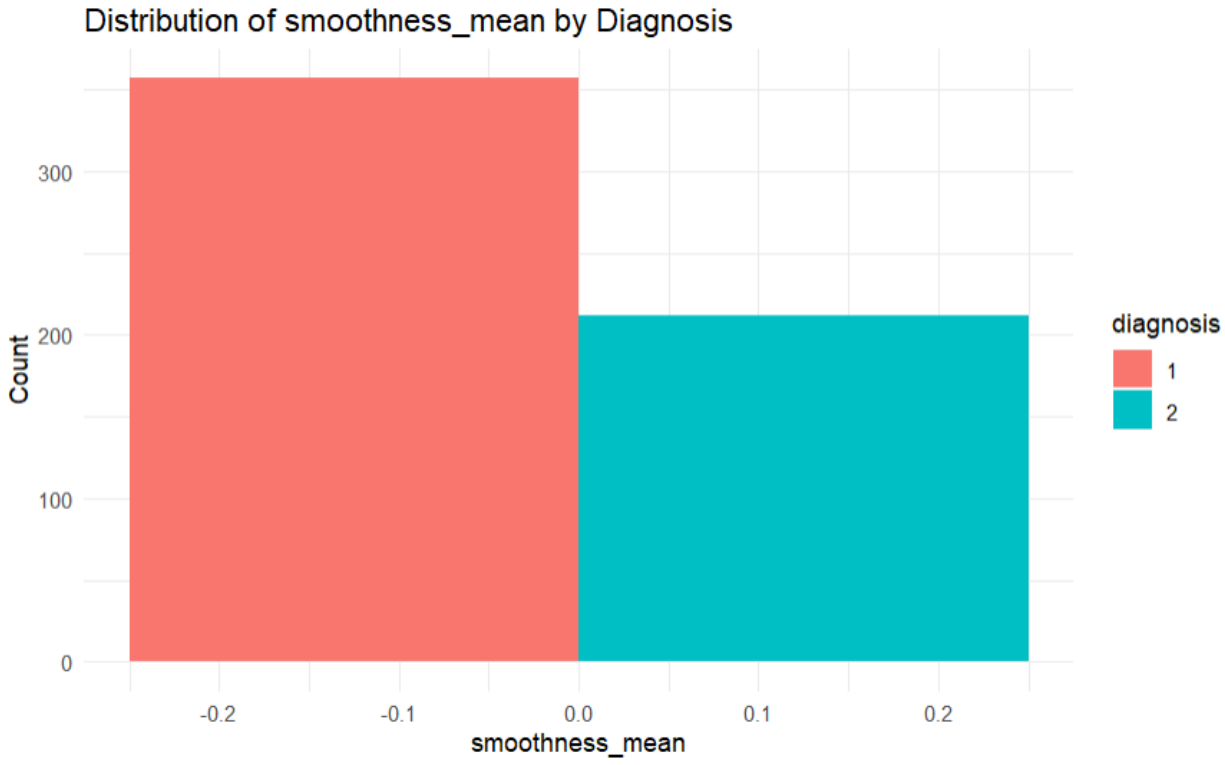
```{r}
library(ggplot2)

# Assuming 'Breastcancer_data2' is your dataset
features <- c("radius_mean", "texture_mean", "perimeter_mean", "area_mean", "smoothness_mean")

# Loop through features and create histograms
for (feature in features) {
  print(ggplot(Breastcancer_data2, aes_string(x = feature, fill = "diagnosis")) +
    geom_histogram(position = "dodge", binwidth = 0.5) +
    labs(title = paste("Distribution of", feature, "by Diagnosis"),
         x = feature,
         y = "Count") +
    theme_minimal())
}
```



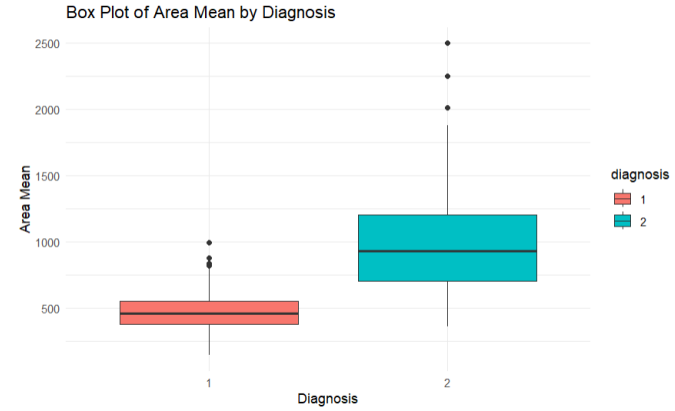

Distribution of texture_mean by Diagnosis
Count
texture_mean
diagnosis
1
2
0
5
10
15
20
10
20
30
40


Distribution of perimeter_mean by Diagnosis
Count
perimeter_mean
diagnosis
1
2
0.0
2.5
5.0
7.5
10.0
12.5
40
80
120
160

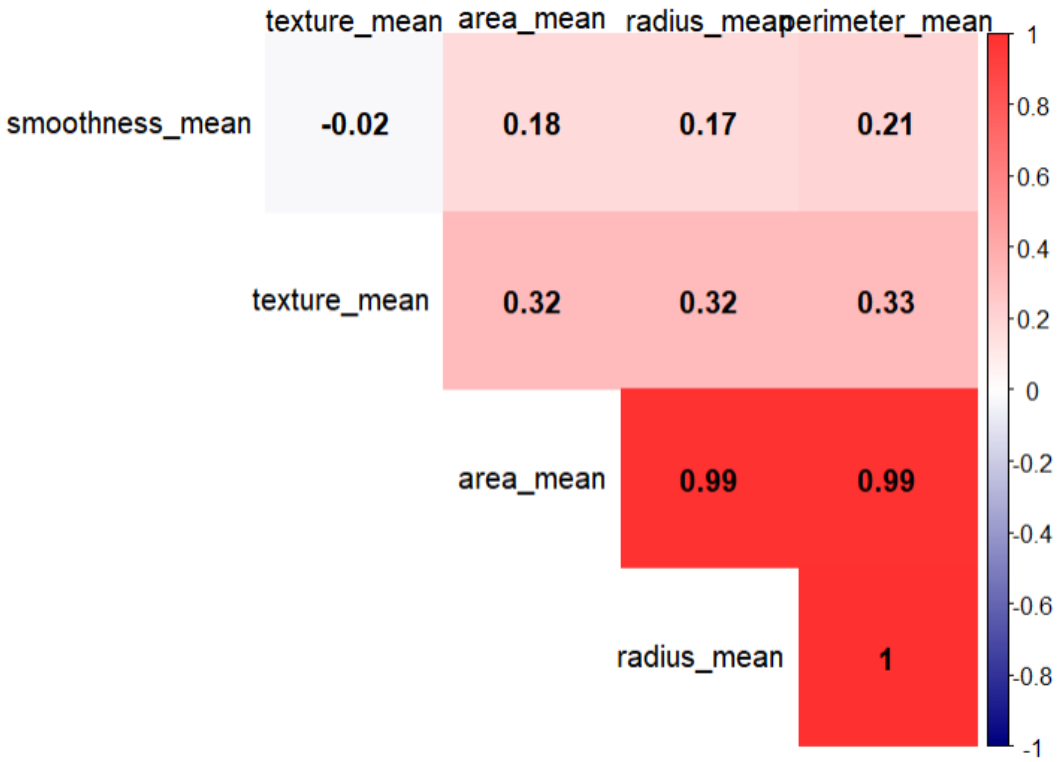

Distribution of area_mean by Diagnosis
Count
area_mean
diagnosis
1
2
0
1
2
3
4
500
1000
1500
2000
2500


Distribution of smoothness_mean by Diagnosis
Count
smoothness_mean
diagnosis
1
2
0
100
200
300
-0.2
-0.1
0.0
0.1
0.2

```r
ggplot(Breastcancer_data2, aes(x = diagnosis, y = area_mean, fill = diagnosis)) +
  geom_boxplot() +
  labs(title = "Box Plot of Area Mean by Diagnosis",
       x = "Diagnosis",
       y = "Area Mean") +
  theme_minimal()
```



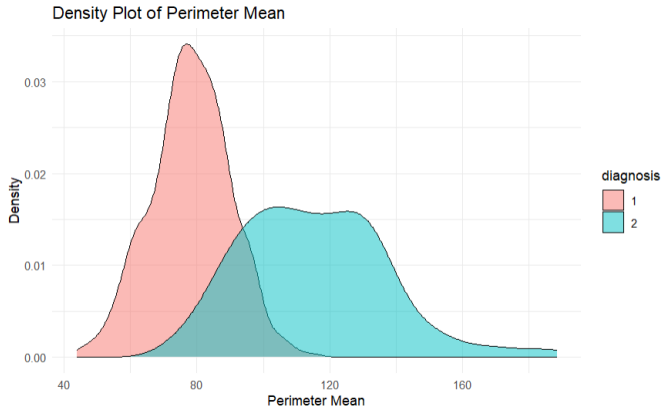

```r
library(corrplot)

# Compute the correlation matrix
cor_matrix <- cor(Breastcancer_data2[, c("radius_mean", "texture_mean", "perimeter_mean", "area_mean", "smoothness_mean")])

# Plot the correlation matrix with a straightforward and safe color scheme
corrplot(cor_matrix, method = "color",
         col = colorRampPalette(c("navy", "white", "firebrick1"))(200),  # Simple and effective gradient: navy to white to firebrick
         type = "upper",  # Display only the upper part of the matrix
         order = "hclust",  # Order variables by hierarchical clustering
         tl.col = "black",  # Text label color
         tl.srt = 0,  # Text label rotation set to 0 to avoid problems
         addCoef.col = "black",  # Color for the correlation coefficients
         number.cex = 1.0,  # Adjust size of the coefficient labels for clarity
         diag = FALSE)  # Avoid showing the diagonal (self-correlation is always 1)
```



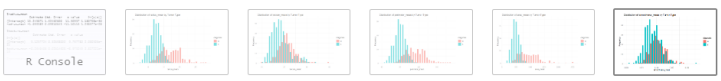
```{r}
library(caret)
# Split data into training and test sets
set.seed(123)
index <- createDataPartition(Breastcancer_data2$diagnosis, p = 0.75, list = FALSE)
train_set <- Breastcancer_data2[index,]
test_set <- Breastcancer_data2[-index,]

# Fit the logistic regression model
model <- glm(diagnosis ~ ., data = train_set, family = "binomial")

# Predict probabilities
predictions <- predict(model, newdata = test_set, type = "response")


```

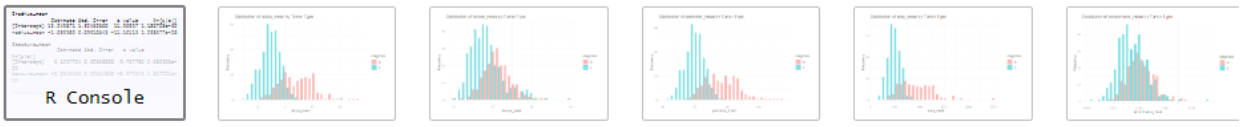
```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```{r}
ggplot(Breastcancer_data2, aes(x = perimeter_mean, fill = diagnosis)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot of Perimeter Mean",
       x = "Perimeter Mean",
       y = "Density") +
  theme_minimal()
```

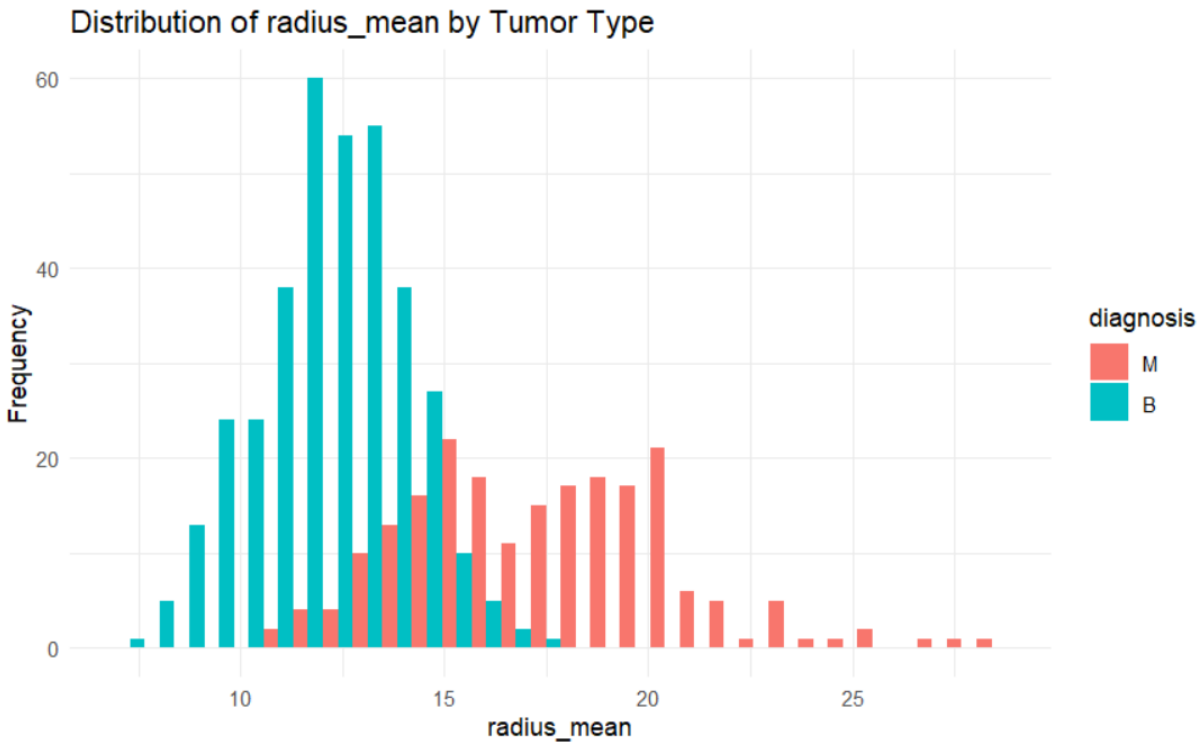


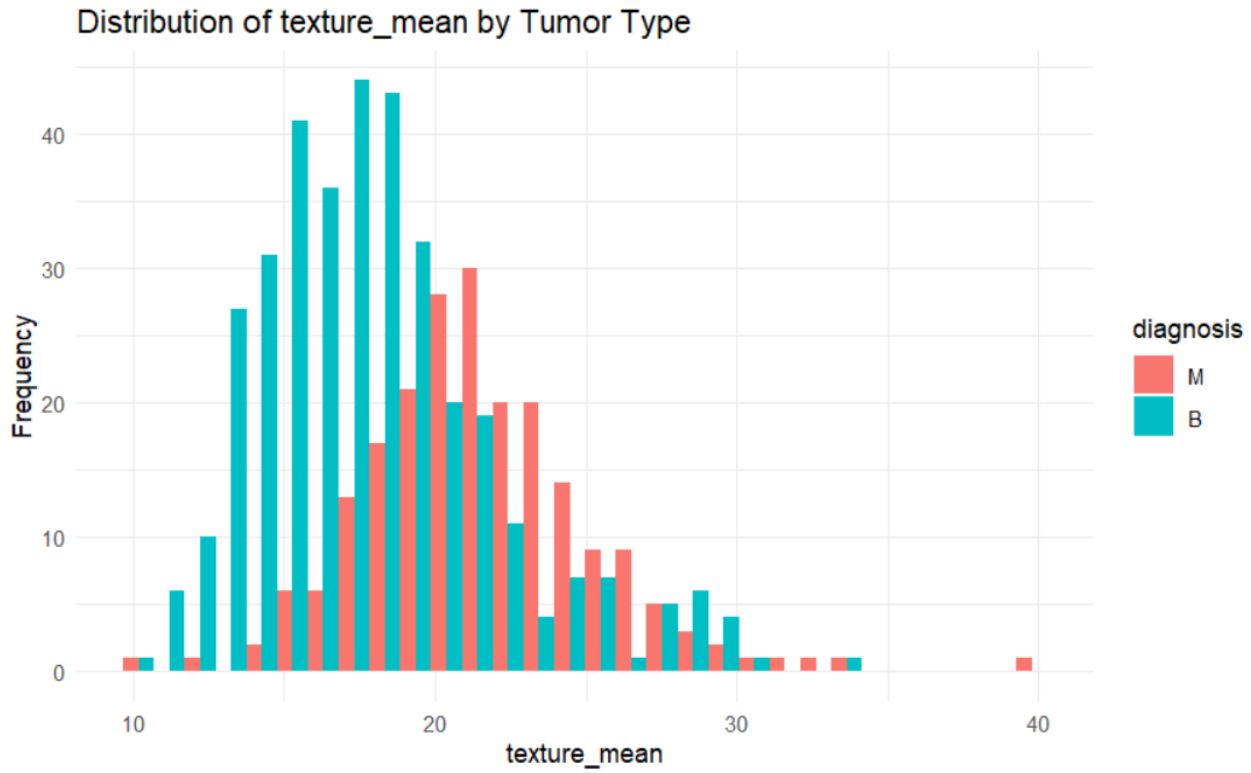

```{r}
# Read and clean data
BreastCancerData <- read.csv("C:\\Users\\sunee\\Desktop\\data.csv")
BreastCancerData <- BreastCancerData %>%
  clean_names() %>%
  distinct() %>%
  select(-starts_with("x")) %>%
  mutate(diagnosis = factor(diagnosis, levels = c("M", "B")))

# Perform logistic regression for each feature and summarize results
logistic_results <- lapply(features, function(feature) {
  model <- glm(as.formula(paste("diagnosis ~", feature)), family = binomial, data = BreastCancerData)
  summary(model)$coefficients
})

# Print results
names(logistic_results) <- features
print(logistic_results)

# Visualizations for each feature comparing malignant and benign
plots <- lapply(features, function(feature) {
  ggplot(BreastCancerData, aes_string(x = feature, fill = "diagnosis")) +
    geom_histogram(position = "dodge", bins = 30) +
    labs(title = paste("Distribution of", feature, "by Tumor Type"), x = feature, y = "Frequency") +
    theme_minimal()
})
print(plots)

```



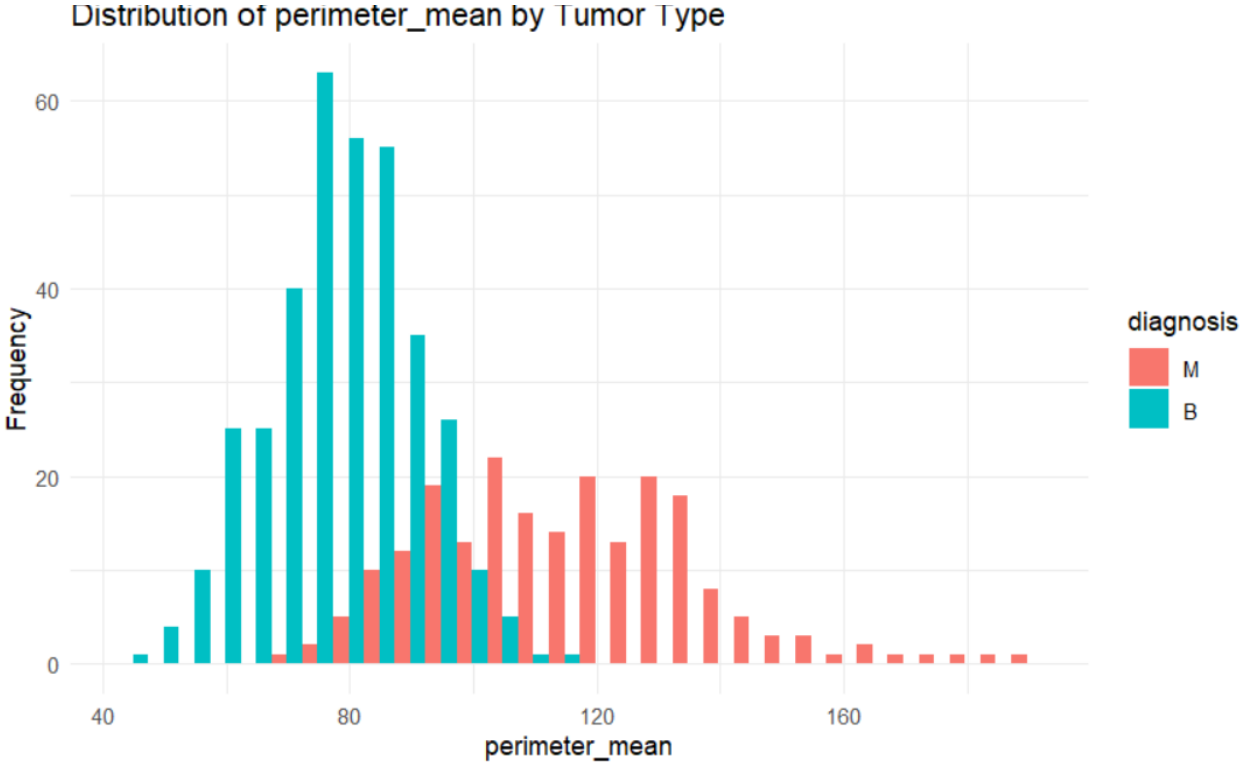


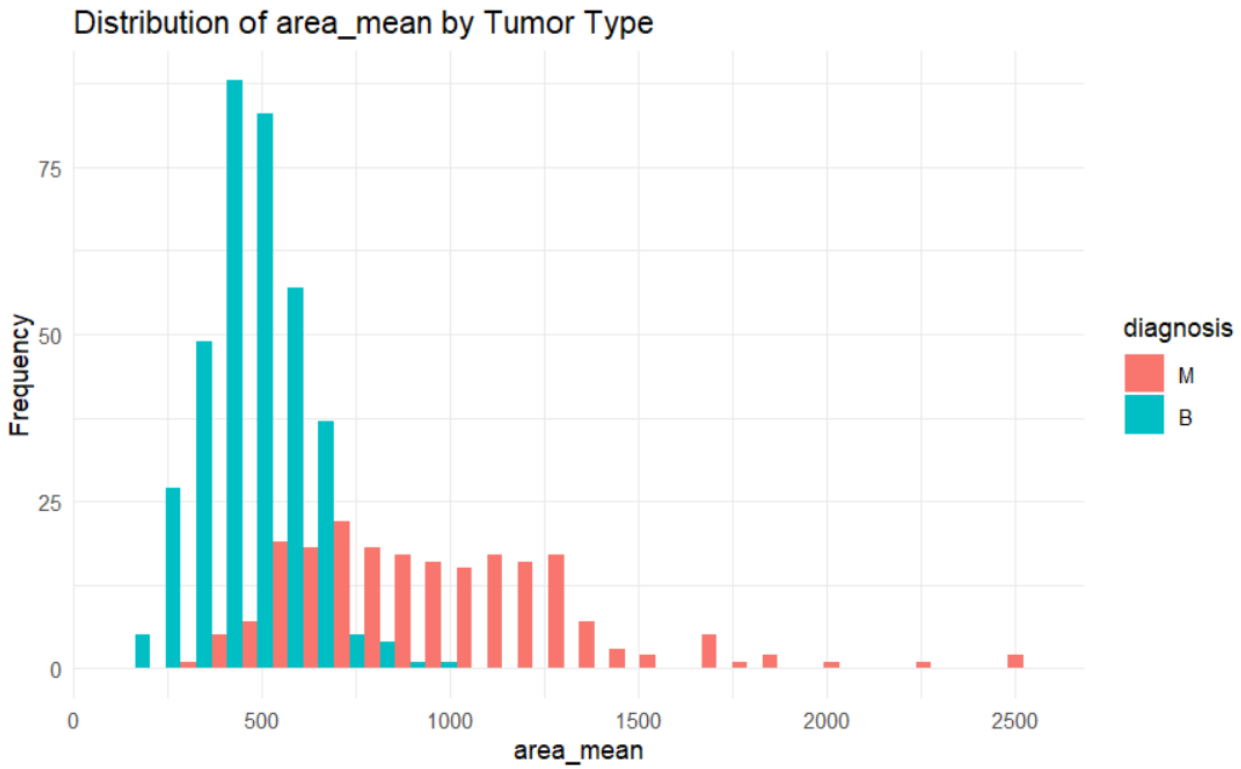

```
texture_mean -0.2346406 0.02614308 -8.975245 2.827221e-19

$perimeter_mean
                 Estimate Std. Error   z value     Pr(>|z|)
(Intercept)    15.7133279 1.37475538  11.42991 2.964217e-30
perimeter_mean -0.1639859 0.01485368 -11.04009 2.447935e-28

$area_mean
               Estimate  Std. Error   z value     Pr(>|z|)
(Intercept)  7.97409315 0.682863094  11.67744 1.662256e-31
area_mean   -0.01176793 0.001089694 -10.79929 3.468757e-27

$smoothness_mean
                  Estimate Std. Error   z value     Pr(>|z|)
(Intercept)       6.377306   0.747421  8.532415 1.433238e-17
smoothness_mean -60.085710   7.549692 -7.958697 1.738602e-15

[[1]]

[[2]]

[[3]]

[[4]]

[[5]]
```

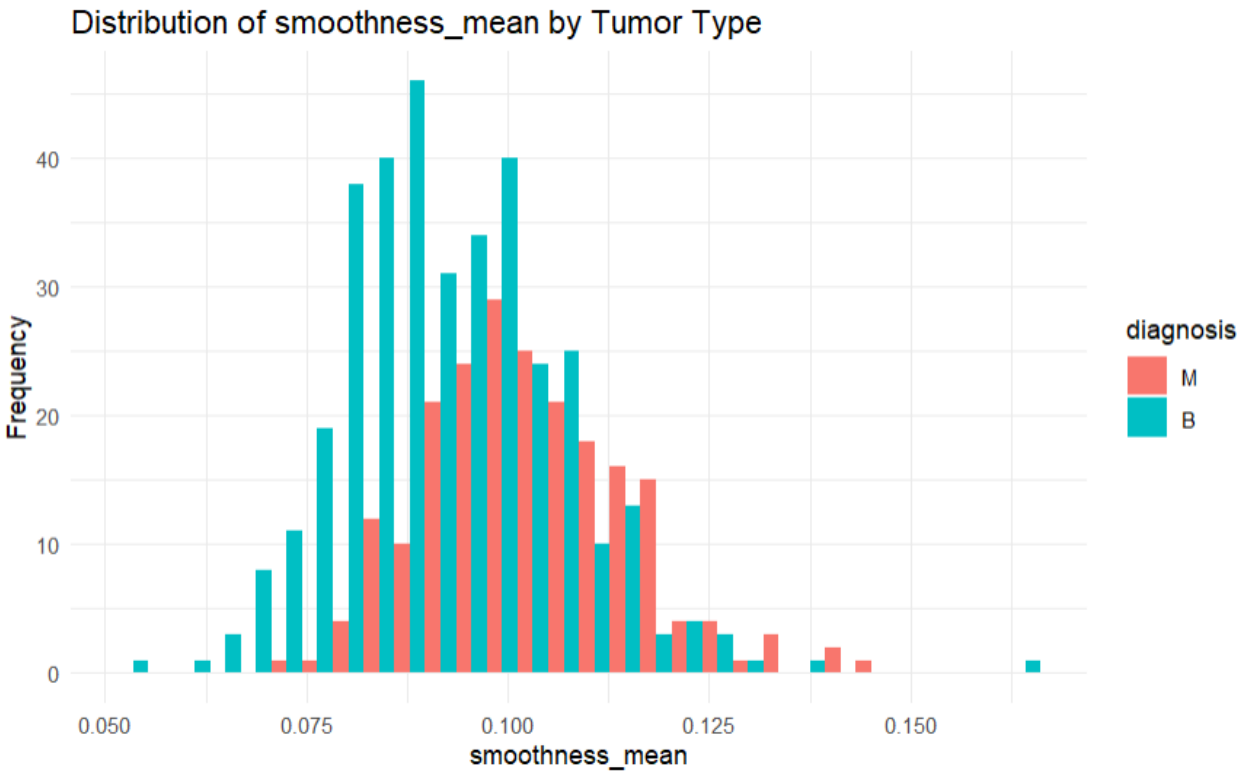



Distribution of texture_mean by Tumor Type
Frequency
0
10
20
30
40
10
20
30
40
texture_mean
diagnosis
M
B

Chunk 54


Distribution of perimeter_mean by Tumor Type
Frequency
0
20
40
60
40
80
120
160
perimeter_mean
diagnosis
M
B


Distribution of area_mean by Tumor Type
Frequency
0
25
50
75
0
500
1000
1500
2000
2500
area_mean
diagnosis
M
B


Distribution of smoothness_mean by Tumor Type
Frequency
0
10
20
30
40
0.050
0.075
0.100
0.125
0.150
smoothness_mean
diagnosis
M
B

```{r}
# Check unique values of the diagnosis column
unique_values <- unique(BreastCancerData$diagnosis)
print(unique_values)

# Check for missing values in the diagnosis column
missing_values <- sum(is.na(BreastCancerData$diagnosis))
print(missing_values)

```

```
 [1] M B
 Levels: M B
 [1] 0
```