DS-6030 Homework Module 3

Tom Lever

06/08/2023

DS 6030 | Spring 2023 | University of Virginia

14. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the `Auto` data set.

    (a) Create a binary variable, `mpg01`, that contains a 1 if `mpg` contains a value above its median, and a 0 if `mpg` contains a value below its median. You can compute the median using the `median()` function. Note you may find it helpful to use the `data.frame()` function to create a single data set containing both `mpg01` and the other `Auto` variables.

```
head(Auto, n = 3)
```

```
#   mpg cylinders displacement horsepower weight acceleration year origin
# 1  18         8          307        130   3504         12.0   70      1
# 2  15         8          350        165   3693         11.5   70      1
# 3  18         8          318        150   3436         11.0   70      1
#                        name
# 1 chevrolet chevelle malibu
# 2         buick skylark 320
# 3        plymouth satellite
```

```
median_fuel_efficiency <- median(Auto$mpg)
number_of_observations <- nrow(Auto)
mpg01 <- rep(0, number_of_observations)
condition <- Auto$mpg > median_fuel_efficiency
mpg01[condition] <- 1
mpg01 <- factor(mpg01)
data_frame <- data.frame(Auto, mpg01)
```

    (b) Explore the data graphically in order to investigate the association between `mpg01` and the other features. Which of the other features seem most likely to be useful in predicting `mpg01`? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

    Response $mpg01$ has a high positive correlation with mpg, high negative correlations with $cylinders$, $displacement$, and $weight$, a moderate positive correlation with $origin$, a moderate negative correlation with $horsepower$, and low positive correlations with $acceleration$ and $year$. mpg, $cylinders$, $displacement$, and $weight$ seem most likely to be useful in predicting $mpg01$.

    According to scatterplots, boxplots, and a barplot, obviously, all low fuel efficiencies correspond to $mpg01 = 0$ and all high fuel efficiencies correspond to $mpg01 = 1$. All non-outlying high fuel efficiencies correspond to the first quarter of number of cylinders and to four cylinders. High displacements correspond to low fuel efficiency; all non-outlying high fuel efficiencies correspond to the first quarter of displacements. High horsepowers correspond to low fuel efficiency; all non-outlying high fuel efficiencies correspond to the first half of horsepowers. High weights correspond to low fuel efficiency; all non-outlying high fuel efficiencies correspond to the first half of weights. Low accelerations correspond to low fuel efficiency; the first quartile of accelerations corresponding to high fuel efficiencies is greater than the median acceleration corresponding to low fuel efficiencies. The first quartile of years corresponding to high fuel efficiencies is higher than the median year corresponding to low fuel efficiencies; fuel efficiency increases with year. A supermajority of American automobiles have low fuel efficiencies. A supermajority of European automobiles have high fuel efficiencies. A supermajority of Japanese automobiles have high fuel efficiencies. A supermajority of automobiles with low fuel efficiency are American. Automobiles with high fuel efficiency are evenly distributed across American, European, and Japanese origin. There are about 3 times as many American automobiles as European automobiles and about 3 times as many American automobiles as Japanese automobiles.

```r
library(TomLeversRPackage)
index_of_column_name <- get_index_of_column_of_data_frame(data_frame, "name")
data_frame_of_columns_except_name <- data_frame[, -index_of_column_name]
correlation_matrix <- cor(data_frame_of_columns_except_name)
```

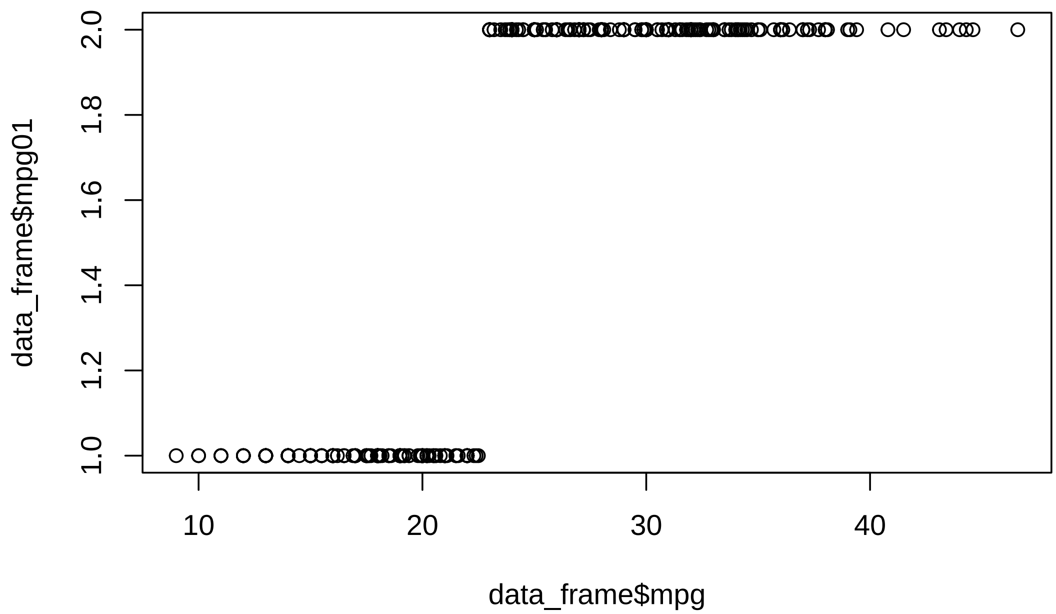
```
# Error in cor(data_frame_of_columns_except_name): 'x' must be numeric
```

```r
analyze_correlation_matrix(correlation_matrix)
```

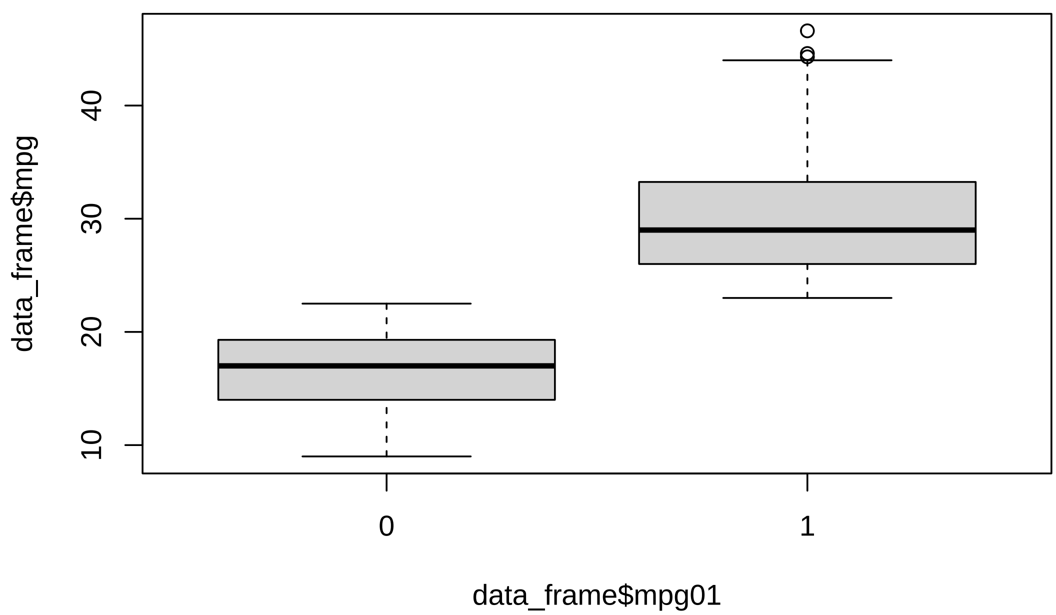
```
# Year
#     V+:  Year
#     V-:
#     H+:  Volume
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Lag1, Lag2, Lag3, Lag4, Lag5, Today
# Lag1
#     V+:  Lag1
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag2, Lag3, Lag4, Lag5, Volume, Today
# Lag2
#     V+:  Lag2
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag3, Lag4, Lag5, Volume, Today
# Lag3
#     V+:  Lag3
#     V-:
```

```
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag4, Lag5, Volume, Today
# Lag4
#     V+:  Lag4
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag3, Lag5, Volume, Today
# Lag5
#     V+:  Lag5
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag3, Lag4, Volume, Today
# Volume
#     V+:  Volume
#     V-:
#     H+:  Year
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Lag1, Lag2, Lag3, Lag4, Lag5, Today
# Today
#     V+:  Today
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag3, Lag4, Lag5, Volume
```

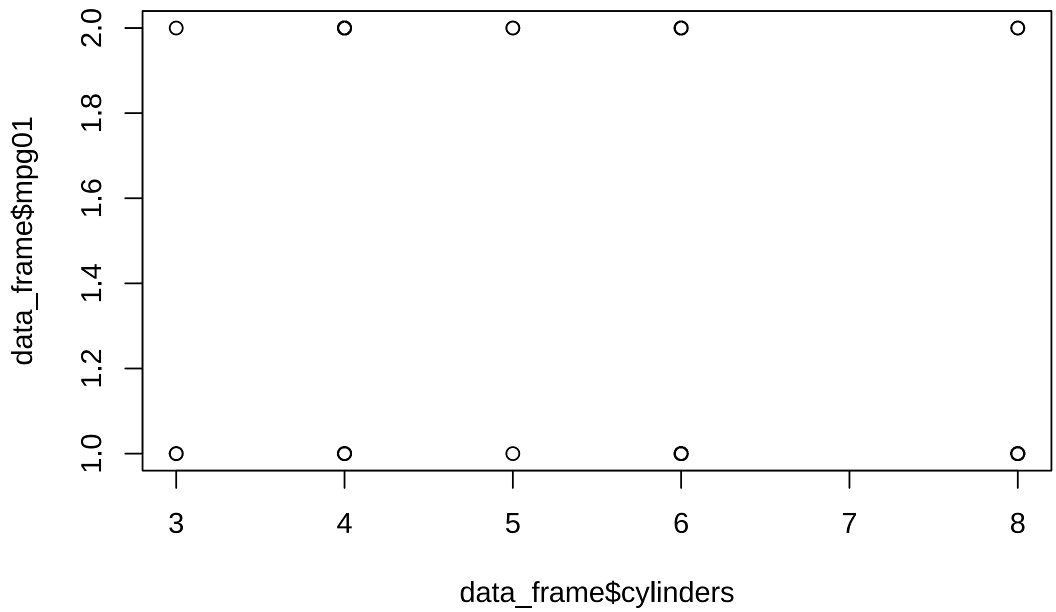
```
plot(x = data_frame$mpg, y = data_frame$mpg01)
```



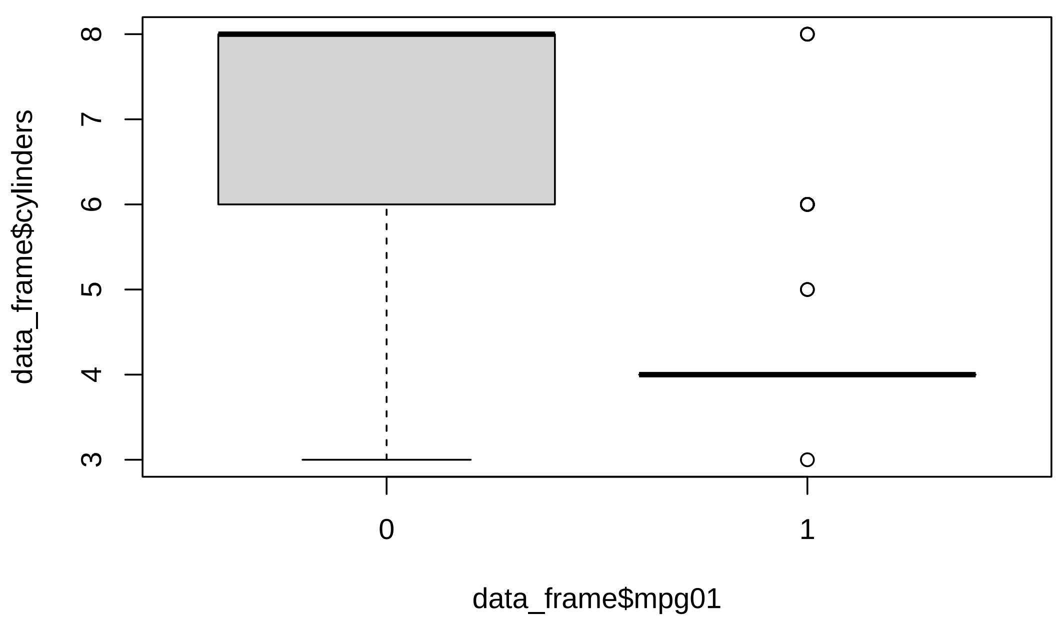

```
boxplot(data_frame$mpg ~ data_frame$mpg01, data = data_frame)
```



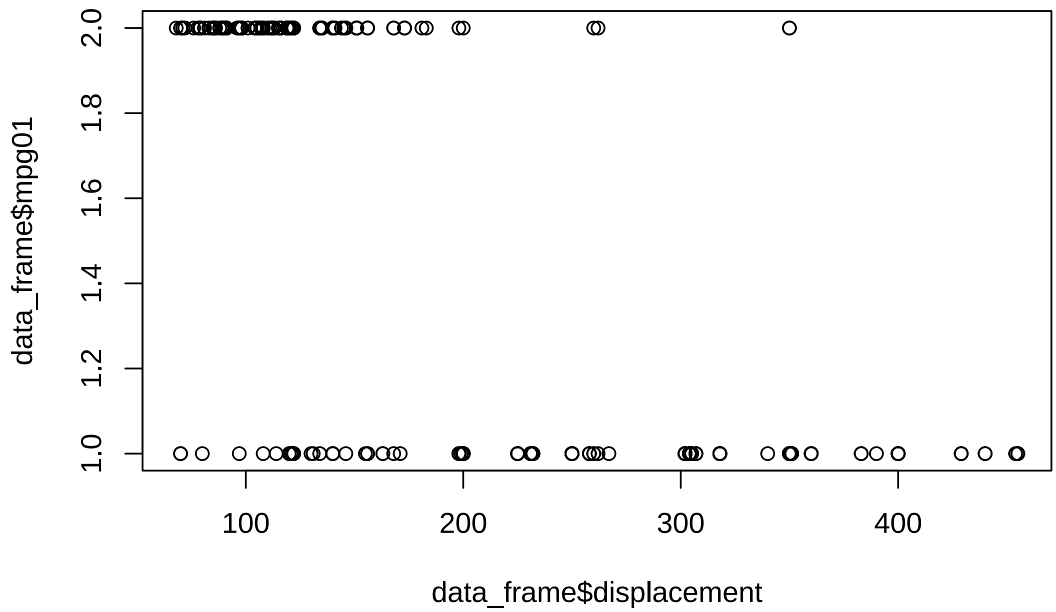
```r
plot(x = data_frame$cylinders, y = data_frame$mpg01)
```



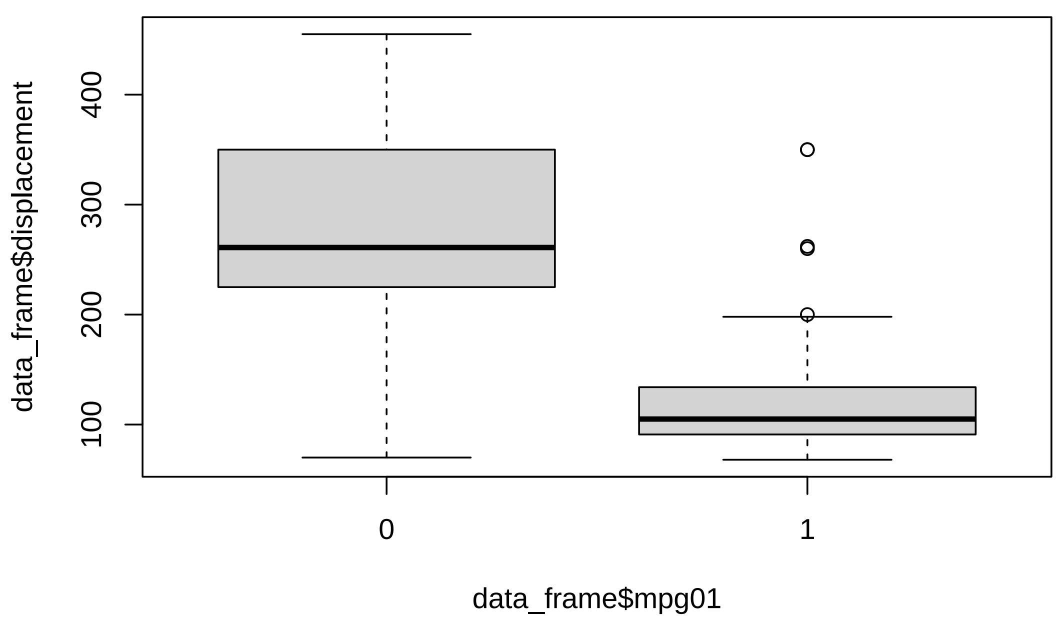

```r
boxplot(data_frame$cylinders ~ data_frame$mpg01, data = data_frame)
```



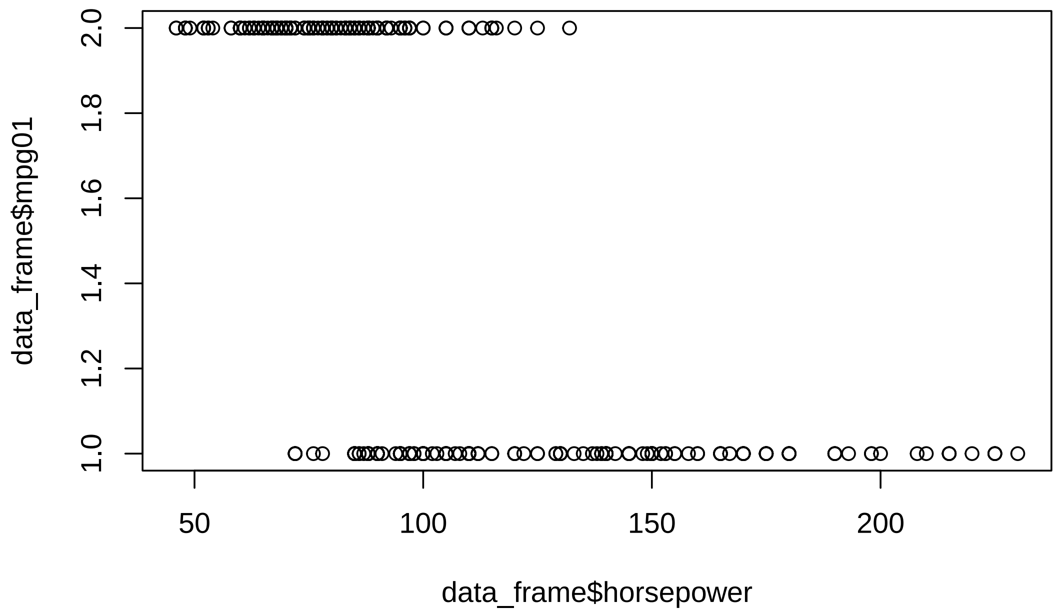
```r
plot(x = data_frame$displacement, y = data_frame$mpg01)
```



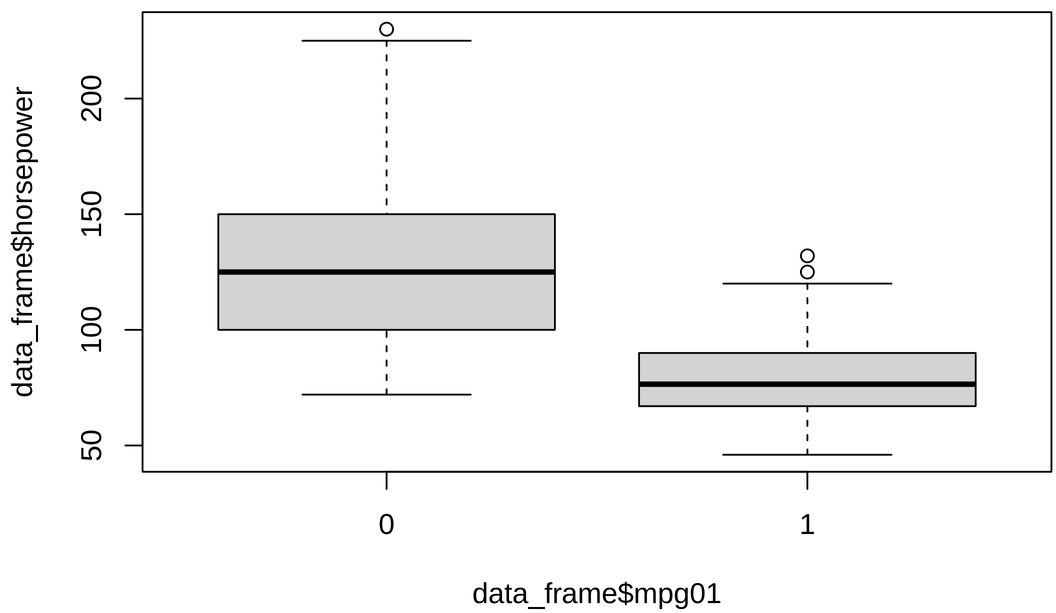

```r
boxplot(data_frame$displacement ~ data_frame$mpg01, data = data_frame)
```



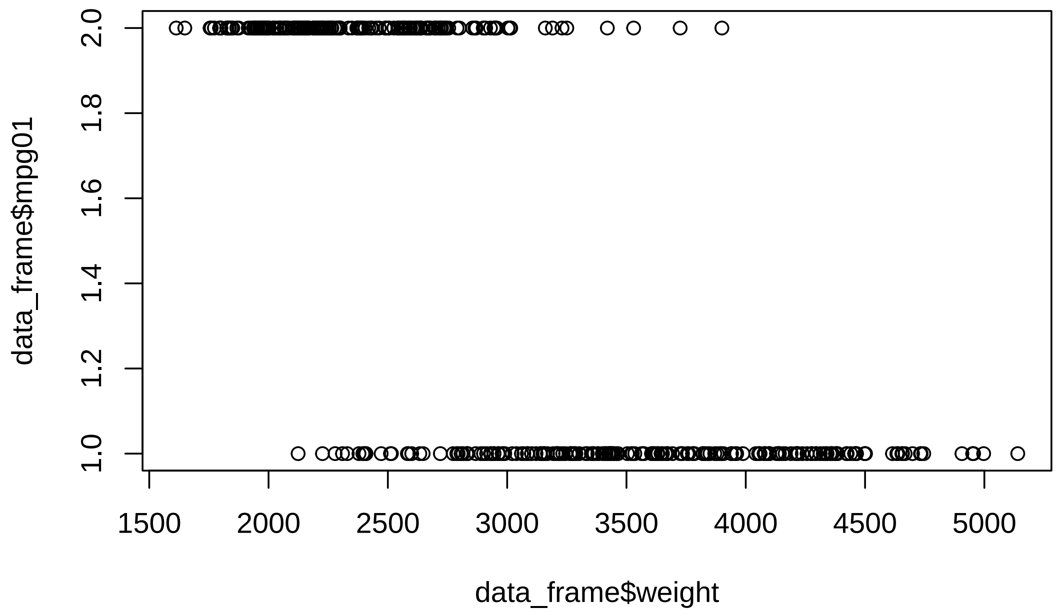
```r
plot(x = data_frame$horsepower, y = data_frame$mpg01)
```



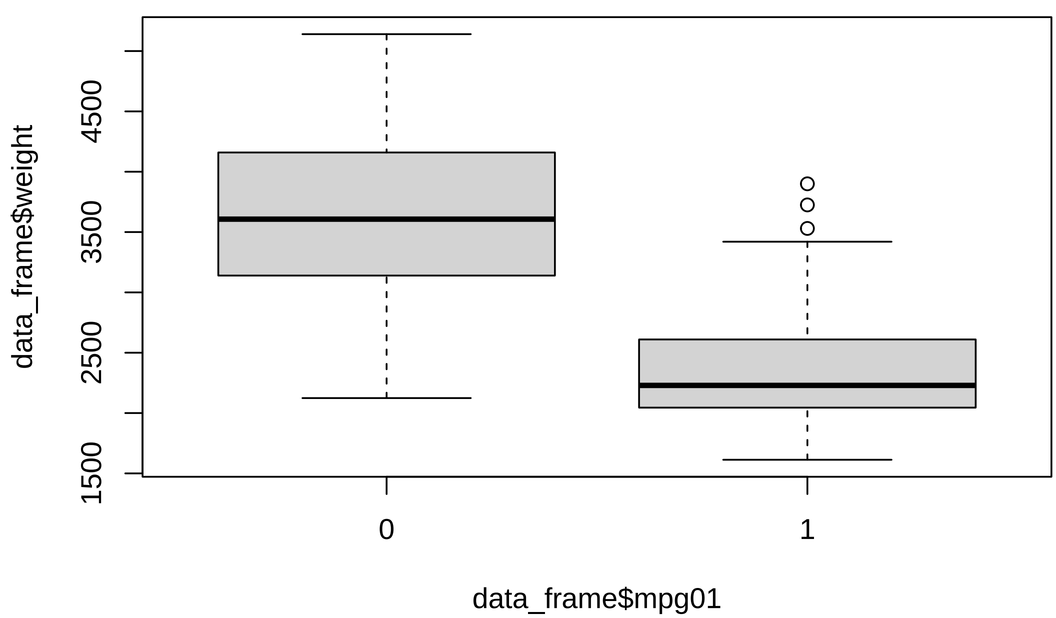

```r
boxplot(data_frame$horsepower ~ data_frame$mpg01, data = data_frame)
```



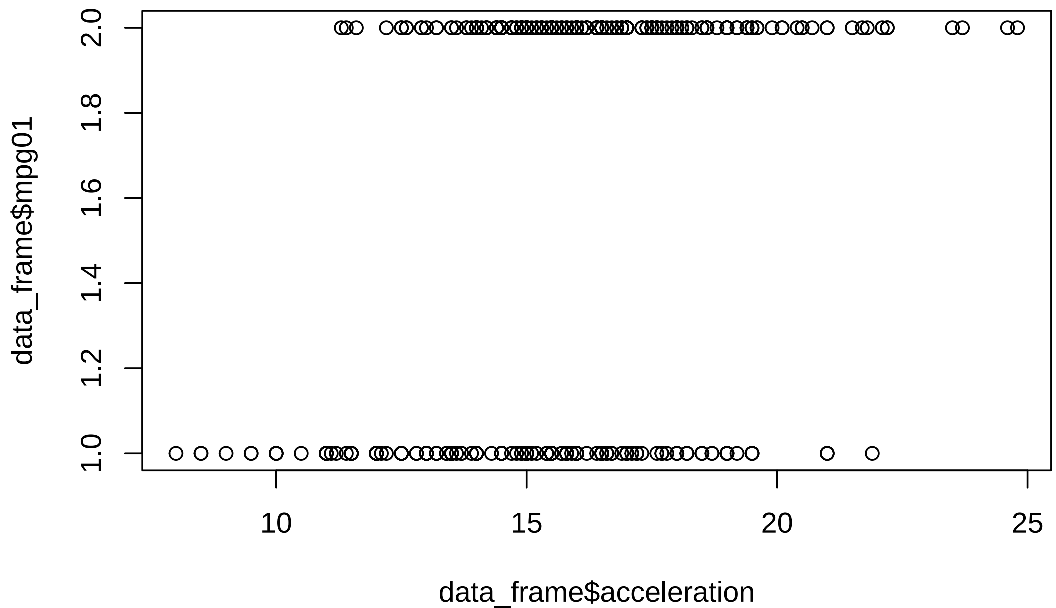
```r
plot(x = data_frame$weight, y = data_frame$mpg01)
```



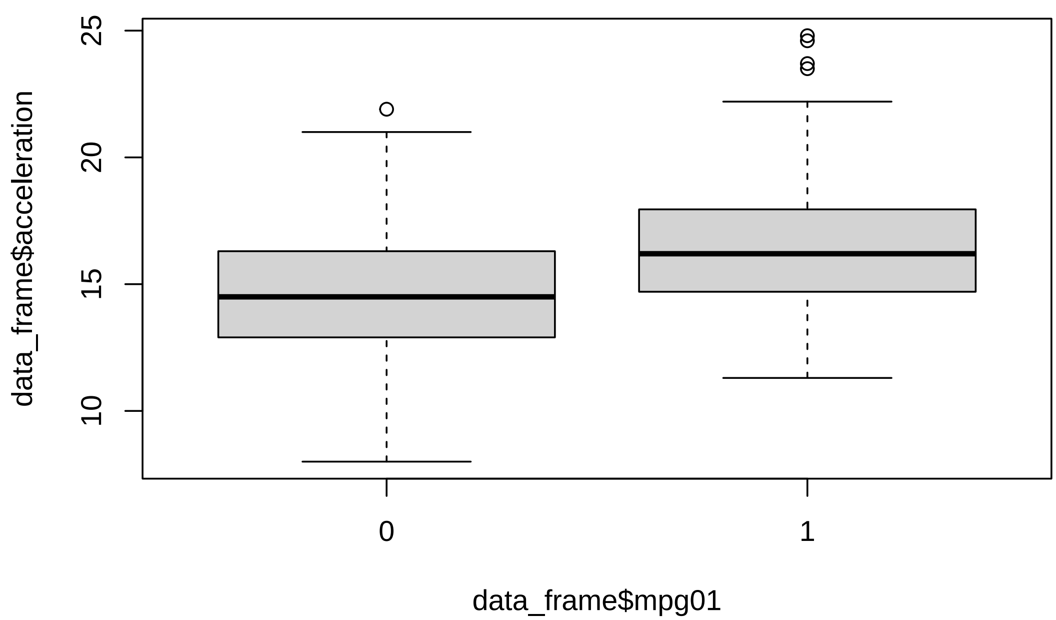

```r
boxplot(data_frame$weight ~ data_frame$mpg01, data = data_frame)
```



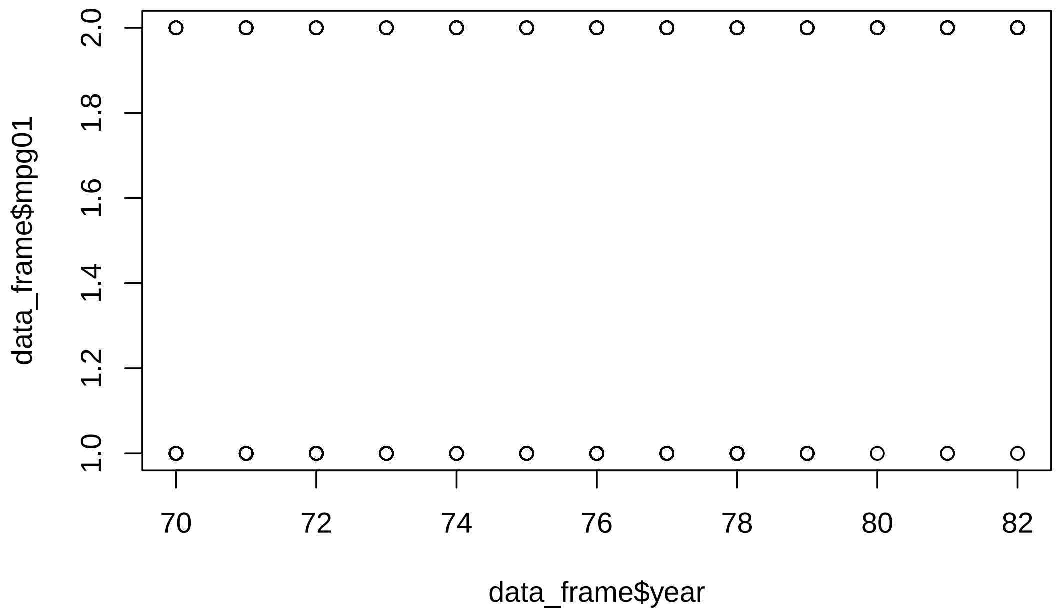
```
plot(x = data_frame$acceleration, y = data_frame$mpg01)
```



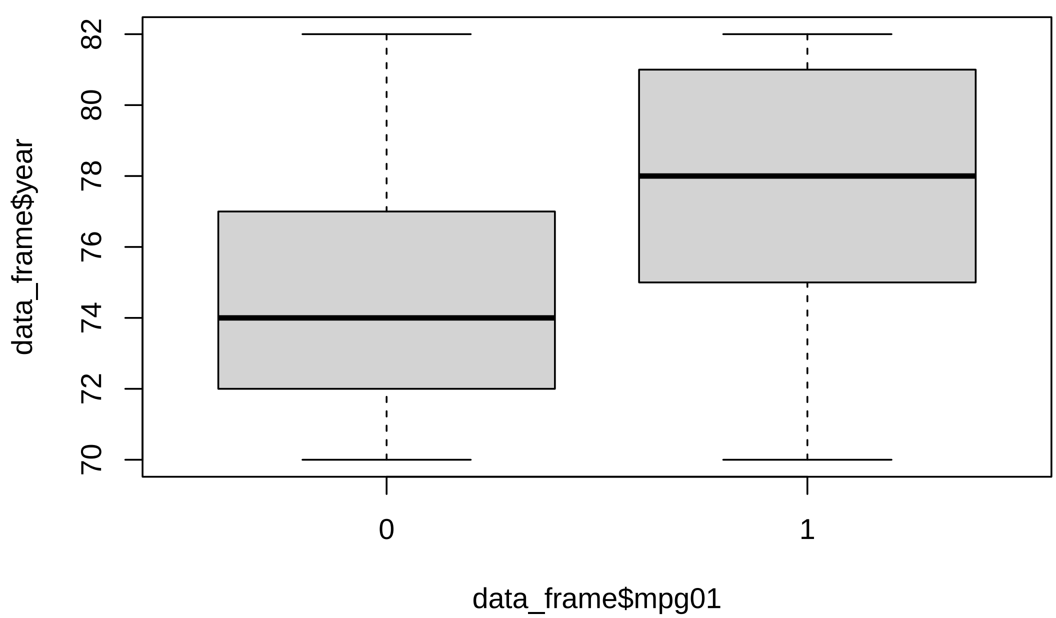

```
boxplot(data_frame$acceleration ~ data_frame$mpg01, data = data_frame)
```



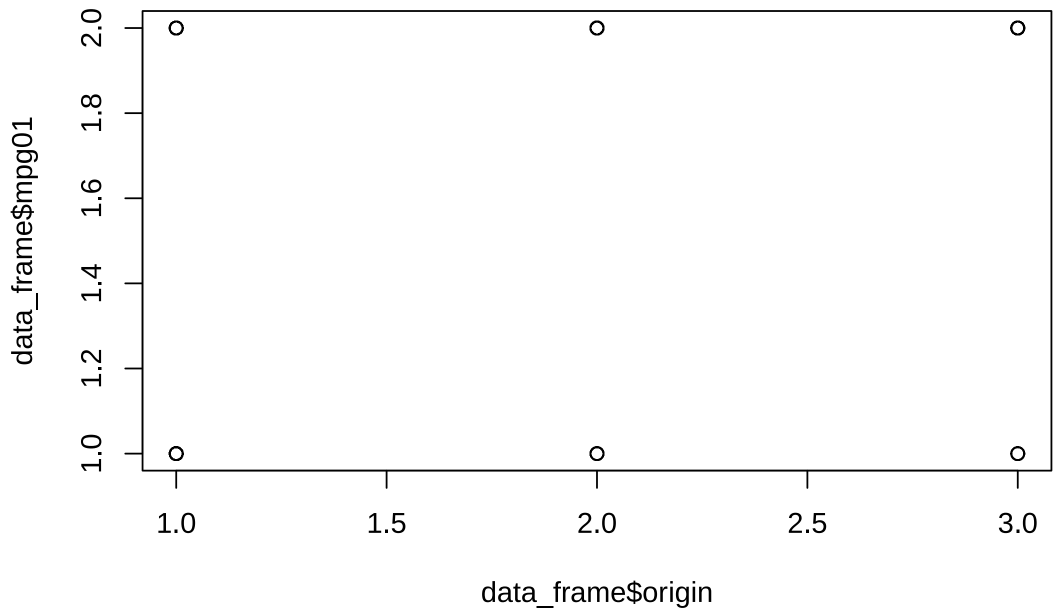
```r
plot(x = data_frame$year, y = data_frame$mpg01)
```



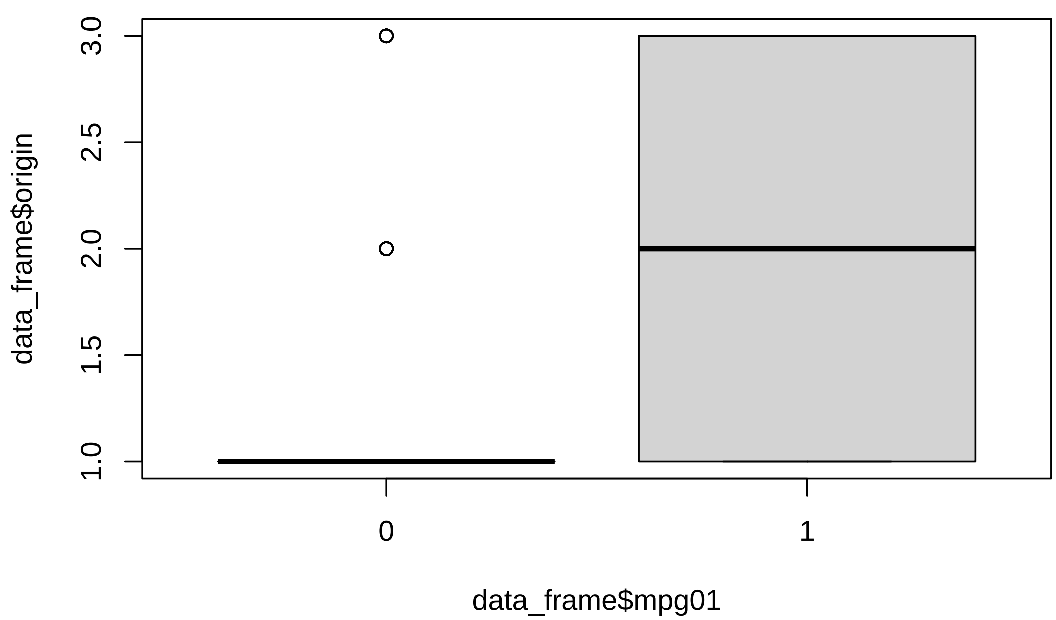

```r
boxplot(data_frame$year ~ data_frame$mpg01, data = data_frame)
```



```r
plot(x = data_frame$origin, y = data_frame$mpg01)
```




```r
boxplot(data_frame$origin ~ data_frame$mpg01, data = data_frame)
```



```r
table_of_origins_and_indicators_of_fuel_efficiency <- table(
    data_frame$origin,
    data_frame$mpg01
)
table_of_proportions_split_by_rows <- prop.table(
    x = table_of_origins_and_indicators_of_fuel_efficiency,
    margin = 1
)
table_of_percents <- table_of_proportions_split_by_rows * 100
table_of_rounded_percents <- round(table_of_percents, 2)
table_of_rounded_percents
```

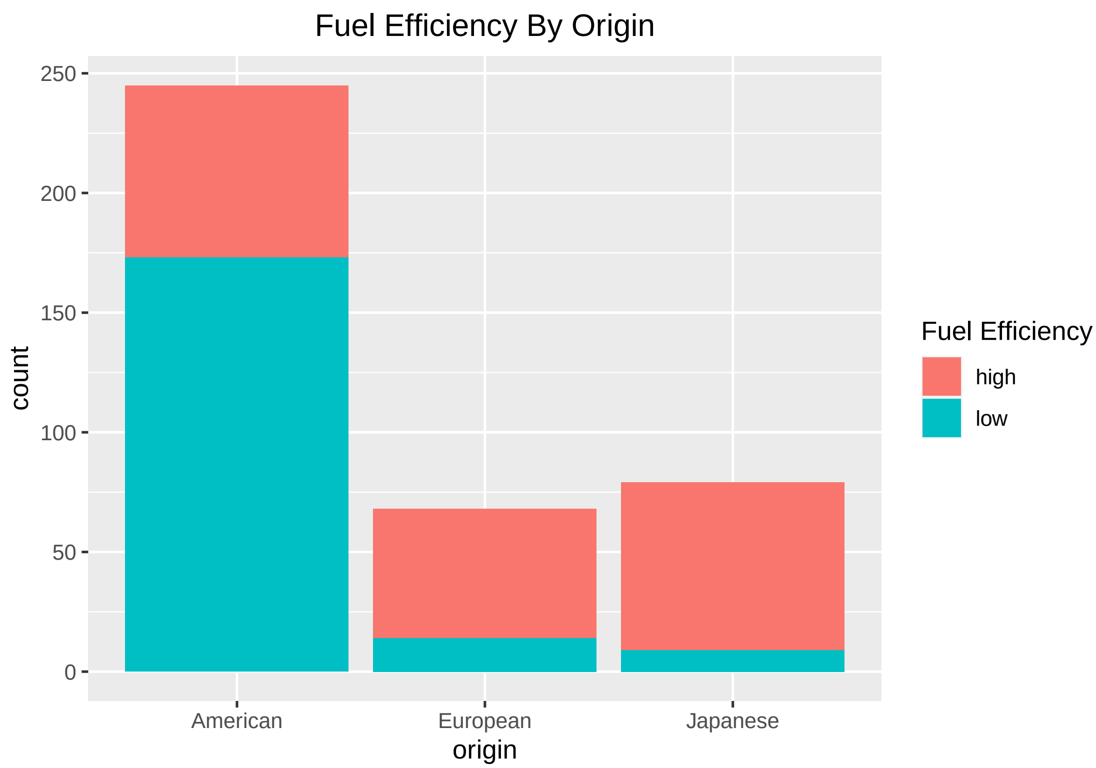
```
#
#         0     1
#   1 70.61 29.39
#   2 20.59 79.41
#   3 11.39 88.61
```

```r
table_of_proportions_split_by_rows <- prop.table(
    x = table_of_origins_and_indicators_of_fuel_efficiency,
    margin = 2
)
table_of_percents <- table_of_proportions_split_by_rows * 100
table_of_rounded_percents <- round(table_of_percents, 2)
table_of_rounded_percents
```

```
#
#         0     1
#   1 88.27 36.73
#   2  7.14 27.55
#   3  4.59 35.71
```

```r
vector_of_English_indicators_of_origin <-
    rep("American", number_of_observations)
condition <- data_frame$origin == 2
vector_of_English_indicators_of_origin[condition] <- "European"
condition <- data_frame$origin == 3
vector_of_English_indicators_of_origin[condition] <- "Japanese"
vector_of_English_indicators_of_fuel_efficiency <-
    rep("low", number_of_observations)
condition <- data_frame$mpg01 == 1
vector_of_English_indicators_of_fuel_efficiency[condition] <- "high"
library(ggplot2)
ggplot(
    data = data_frame,
    mapping = aes(
        x = vector_of_English_indicators_of_origin,
        fill = vector_of_English_indicators_of_fuel_efficiency
    )
) +
    geom_bar(position = "stack") +
    labs(
        x = "origin",
        y = "count",
```

```
        title = "Fuel Efficiency By Origin"
    ) +
     theme(
         plot.title = element_text(hjust = 0.5)
     ) +
     guides(
         fill = guide_legend(
             title = "Fuel Efficiency"
         )
     )
```



(c) Split the data into a training set and a test set.

```
set.seed(1)
vector_of_random_indices <- sample(1:number_of_observations)
shuffled_data_frame <- data_frame[vector_of_random_indices, ]
number_of_training_data <- 392 / 14 * 12
training_data <- shuffled_data_frame[1:number_of_training_data, ]
testing_data <-
    shuffled_data_frame[(number_of_training_data + 1) : number_of_observations, ]
```

(d) Perform LDA on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

```
generate_summary_of_performance(
    type_of_model = "LDA",
    formula = mpg01 ~ cylinders + displacement + weight,
    training_data = training_data,
    test_data = testing_data
)
```

```
# $confusion_matrix
#
# vector_of_predicted_directions  0  1
#                               0 26  0
#                               1  5 25
#
# $decimal_of_correct_predictions
# [1] 0.9107143
#
# $error_rate
# [1] 0.08928571
#
# $fraction_of_correct_predictions
# [1] "51 / 56"
#
# $map_of_binary_value_to_response_value
#   1
# 0 0
# 1 1
#
# $equation_for_number_of_true_negatives
# [1] "TN = CM[1, 1] = 26"
#
# $equation_for_number_of_false_negatives
# [1] "FN = CM[1, 2] = 0"
#
# $equation_for_number_of_false_positives
# [1] "FP = CM[2, 1] = 5"
#
# $equation_for_number_of_true_positives
# [1] "TP = CM[2, 2] = 25"
#
# $equation_for_true_positive_rate
# [1] "TPR = Sensitivity = Recall = Hit Rate = TP/P = TP/(TP+FN) = 1"
#
# $equation_for_true_negative_rate
# [1] "TNR = Specificity = Selectivity = TN/N = TN/(TN+FP) = 0.838709677419355"
#
# attr(,"class")
# [1] "summary_of_performance"
```

The test error rate of our LDA model with formula $mpg01 \sim cylinders + displacement + weight$ is 8.9 percent.

(e) Perform QDA on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

```r
generate_summary_of_performance(
    type_of_model = "QDA",
    formula = mpg01 ~ cylinders + displacement + weight,
    training_data = training_data,
    test_data = testing_data
)
```

```
# $confusion_matrix
#
```

```
# vector_of_predicted_directions  0  1
#                               0 26  1
#                               1  5 24
#
# $decimal_of_correct_predictions
# [1] 0.8928571
#
# $error_rate
# [1] 0.1071429
#
# $fraction_of_correct_predictions
# [1] "50 / 56"
#
# $map_of_binary_value_to_response_value
#   1
# 0 0
# 1 1
#
# $equation_for_number_of_true_negatives
# [1] "TN = CM[1, 1] = 26"
#
# $equation_for_number_of_false_negatives
# [1] "FN = CM[1, 2] = 1"
#
# $equation_for_number_of_false_positives
# [1] "FP = CM[2, 1] = 5"
#
# $equation_for_number_of_true_positives
# [1] "TP = CM[2, 2] = 24"
#
# $equation_for_true_positive_rate
# [1] "TPR = Sensitivity = Recall = Hit Rate = TP/P = TP/(TP+FN) = 0.96"
#
# $equation_for_true_negative_rate
# [1] "TNR = Specificity = Selectivity = TN/N = TN/(TN+FP) = 0.838709677419355"
#
# attr(,"class")
# [1] "summary_of_performance"
```

The test error rate of our QDA model with formula $mpg01 \sim cylinders + displacement + weight$ is 10.7 percent.

(f) Perform logistic regression on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

```
generate_summary_of_performance(
    type_of_model = "LR",
    formula = mpg01 ~ cylinders + displacement + weight,
    training_data = training_data,
    test_data = testing_data
)
```

```
# $confusion_matrix
#
# vector_of_predicted_directions  0  1
#                            Down 25  1
```

```
#                            Up    6 24
#
# $decimal_of_correct_predictions
# [1] 0.875
#
# $error_rate
# [1] 0.125
#
# $fraction_of_correct_predictions
# [1] "49 / 56"
#
# $map_of_binary_value_to_response_value
#   1
# 0 0
# 1 1
#
# $equation_for_number_of_true_negatives
# [1] "TN = CM[1, 1] = 25"
#
# $equation_for_number_of_false_negatives
# [1] "FN = CM[1, 2] = 1"
#
# $equation_for_number_of_false_positives
# [1] "FP = CM[2, 1] = 6"
#
# $equation_for_number_of_true_positives
# [1] "TP = CM[2, 2] = 24"
#
# $equation_for_true_positive_rate
# [1] "TPR = Sensitivity = Recall = Hit Rate = TP/P = TP/(TP+FN) = 0.96"
#
# $equation_for_true_negative_rate
# [1] "TNR = Specificity = Selectivity = TN/N = TN/(TN+FP) = 0.806451612903226"
#
# attr(,"class")
# [1] "summary_of_performance"
```

The test error rate of our LR model with formula $mpg01 \sim cylinders + displacement + weight$ is 12.5 percent.

(g) Perform naive Bayes on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained? (skip this exercise)

(h) Perform KNN on the training data, with several values of K, in order to predict `mpg01`. Use only the variables that seemed most associated with `mpg01` in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

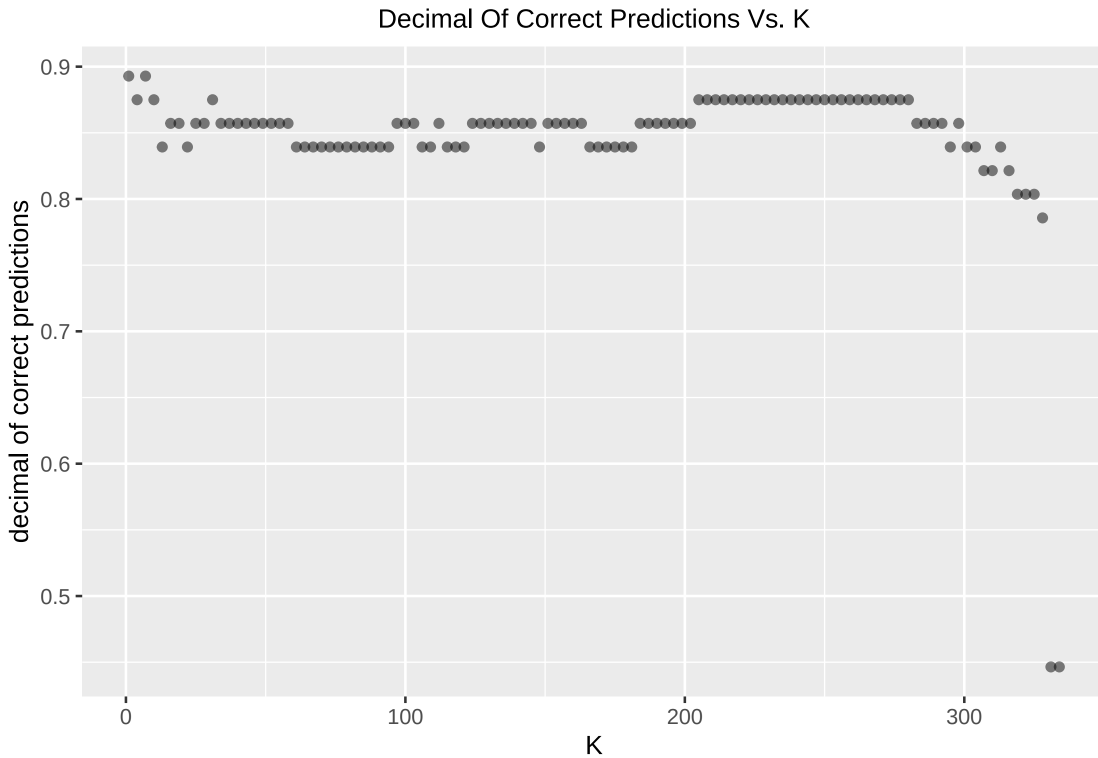
```
generate_summary_of_performance(
    type_of_model = "KNN",
    formula = mpg01 ~ cylinders + displacement + weight,
    training_data = training_data,
    test_data = testing_data
)
```

```
# $confusion_matrix
```

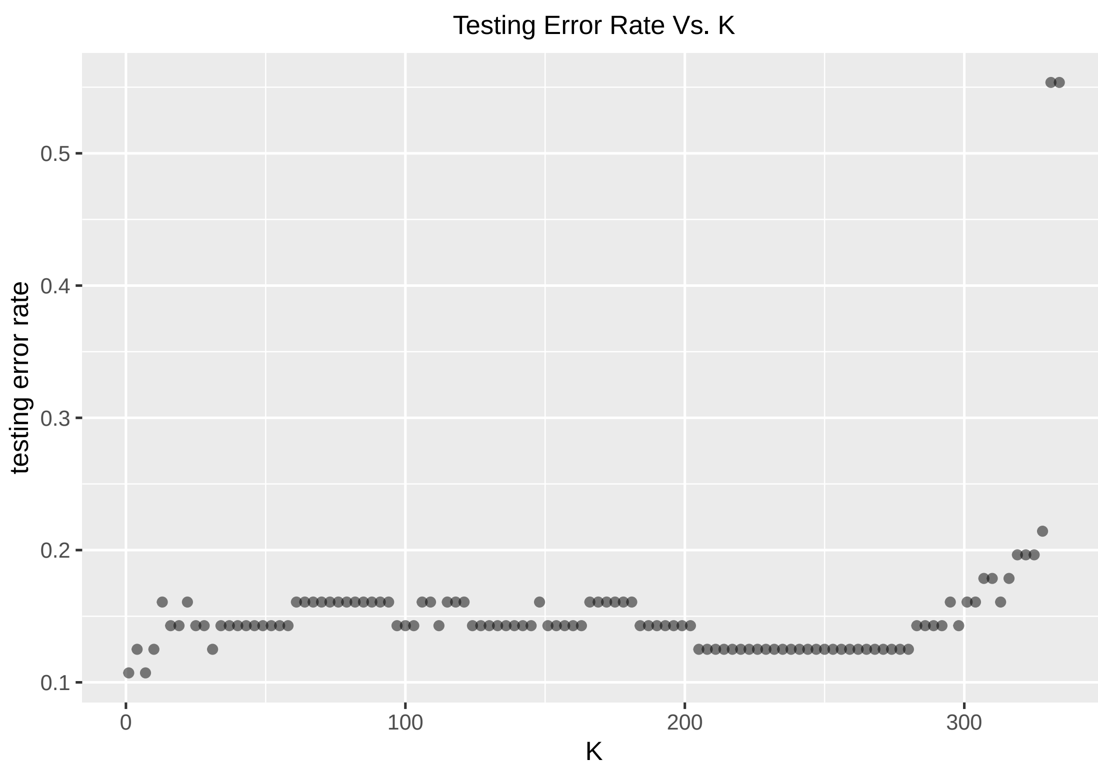
```
#
# vector_of_predicted_directions  0  1
#                               0 26  1
#                               1  5 24
#
# $decimal_of_correct_predictions
# [1] 0.8928571
#
# $error_rate
# [1] 0.1071429
#
# $fraction_of_correct_predictions
# [1] "50 / 56"
#
# $map_of_binary_value_to_response_value
#   1
# 0 0
# 1 1
#
# $equation_for_number_of_true_negatives
# [1] "TN = CM[1, 1] = 26"
#
# $equation_for_number_of_false_negatives
# [1] "FN = CM[1, 2] = 1"
#
# $equation_for_number_of_false_positives
# [1] "FP = CM[2, 1] = 5"
#
# $equation_for_number_of_true_positives
# [1] "TP = CM[2, 2] = 24"
#
# $equation_for_true_positive_rate
# [1] "TPR = Sensitivity = Recall = Hit Rate = TP/P = TP/(TP+FN) = 0.96"
#
# $equation_for_true_negative_rate
# [1] "TNR = Specificity = Selectivity = TN/N = TN/(TN+FP) = 0.838709677419355"
#
# attr(,"class")
# [1] "summary_of_performance"
```

```
optimize_K(
    formula = mpg01 ~ cylinders + displacement + weight,
    training_data = training_data,
    testing_data = testing_data
)
```

```
# $Decimal_Of_Correct_Predictions_Vs_K
```




```
#
# $Error_Rates_Vs_K
```




```
#
```

```
# $vector_of_optimal_values_of_K
# [1] 1 7
#
# attr(,"class")
# [1] "summary_of_optimizing_K"
```

The test error rate of our LDA model with formula $mpg01 \sim cylinders + displacement + weight$ is 12.5 percent.