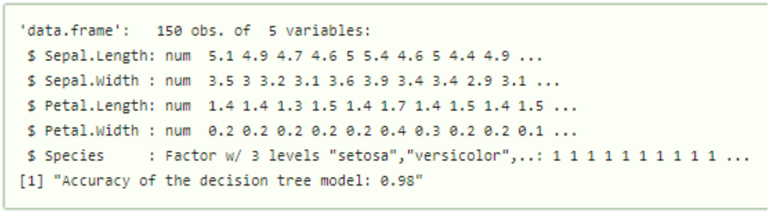
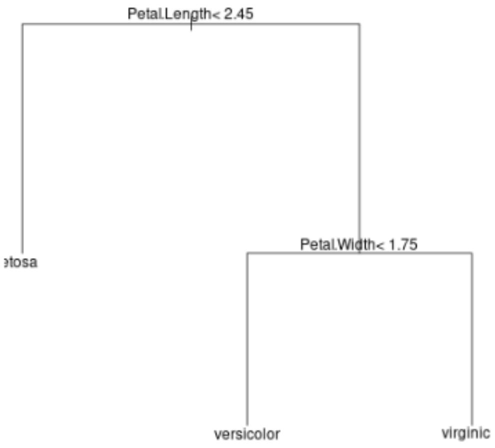
Practical 7 Perform the data classification using classification algorithm.(https://rdrr.io/snippets/)

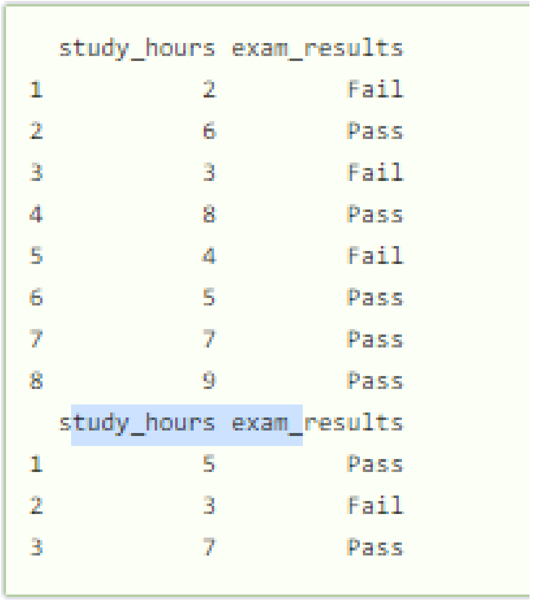
```
# Load the required library
library(rpart)

# Load the Iris dataset
data(iris)
# Check the structure of the dataset
str(iris)

# Split the dataset into training and testing sets
set.seed(123)  # for reproducibility
sample_index <- sample(1:nrow(iris), 0.7*nrow(iris))  # 70% for training, 30% for testing
train_data <- iris[sample_index, ]
test_data <- iris[-sample_index, ]

# Build the decision tree model using the training data
tree_model <- rpart(Species ~ ., data = train_data, method = "class")

# Visualize the decision tree
plot(tree_model)
text(tree_model)

# Make predictions on the testing set
predictions <- predict(tree_model, test_data, type = "class")

# Evaluate the accuracy of the model
accuracy <- mean(predictions == test_data$Species)
print(paste("Accuracy of the decision tree model:", round(accuracy, 2)))
```



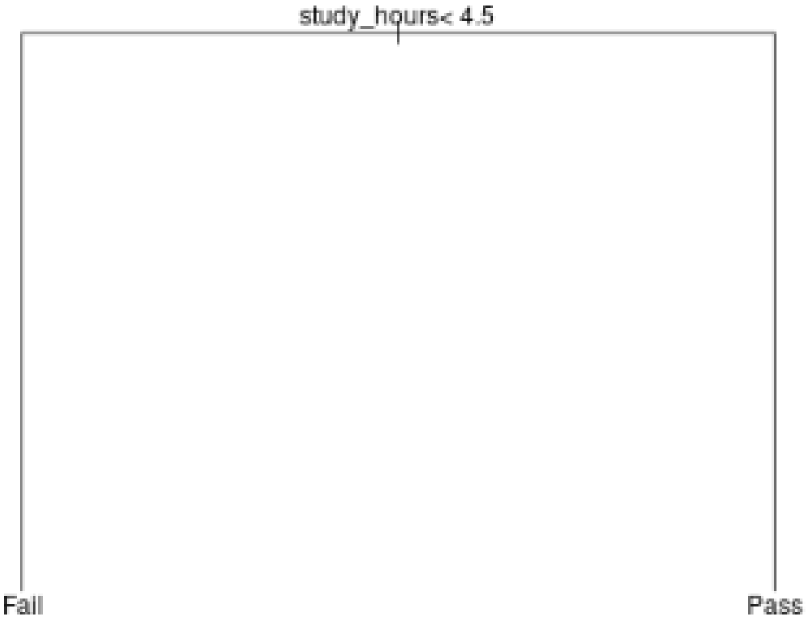
```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
[1] "Accuracy of the decision tree model: 0.98"
```



```
# Create a simple dataset
student_data <- data.frame(
  study_hours = c(2, 6, 3, 8, 4, 5, 7, 9),
  exam_results = c("Fail", "Pass", "Fail", "Pass", "Fail", "Pass", "Pass", "Pass")
)

# Print the dataset
print(student_data)

# Load the required library
library(rpart)

# Build the decision tree model
tree_model <- rpart(exam_results ~ ., data = student_data, method = "class", control = rpart.control(minsplit = 2))

# Visualize the decision tree
plot(tree_model)
text(tree_model)

# Sample predictions (you can input your own data)
new_data <- data.frame(
  study_hours = c(5, 3, 7),
  exam_results = NA
)

# Make predictions on new data
new_data$exam_results <- predict(tree_model, new_data, type = "class")

# Print predictions
print(new_data)
```



```
  study_hours exam_results
1           2         Fail
2           6         Pass
3           3         Fail
4           8         Pass
5           4         Fail
6           5         Pass
7           7         Pass
8           9         Pass
  study_hours exam_results
1           5         Pass
2           3         Fail
3           7         Pass
```


study_hours< 4.5
Fail
Pass