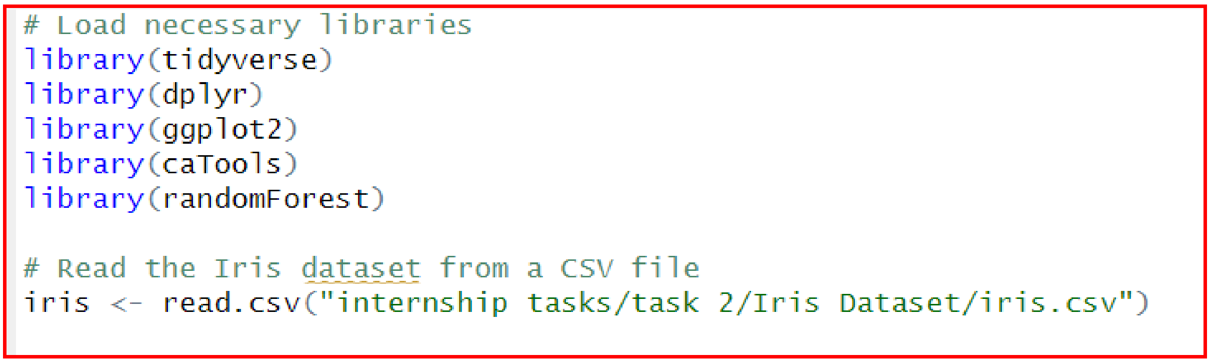
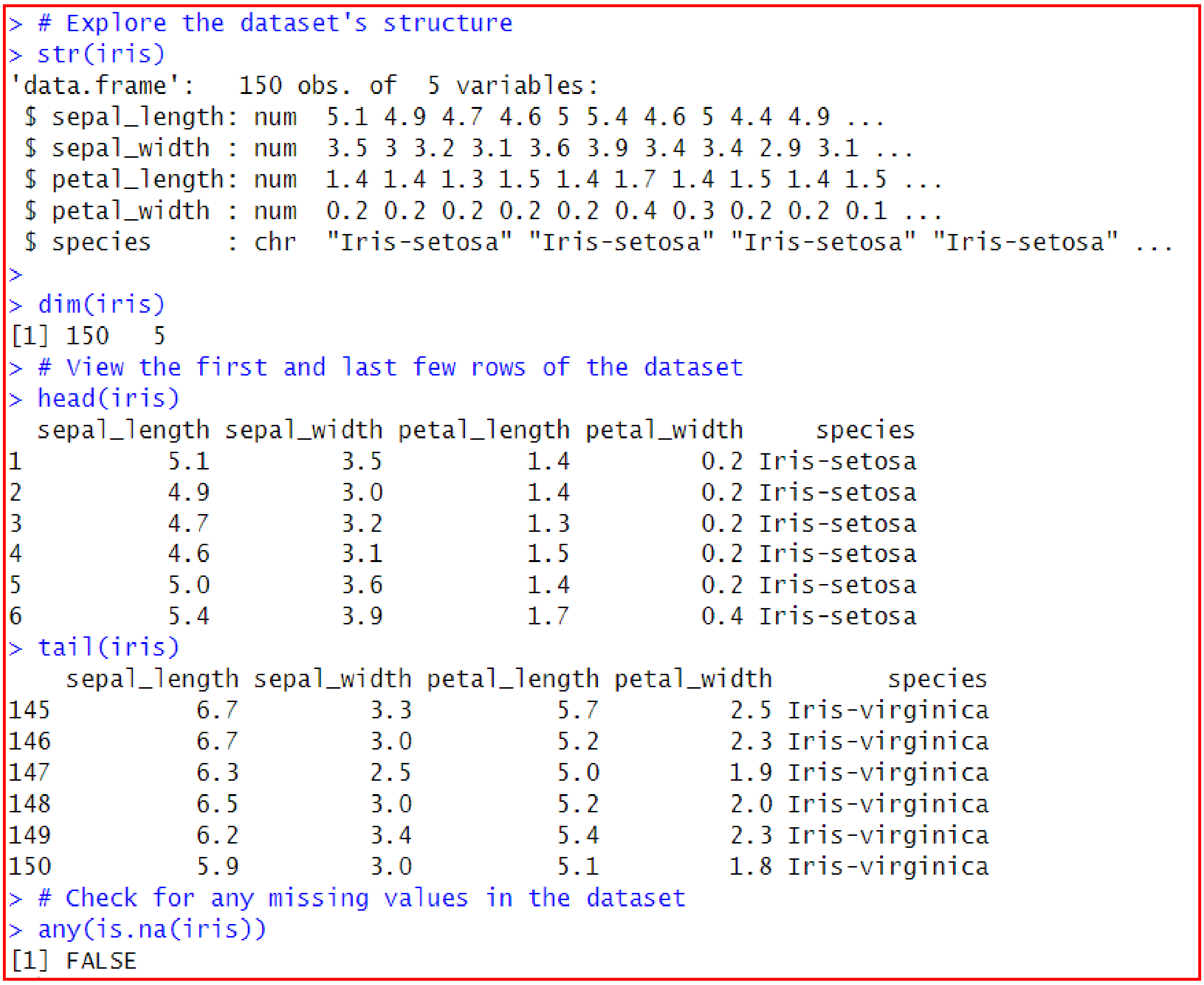
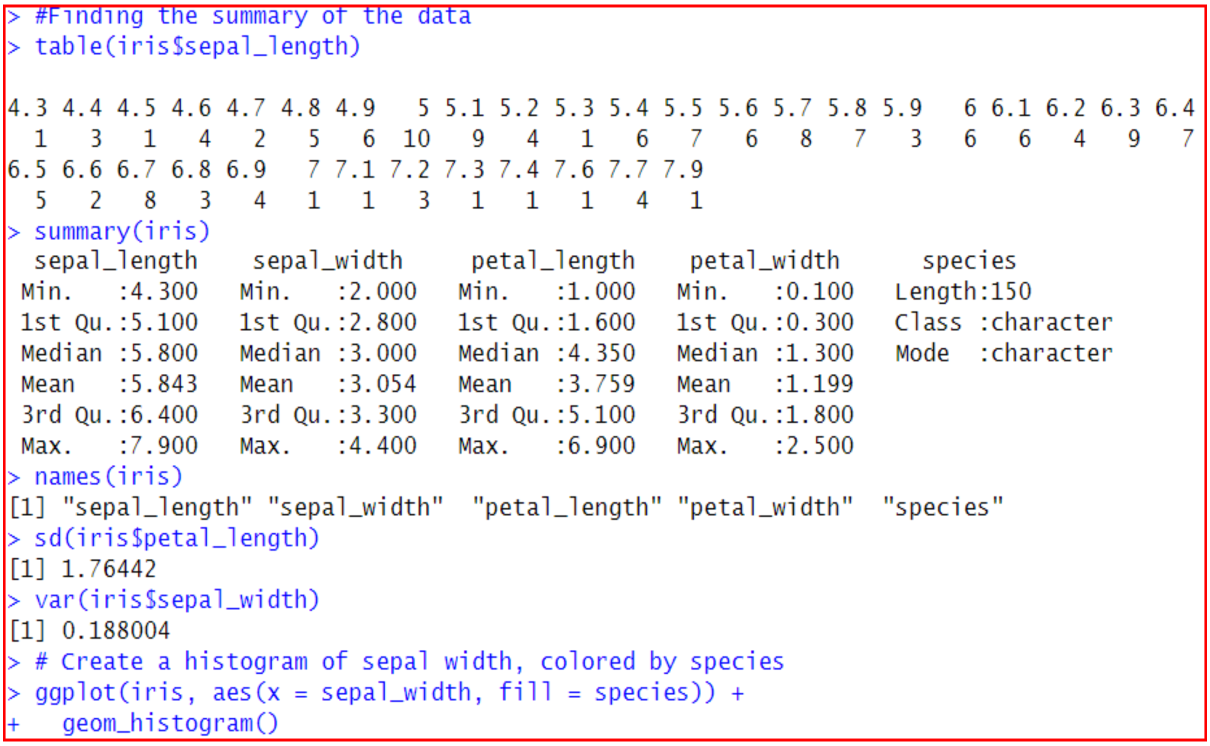
TASK 2: IRIS FLOWER CLASSIFICATION



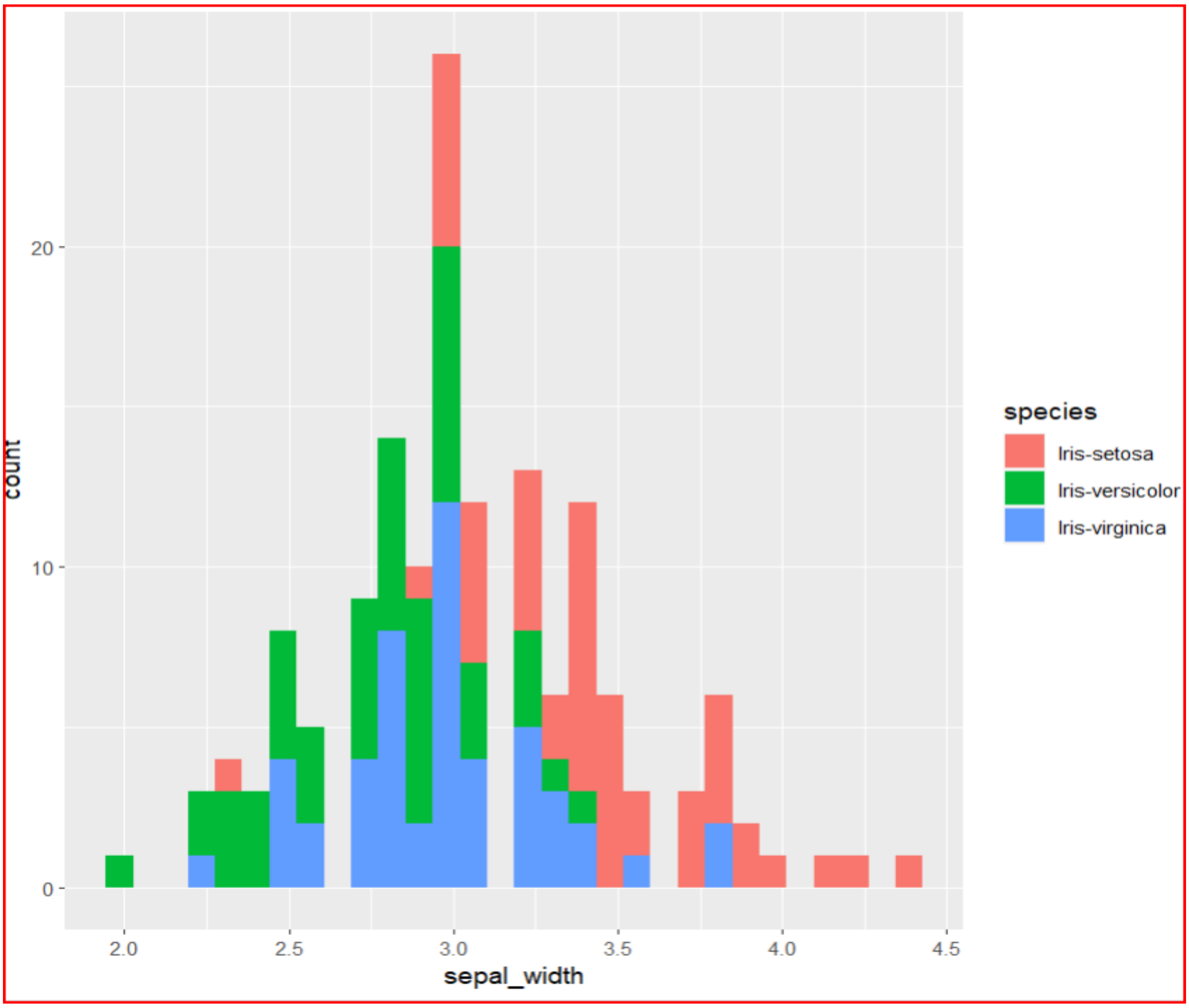
```
# Load necessary libraries
library(tidyverse)
library(dplyr)
library(ggplot2)
library(caTools)
library(randomForest)

# Read the Iris dataset from a CSV file
iris <- read.csv("internship tasks/task 2/Iris Dataset/iris.csv")
```

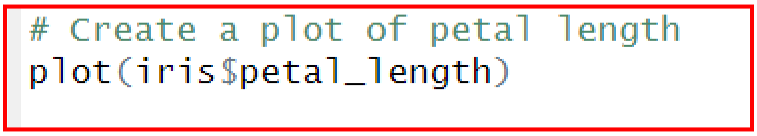


```
> # Explore the dataset's structure
> str(iris)
'data.frame':	150 obs. of  5 variables:
 $ sepal_length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ sepal_width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ petal_length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ petal_width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ species     : chr  "Iris-setosa" "Iris-setosa" "Iris-setosa" "Iris-setosa" ...
>
> dim(iris)
[1] 150   5
> # View the first and last few rows of the dataset
> head(iris)
  sepal_length sepal_width petal_length petal_width     species
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
6          5.4         3.9          1.7         0.4 Iris-setosa
> tail(iris)
    sepal_length sepal_width petal_length petal_width        species
145          6.7         3.3          5.7         2.5 Iris-virginica
146          6.7         3.0          5.2         2.3 Iris-virginica
147          6.3         2.5          5.0         1.9 Iris-virginica
148          6.5         3.0          5.2         2.0 Iris-virginica
149          6.2         3.4          5.4         2.3 Iris-virginica
150          5.9         3.0          5.1         1.8 Iris-virginica
> # Check for any missing values in the dataset
> any(is.na(iris))
[1] FALSE
```

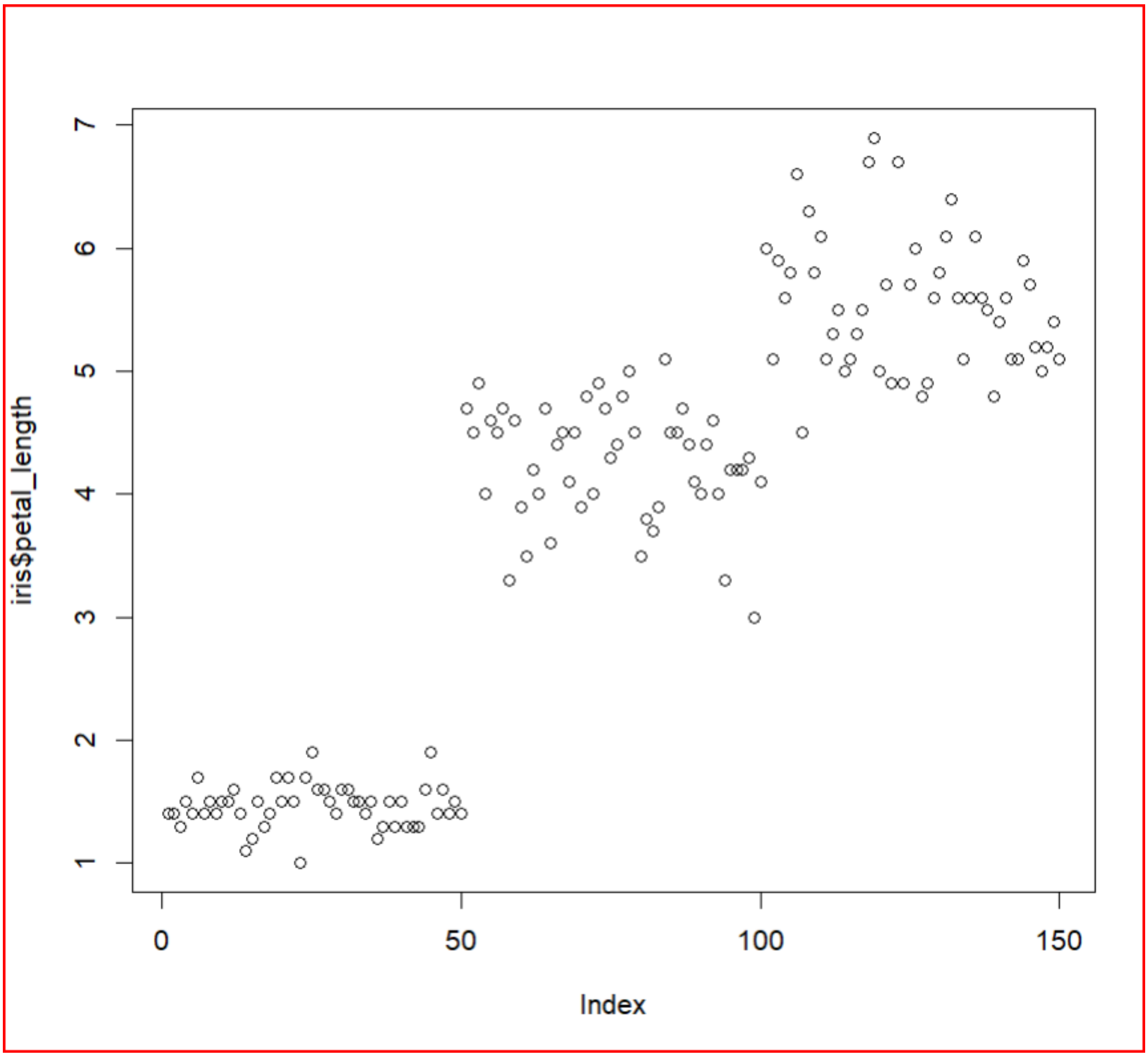


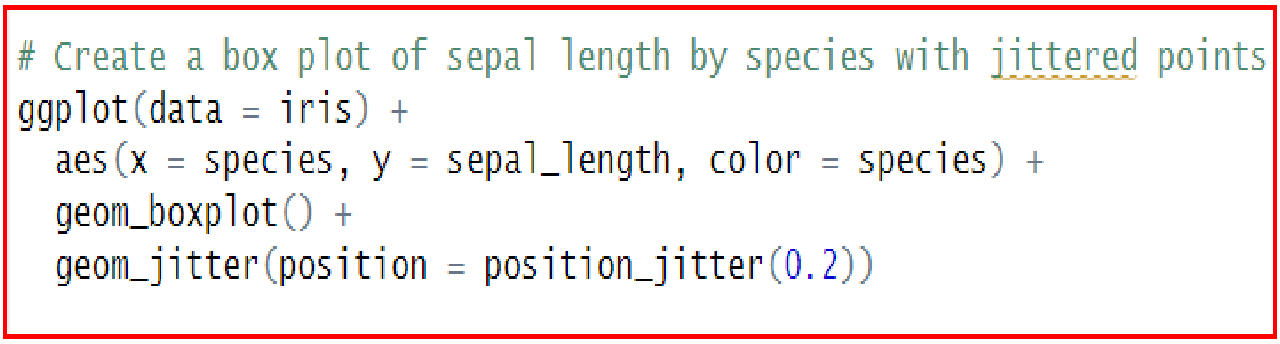
```
> #Finding the summary of the data
> table(iris$sepal_length)

4.3 4.4 4.5 4.6 4.7 4.8 4.9   5 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9   6 6.1 6.2 6.3 6.4
  1   3   1   4   2   5   6  10   9   4   1   6   7   6   8   7   3   6   6   4   9   7
6.5 6.6 6.7 6.8 6.9   7 7.1 7.2 7.3 7.4 7.6 7.7 7.9
  5   2   8   3   4   1   1   3   1   1   1   4   1
> summary(iris)
  sepal_length    sepal_width     petal_length    petal_width      species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Length:150
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class :character
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode  :character
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> names(iris)
[1] "sepal_length" "sepal_width"  "petal_length" "petal_width"  "species"
> sd(iris$petal_length)
[1] 1.76442
> var(iris$sepal_width)
[1] 0.188004
> # Create a histogram of sepal width, colored by species
> ggplot(iris, aes(x = sepal_width, fill = species)) +
+   geom_histogram()
```



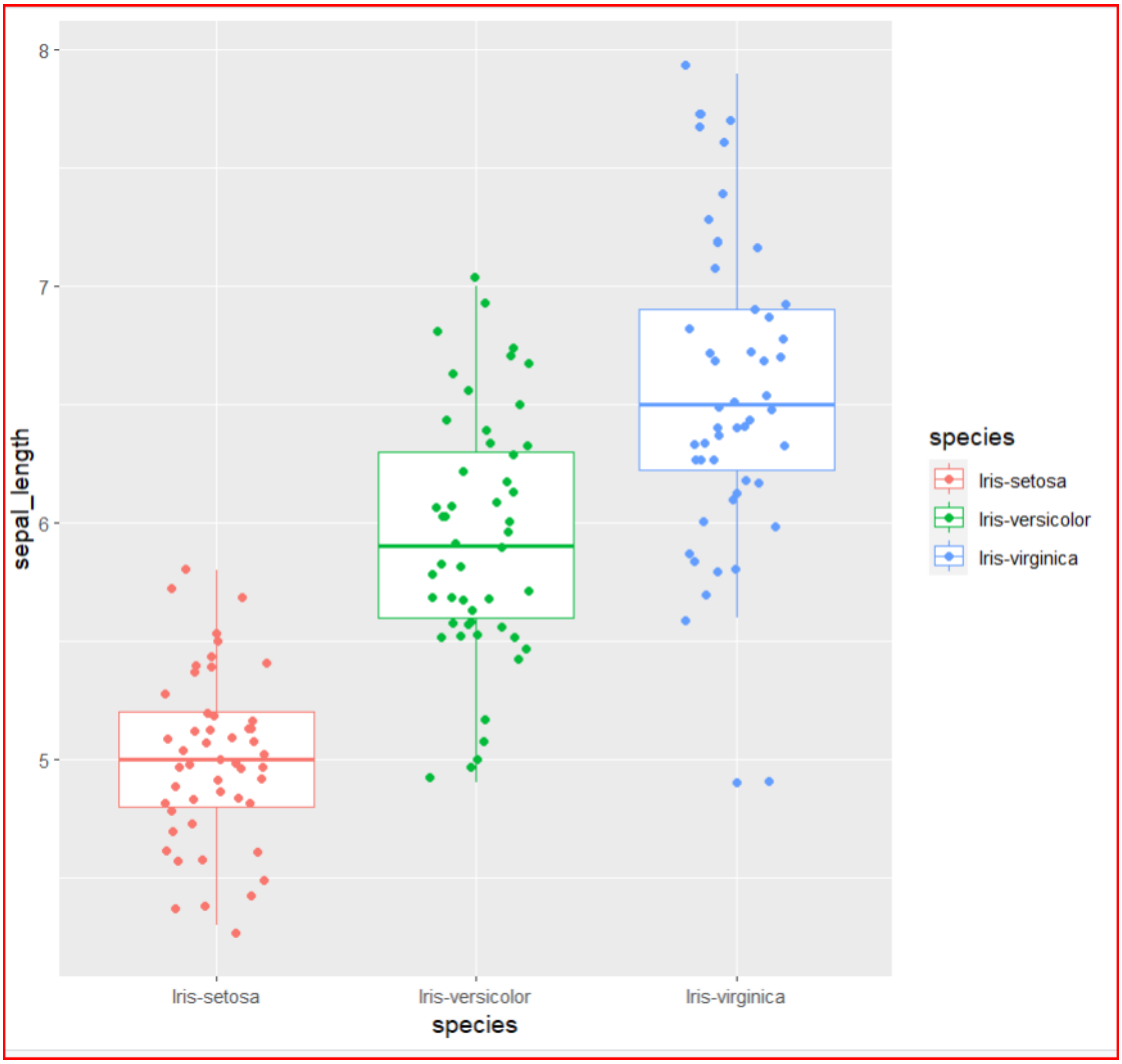


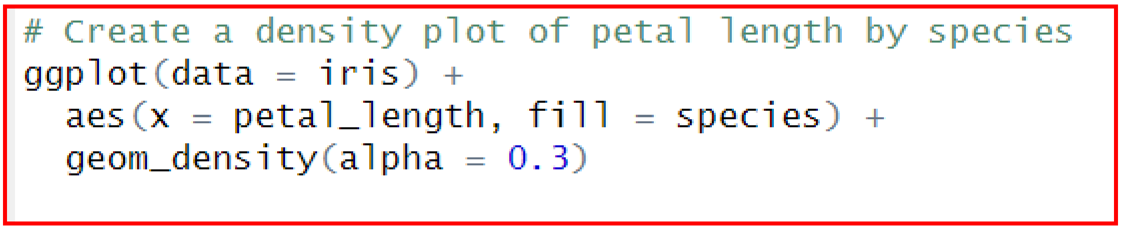
```
# Create a plot of petal length
plot(iris$petal_length)
```



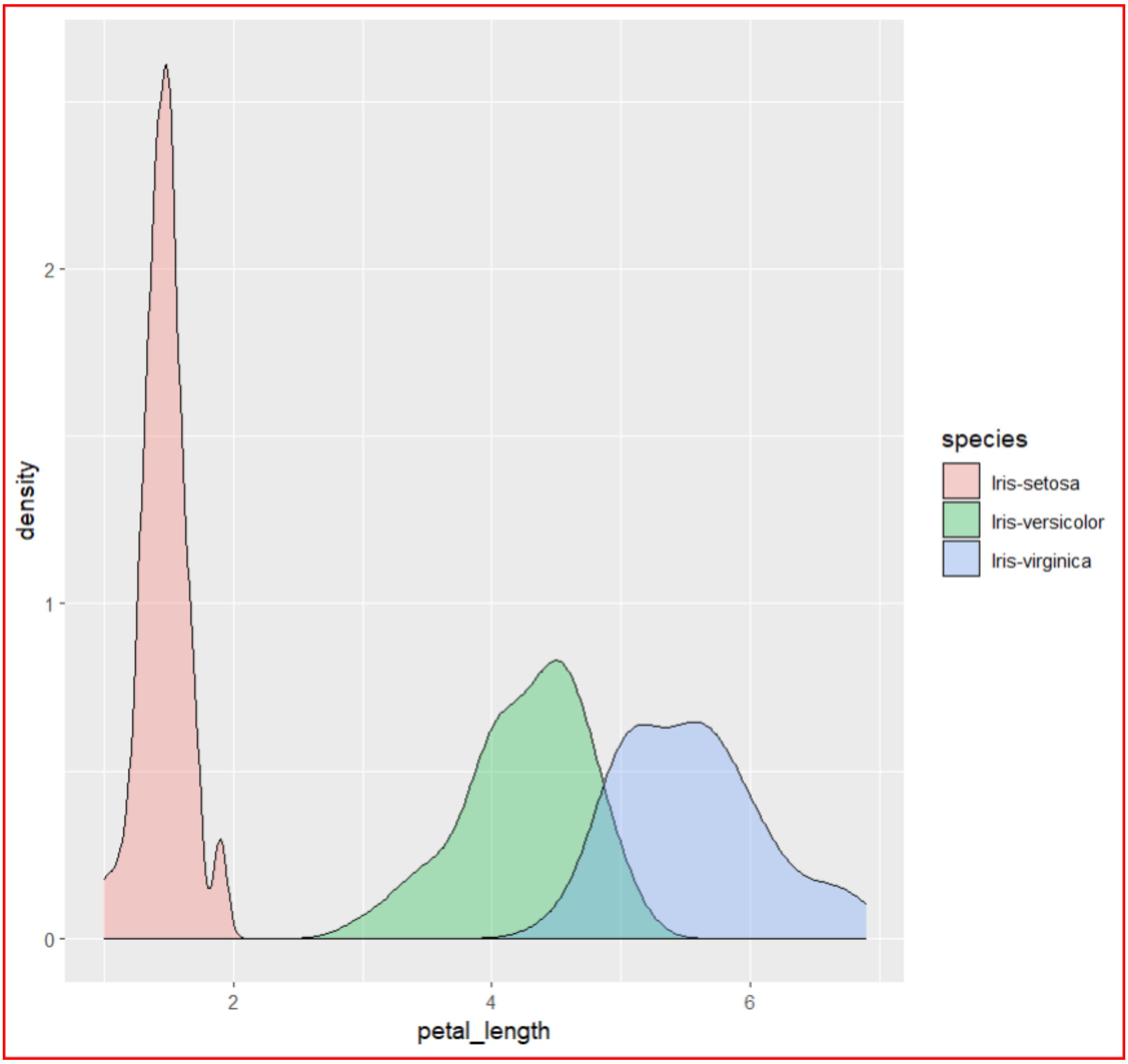


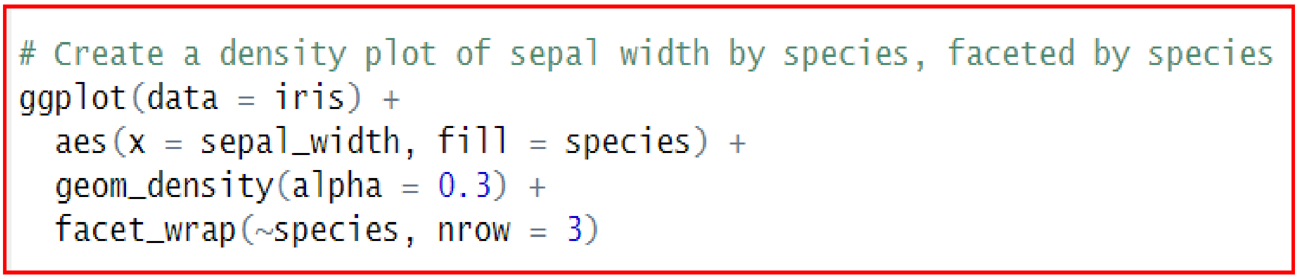
```
# Create a box plot of sepal length by species with jittered points
ggplot(data = iris) +
  aes(x = species, y = sepal_length, color = species) +
  geom_boxplot() +
  geom_jitter(position = position_jitter(0.2))
```



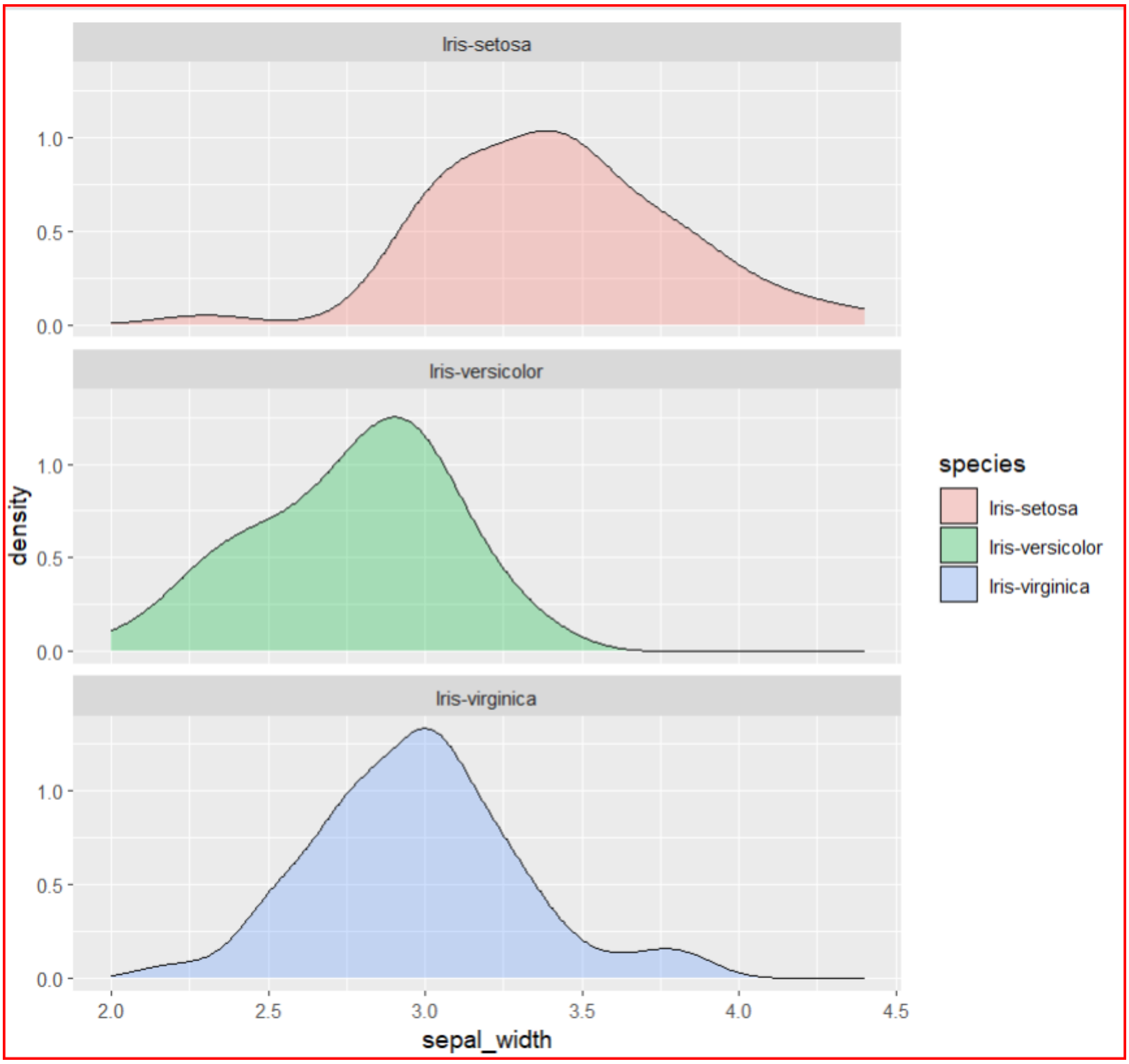


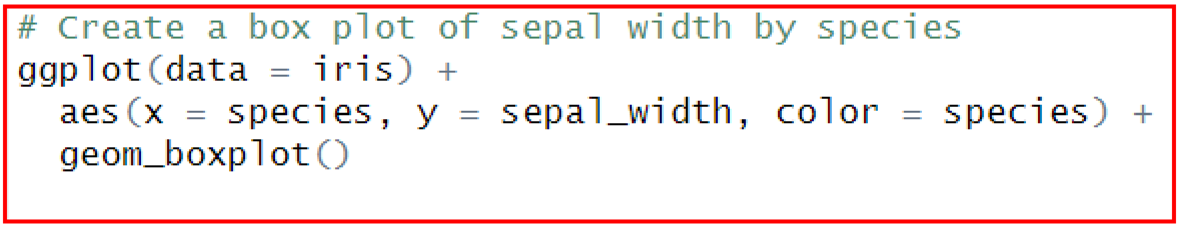
```
# Create a density plot of petal length by species
ggplot(data = iris) +
  aes(x = petal_length, fill = species) +
  geom_density(alpha = 0.3)
```



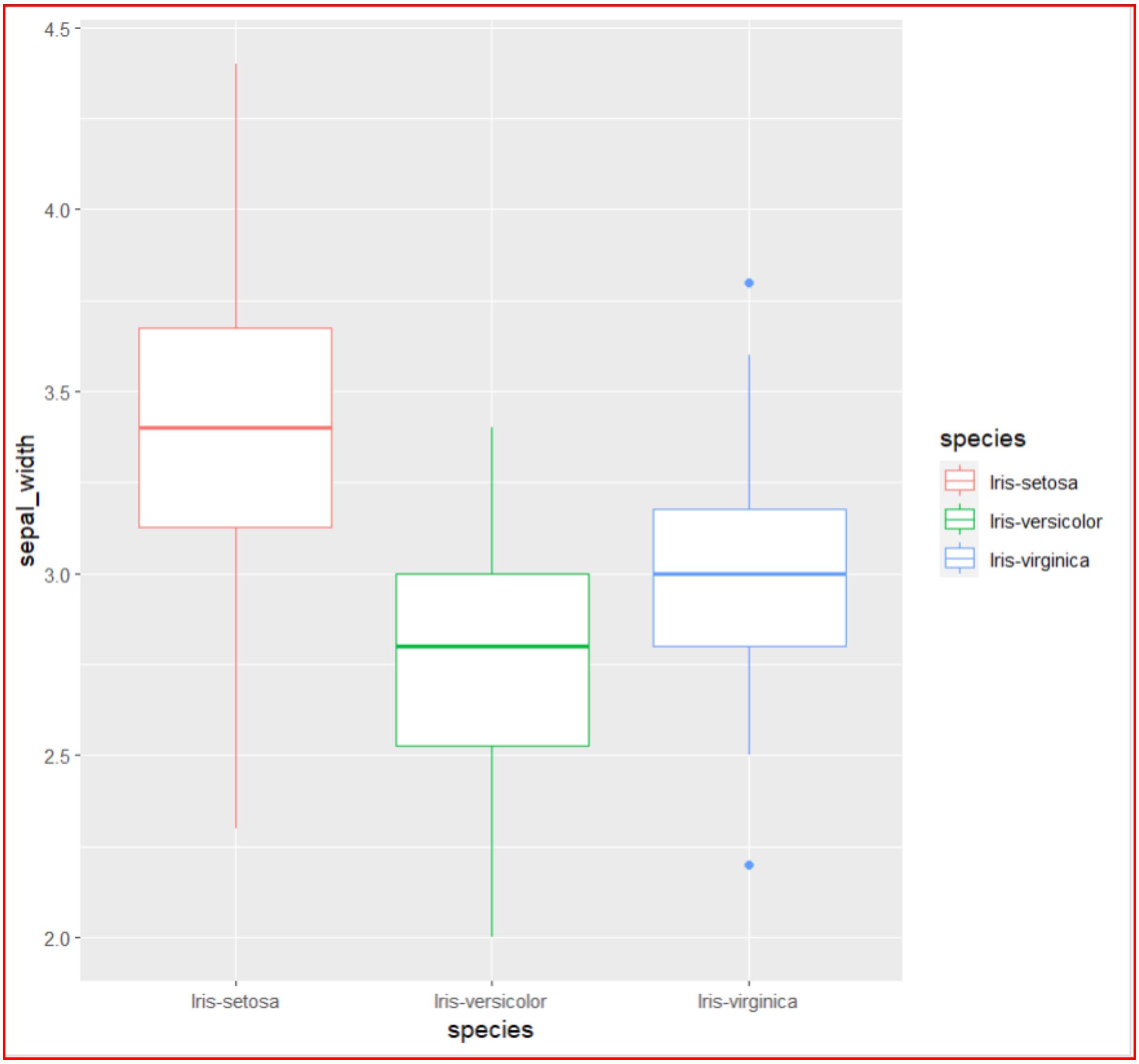


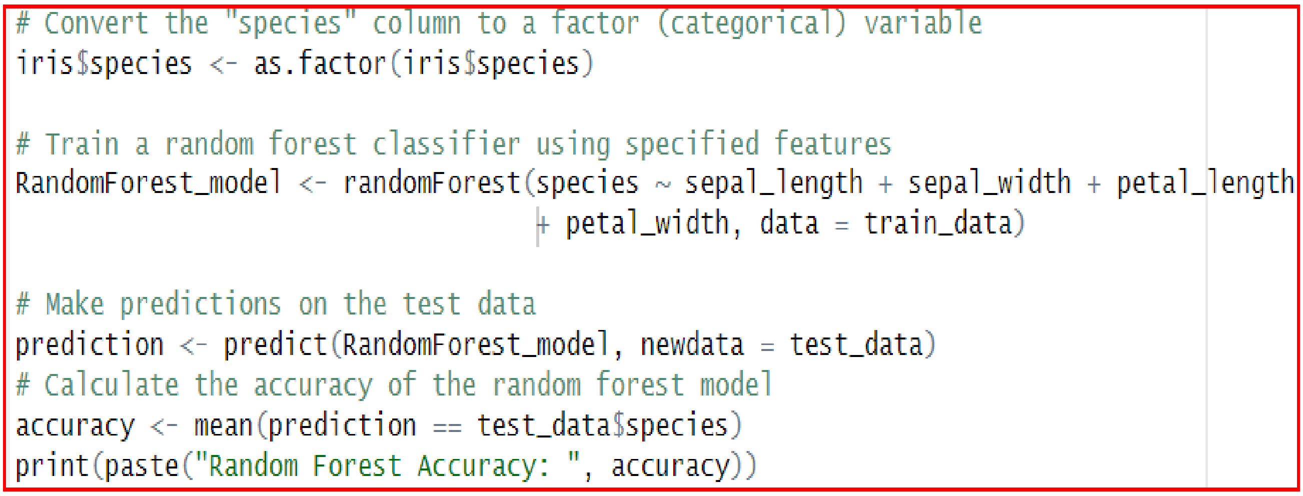
```
# Create a density plot of sepal width by species, faceted by species
ggplot(data = iris) +
  aes(x = sepal_width, fill = species) +
  geom_density(alpha = 0.3) +
  facet_wrap(~species, nrow = 3)
```



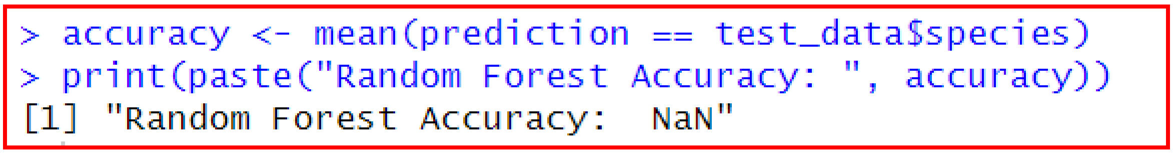


```
# Create a box plot of sepal width by species
ggplot(data = iris) +
  aes(x = species, y = sepal_width, color = species) +
  geom_boxplot()
```

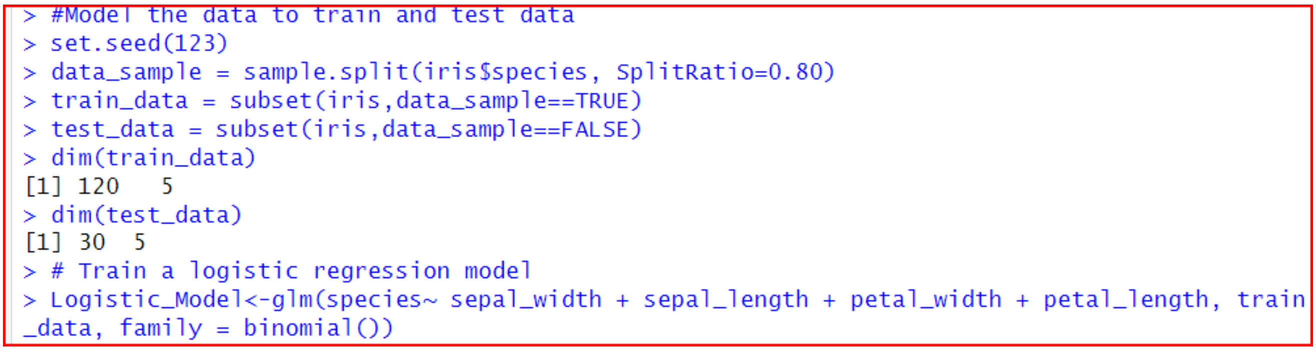




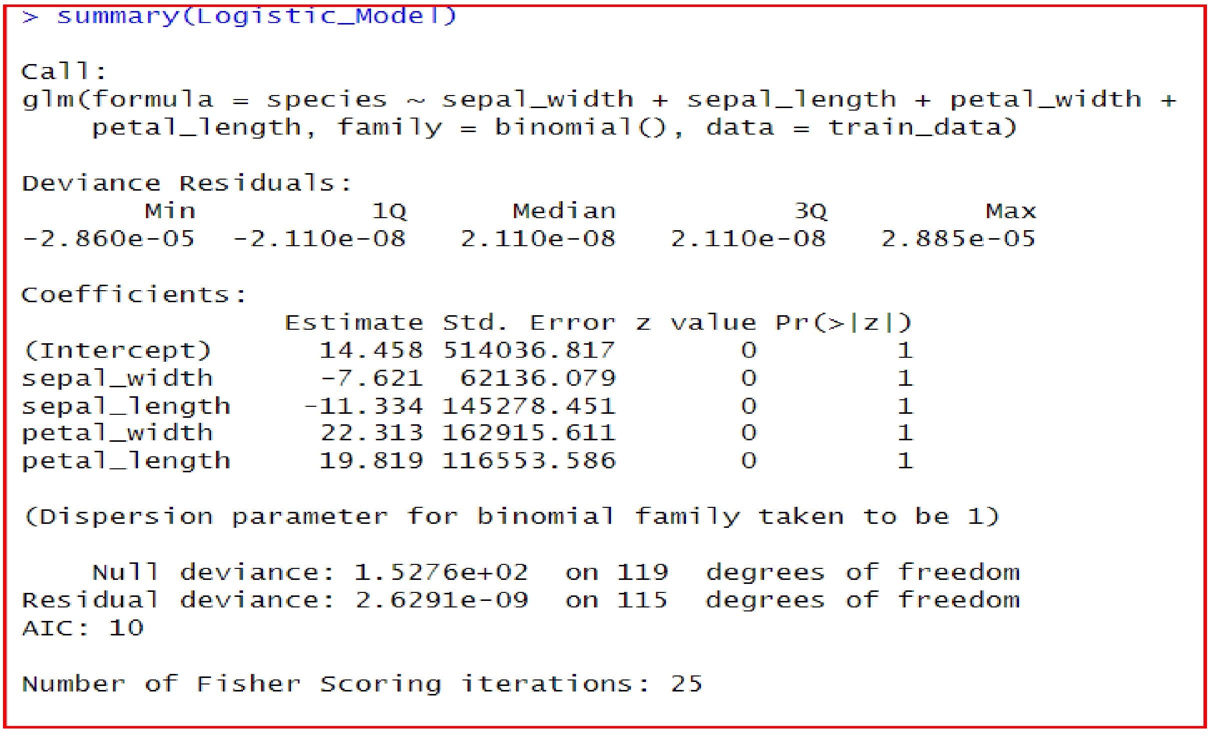
```
# Convert the "species" column to a factor (categorical) variable
iris$species <- as.factor(iris$species)

# Train a random forest classifier using specified features
RandomForest_model <- randomForest(species ~ sepal_length + sepal_width + petal_length
                                   + petal_width, data = train_data)

# Make predictions on the test data
prediction <- predict(RandomForest_model, newdata = test_data)
# Calculate the accuracy of the random forest model
accuracy <- mean(prediction == test_data$species)
print(paste("Random Forest Accuracy: ", accuracy))
```



```
> accuracy <- mean(prediction == test_data$species)
> print(paste("Random Forest Accuracy: ", accuracy))
[1] "Random Forest Accuracy:  NaN"
```



```
> #Model the data to train and test data
> set.seed(123)
> data_sample = sample.split(iris$species, SplitRatio=0.80)
> train_data = subset(iris,data_sample==TRUE)
> test_data = subset(iris,data_sample==FALSE)
> dim(train_data)
[1] 120   5
> dim(test_data)
[1] 30  5
> # Train a logistic regression model
> Logistic_Model<-glm(species~ sepal_width + sepal_length + petal_width + petal_length, train
_data, family = binomial())
```



```
> summary(Logistic_Model)

Call:
glm(formula = species ~ sepal_width + sepal_length + petal_width + 
    petal_length, family = binomial(), data = train_data)

Deviance Residuals: 
       Min          1Q      Median          3Q         Max  
-2.860e-05  -2.110e-08   2.110e-08   2.110e-08   2.885e-05  

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)      14.458 514036.817       0        1
sepal_width      -7.621  62136.079       0        1
sepal_length    -11.334 145278.451       0        1
petal_width      22.313 162915.611       0        1
petal_length     19.819 116553.586       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1.5276e+02  on 119  degrees of freedom
Residual deviance: 2.6291e-09  on 115  degrees of freedom
AIC: 10

Number of Fisher Scoring iterations: 25
```