Set IV

1. (i) Write a function in R programming to print generate Fibonacci sequence using

Recursion in R .

(ii) Find sum of natural numbers up-to 10, without formula using loop

statement.

(iii) create a vector 1:10 and Find a square of each number and store that in a separate list.

**Input:**

i.)fibonacci <- function(n) {

if (n <= 0) {

stop("Input must be a positive integer.")

} else if (n == 1) {

return(0)

} else if (n == 2) {

return(1)

} else {

return(fibonacci(n - 1) + fibonacci(n - 2))

}

}

# Example usage

n <- 10

fib\_sequence <- sapply(1:n, fibonacci)

print(fib\_sequence)

ii.)sum\_natural\_numbers <- 0

for (i in 1:10) {

sum\_natural\_numbers <- sum\_natural\_numbers + i

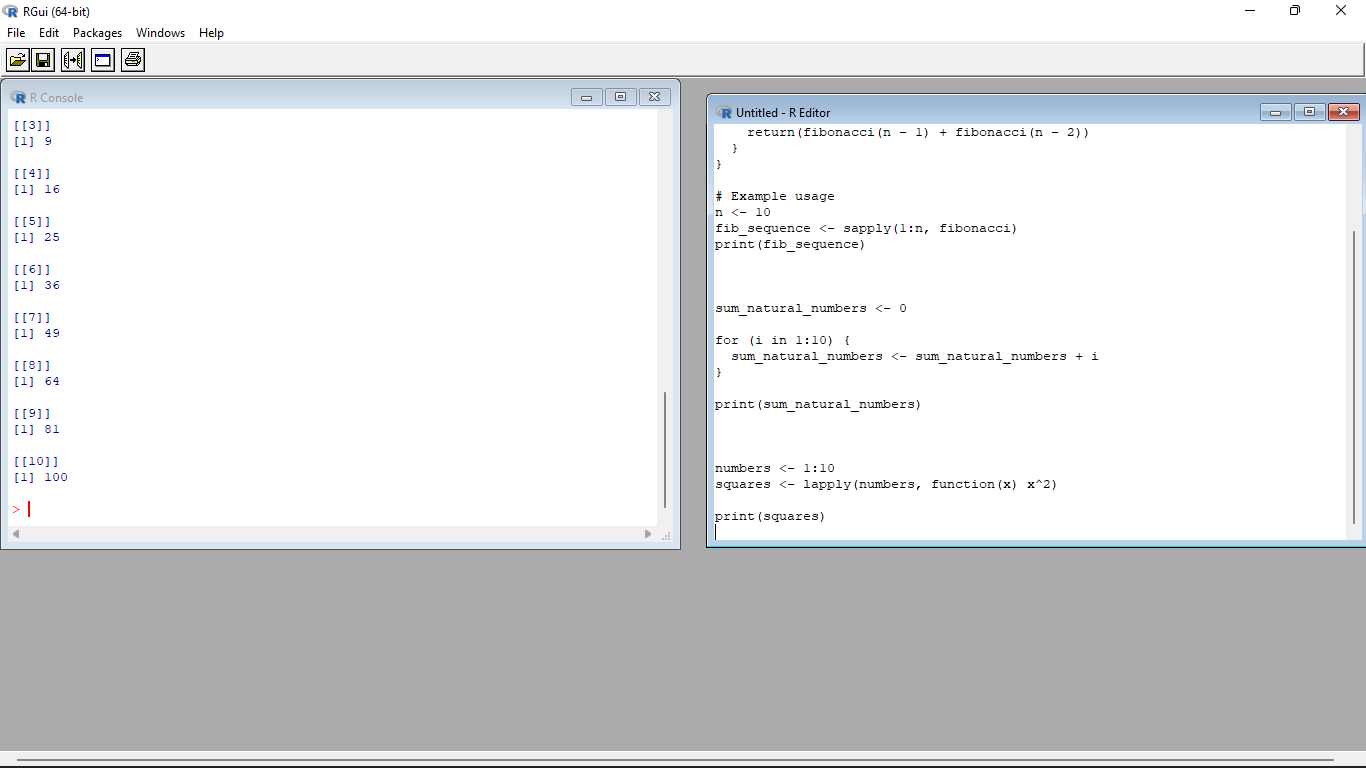
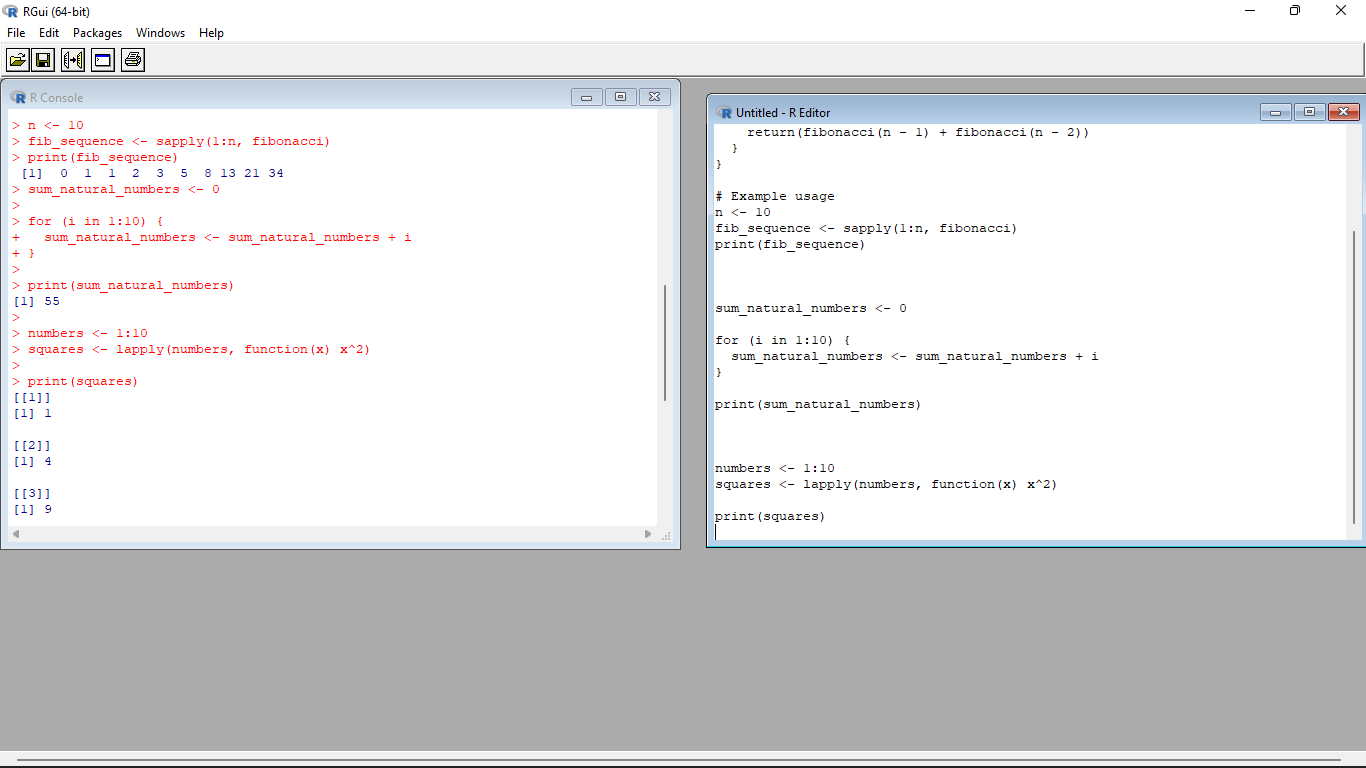
}

print(sum\_natural\_numbers)

iii.)numbers <- 1:10

squares <- lapply(numbers, function(x) x^2)

print(squares)



2. **mtcars**(motor trend car road test) comprises fuel consumption, performance and

10 aspects of automobile design for 32 automobiles. It comes pre-installed with **dplyr** package

in R.

(i)Find the dimension of the data set

(ii)Give the statistical summary of the features.

(iii)Print the categorical features in Dataset

(iv)Find the average weight(wt) grouped by Engine shape(vs)

(v)Find the largest and smallest value of the variable weight with respect to Engine shape

**Input:**

# Load the dplyr package

library(dplyr)

# Check the dimension of the mtcars dataset

dim(mtcars)

# Summarize the features of the mtcars dataset

summary(mtcars)

# Identify the categorical features in the mtcars dataset

categorical\_features <- sapply(mtcars, is.factor)

# Print the categorical features

names(mtcars)[categorical\_features]

# Calculate the average weight (wt) grouped by Engine shape (vs)

mtcars %>%

group\_by(vs) %>%

summarize(avg\_weight = mean(wt))

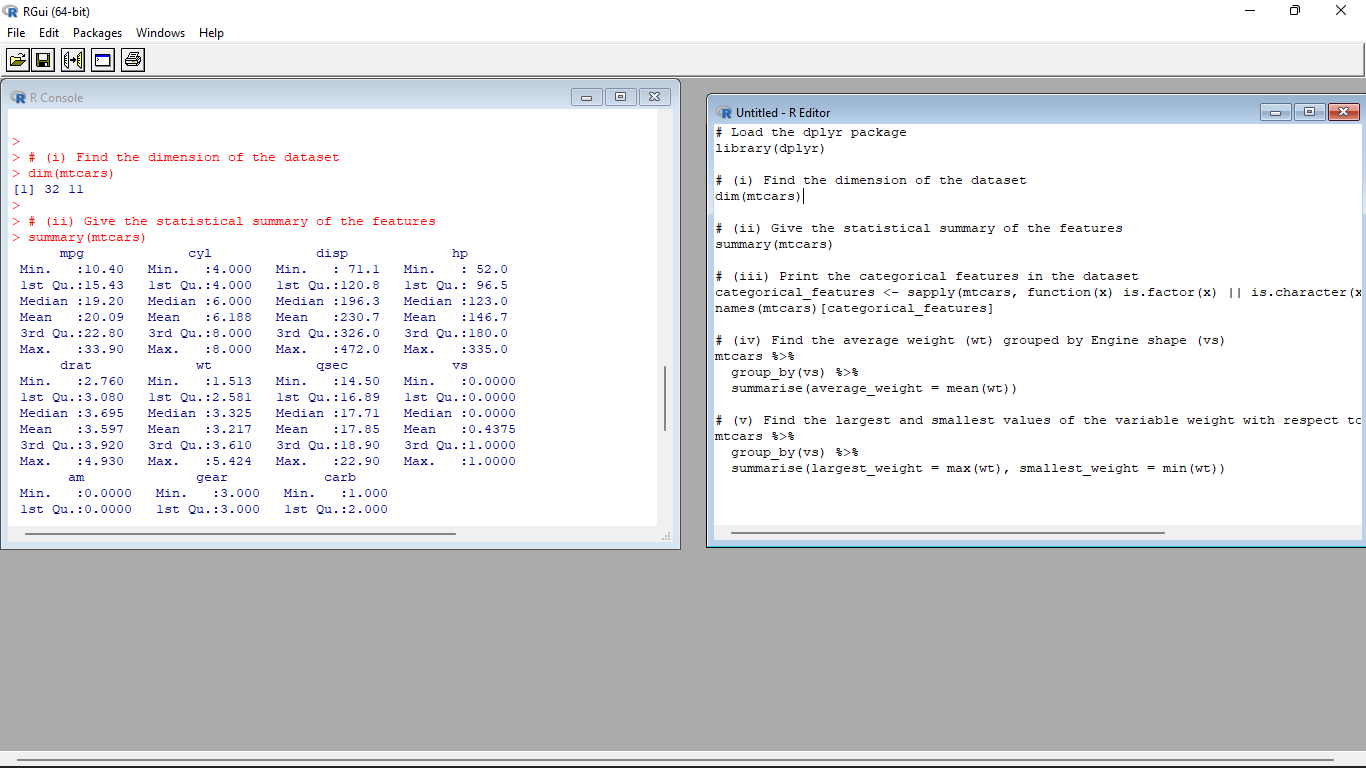
# Find the largest and smallest values of the variable weight with respect to Engine shape

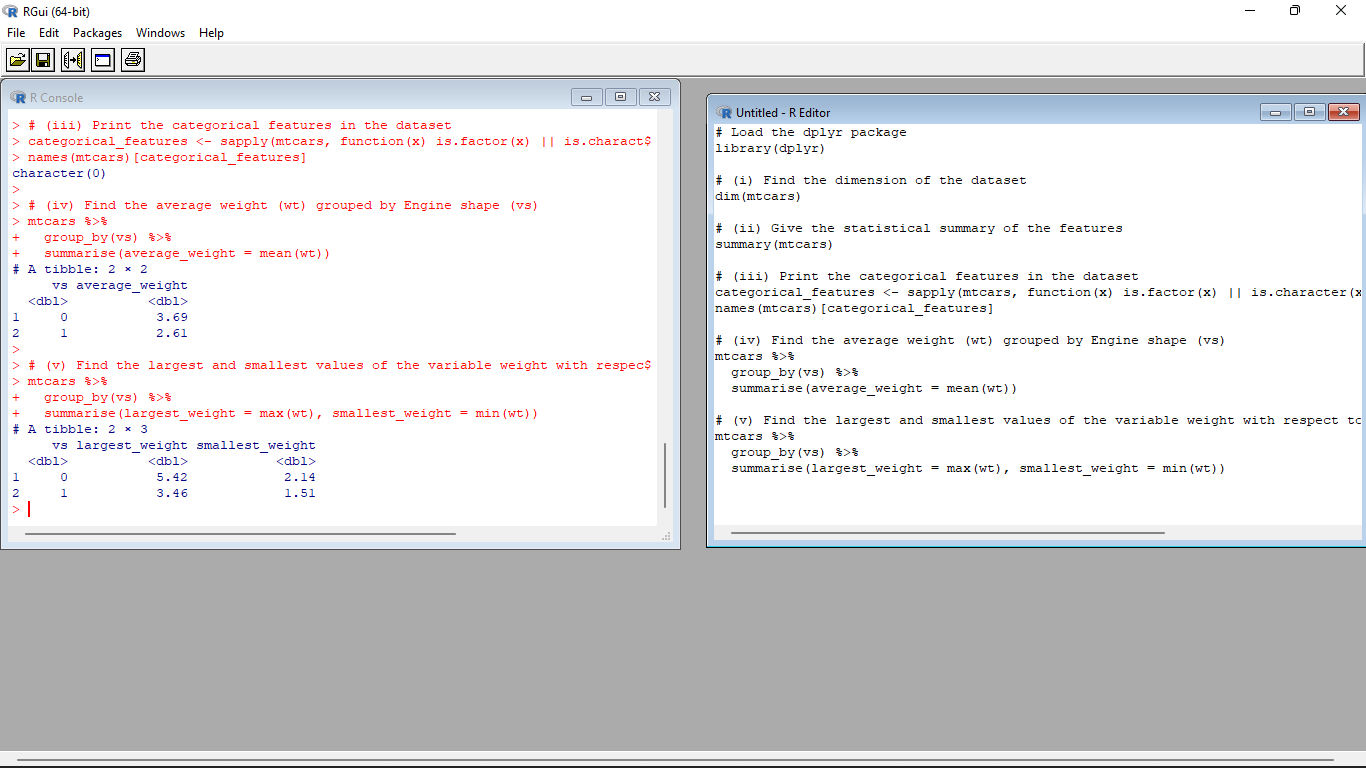
mtcars %>%

group\_by(vs) %>%

summarize(largest\_weight = max(wt),

smallest\_weight = min(wt))





3.Use ggplot package to plot below EDA questions label the plot accordingly

(i)Create weight(wt) vs displacement(disp) scatter plot factor by Engine Shape(vs)

(ii) Create horsepower (hp) vs mileage (mgp) scatter plot factor by Engine Shape(vs)

(iv)In above(ii) plot , Separate columns according to cylinders(cyl) size

(v) Create histogram plot for horsepower (hp) with bin-width size of 5

**Input**:

# Load the ggplot2 package

library(ggplot2)

# Create weight (wt) vs displacement (disp) scatter plot, factorized by Engine Shape (vs)

ggplot(mtcars, aes(x = wt, y = disp, color = factor(vs))) +

geom\_point() +

labs(x = "Weight (wt)", y = "Displacement (disp)", color = "Engine Shape (vs)") +

theme\_minimal()

# Create horsepower (hp) vs mileage (mpg) scatter plot, factorized by Engine Shape (vs)

ggplot(mtcars, aes(x = hp, y = mpg, color = factor(vs))) +

geom\_point() +

labs(x = "Horsepower (hp)", y = "Mileage (mpg)", color = "Engine Shape (vs)") +

theme\_minimal()

# Separate columns in the plot according to cylinder (cyl) size

ggplot(mtcars, aes(x = hp, y = mpg, color = factor(vs))) +

geom\_point() +

labs(x = "Horsepower (hp)", y = "Mileage (mpg)", color = "Engine Shape (vs)") +

facet\_wrap(~ cyl) +

theme\_minimal()

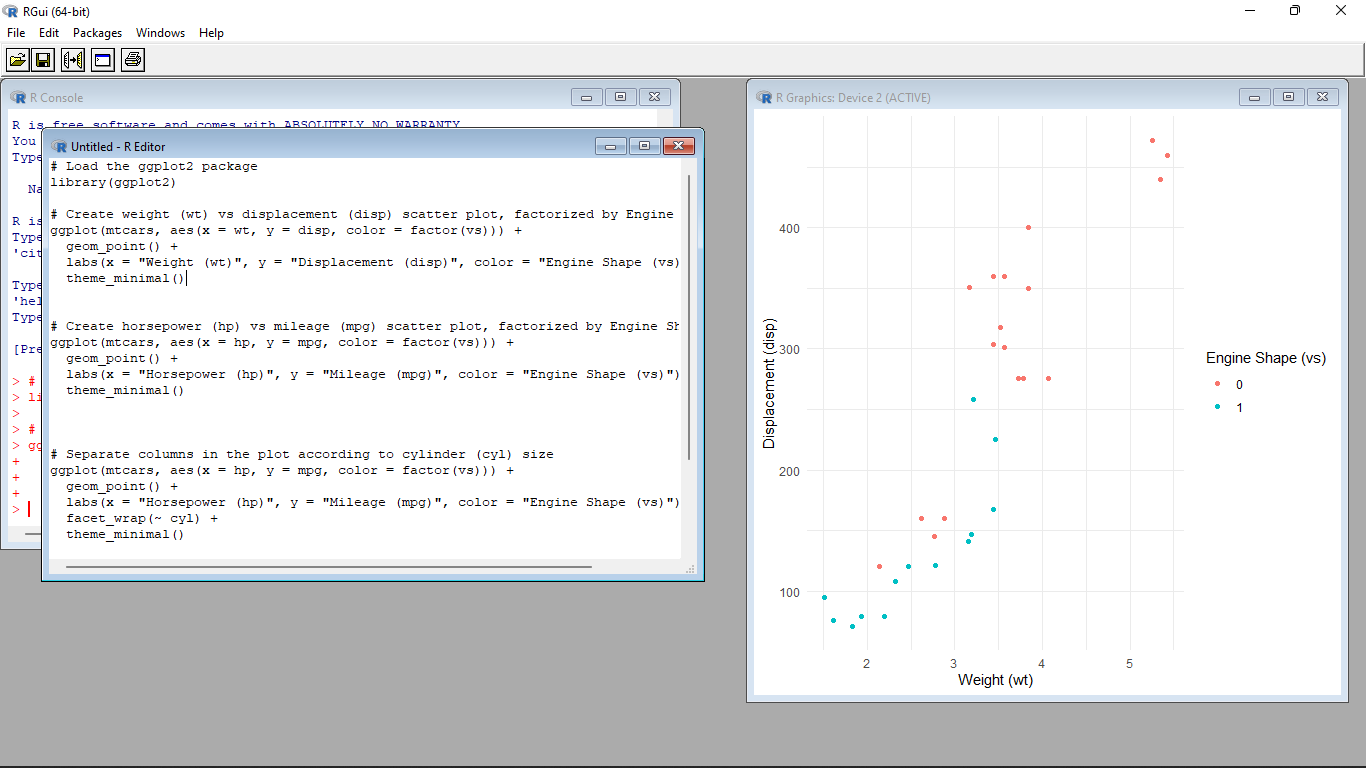
# Create a histogram plot for horsepower (hp) with a bin-width size of 5

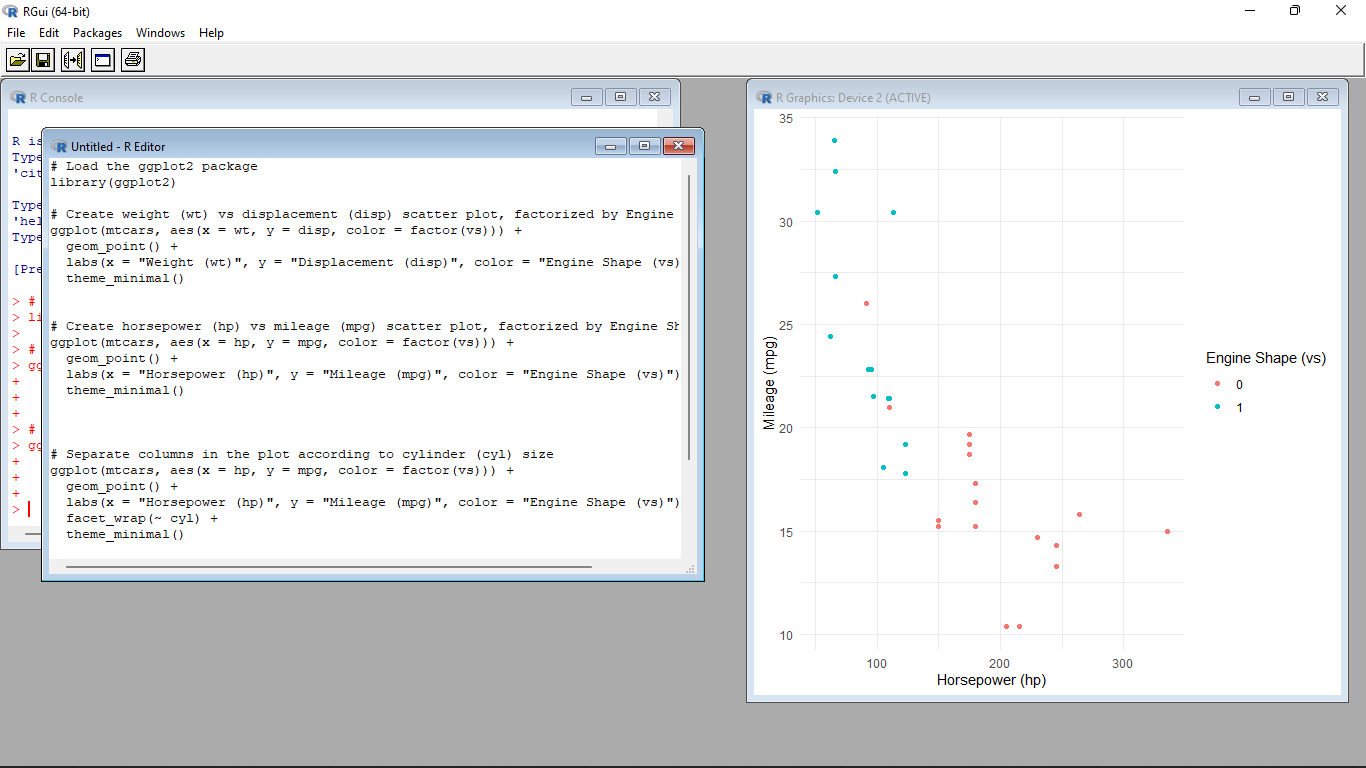
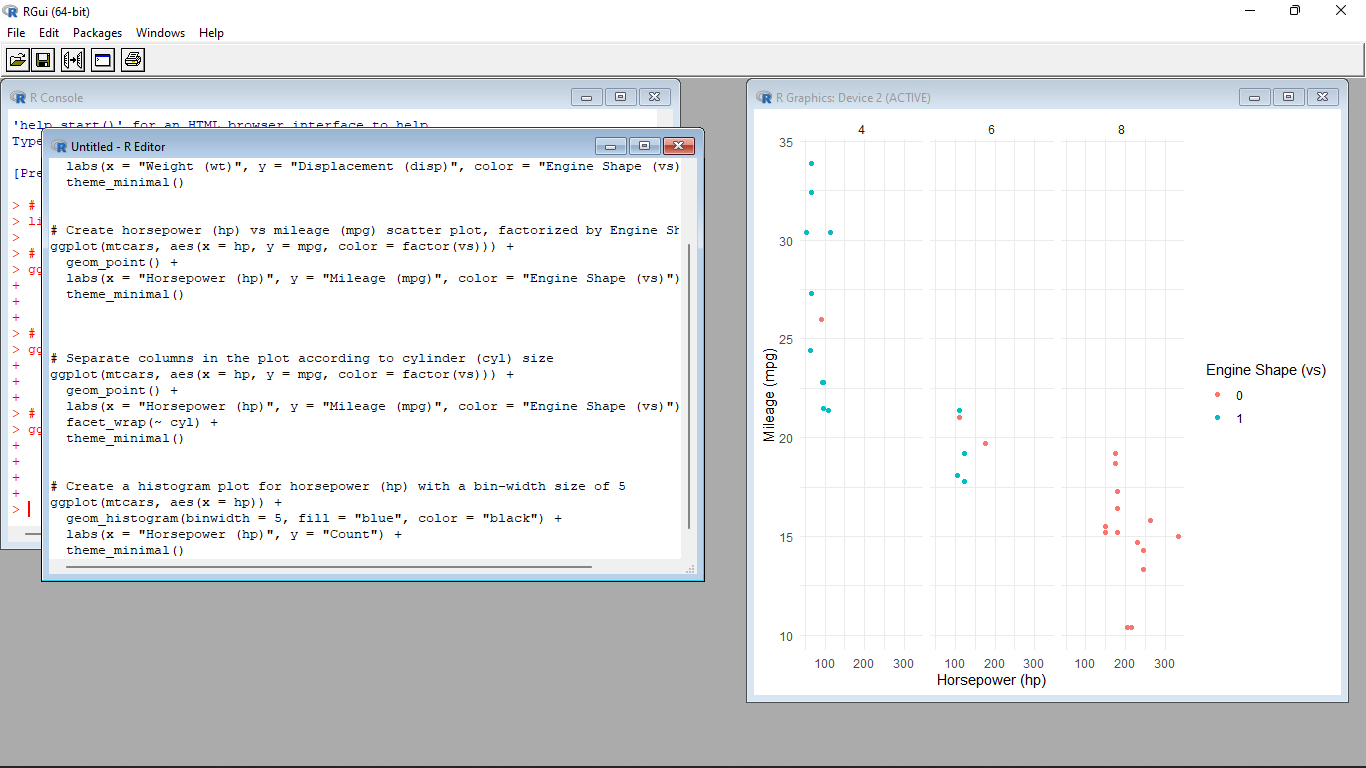
ggplot(mtcars, aes(x = hp)) +

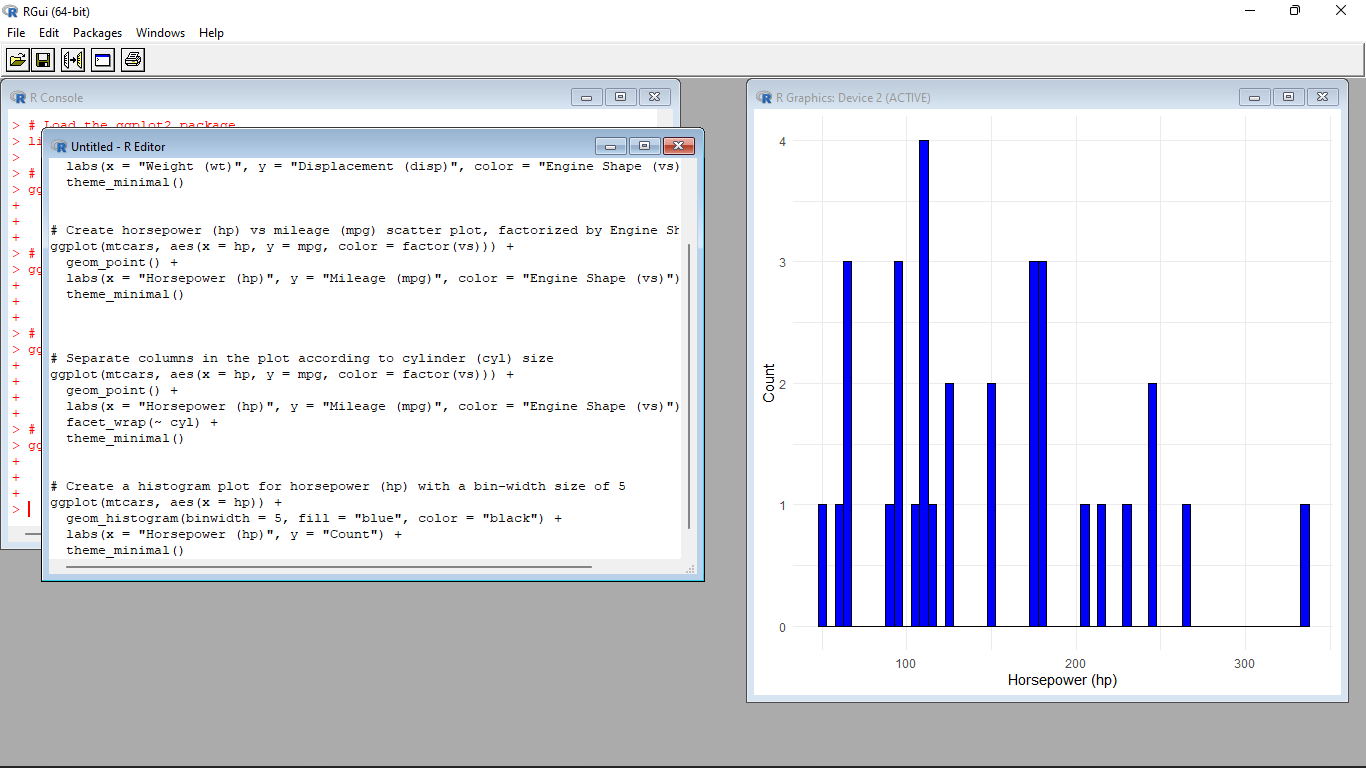
geom\_histogram(binwidth = 5, fill = "blue", color = "black") +

labs(x = "Horsepower (hp)", y = "Count") +

theme\_minimal()







4. Performing Logistic regression on dataset to predict the cars Engine shape(vs) .

(i)Do the EDA analysis and find the features which is impact the Engine shape and use this for model.

(ii) Split the data set randomly with 80:20 ration to create train and test dataset and create logistic

model

(iii)Create the Confusion matrix among prediction and test data.

**Input**:

# Load the required packages

library(ggplot2)

library(dplyr)

# EDA analysis

# Explore the relationship between Engine Shape (vs) and other variables

eda\_data <- mtcars %>%

select(vs, mpg, hp, wt, qsec) # Include features of interest

# Create scatter plots

scatter\_plots <- lapply(names(eda\_data)[-1], function(var) {

ggplot(eda\_data, aes\_string(x = var, y = "vs")) +

geom\_point() +

labs(x = var, y = "Engine Shape (vs)") +

theme\_minimal()

})

# Print scatter plots

print(scatter\_plots)

# Load the required package

library(caret)

# Split the dataset into train and test datasets

set.seed(123)

train\_indices <- createDataPartition(mtcars$vs, p = 0.8, list = FALSE)

train\_data <- mtcars[train\_indices, ]

test\_data <- mtcars[-train\_indices, ]

# Load the required package

library(caret)

# Train the logistic regression model

logistic\_model <- train(vs ~ mpg + hp + wt + qsec, data = train\_data, method = "glm", family = "binomial")

# Make predictions on the test dataset

predictions <- predict(logistic\_model, newdata = test\_data)

# Create the confusion matrix

confusion\_matrix <- confusionMatrix(predictions, test\_data$vs)

print(confusion\_matrix)

