Set-III

1. Suppose you track your commute times for two weeks (10 days) and you find the following

times in minutes 17 16 20 24 22 15 21 15 17 22 Enter this into R as vector data type.

(i)create function maxi to find the longest commute time, the function avger to find the average and

the function mini to find the minimum.

(ii)Oops, the 24 was a mistake. It should have been 18. How can you fix this? Do so, and then find

the new average using above functions.

(iii)How many times was your commute 20 minutes or more?

**Input**:

# Vector of commute times

commute\_times <- c(17, 16, 20, 24, 22, 15, 21, 15, 17, 22)

# Function to find the longest commute time

maxi <- function(commute\_times) {

max(commute\_times)

}

# Function to find the average commute time

avger <- function(commute\_times) {

mean(commute\_times)

}

# Function to find the minimum commute time

mini <- function(commute\_times) {

min(commute\_times)

}

# Calling the functions

max\_time <- maxi(commute\_times)

avg\_time <- avger(commute\_times)

min\_time <- mini(commute\_times)

# Printing the results

print(paste("Longest commute time:", max\_time))

print(paste("Average commute time:", avg\_time))

print(paste("Minimum commute time:", min\_time))

# Correcting the mistake

commute\_times[commute\_times == 24] <- 18

# Finding the new average using the functions

new\_avg\_time <- avger(commute\_times)

# Printing the new average

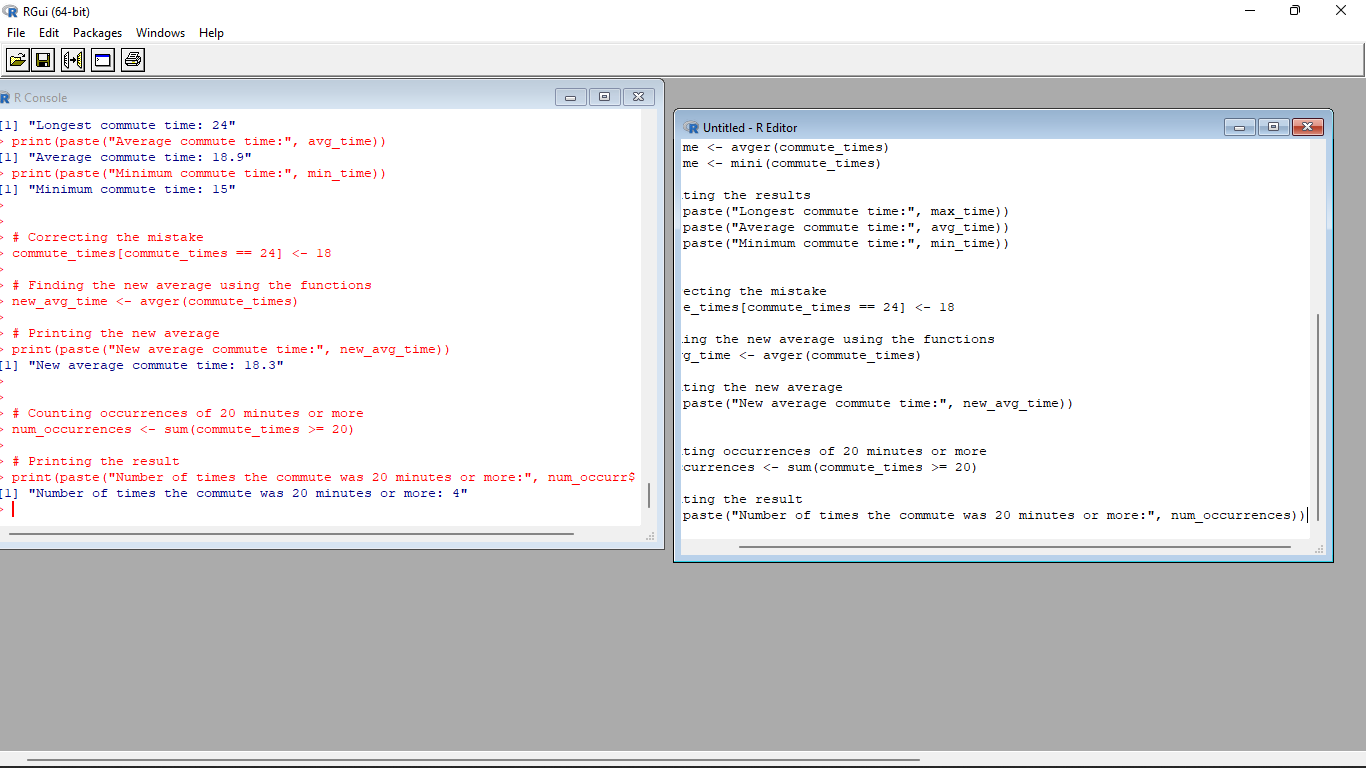
print(paste("New average commute time:", new\_avg\_time))

# Counting occurrences of 20 minutes or more

num\_occurrences <- sum(commute\_times >= 20)

# Printing the result

print(paste("Number of times the commute was 20 minutes or more:", num\_occurrences))



2. There is a popular built-in data set in R called "**mtcars**" (Motor Trend Car Road

Tests), which is retrieved from the 1974 Motor Trend US Magazine.

(i)Find the dimension of the data set

(ii)Give the statistical summary of the features.

(iii)Find the largest and smallest value of the variable hp (horsepower).

(iv)Give the mean of mileage per gallon (mpg) with respect to transmission model (feature named

as ‘am’)

(v)Give the median of horsepower (hp) with respect to cylinder displacement(cyl)

**Input**:

# Load the mtcars dataset

data(mtcars)

# Get the dimension of the dataset

dim(mtcars)

# Get the summary of the dataset

summary(mtcars)

# Find the largest value of "hp"

max\_hp <- max(mtcars$hp)

# Find the smallest value of "hp"

min\_hp <- min(mtcars$hp)

# Print the results

print(paste("Largest value of horsepower (hp):", max\_hp))

print(paste("Smallest value of horsepower (hp):", min\_hp))

# Calculate the mean of mpg by transmission type (am)

mean\_mpg <- tapply(mtcars$mpg, mtcars$am, mean)

# Print the mean mpg for each transmission type

print("Mean mpg by transmission type:")

print(mean\_mpg)

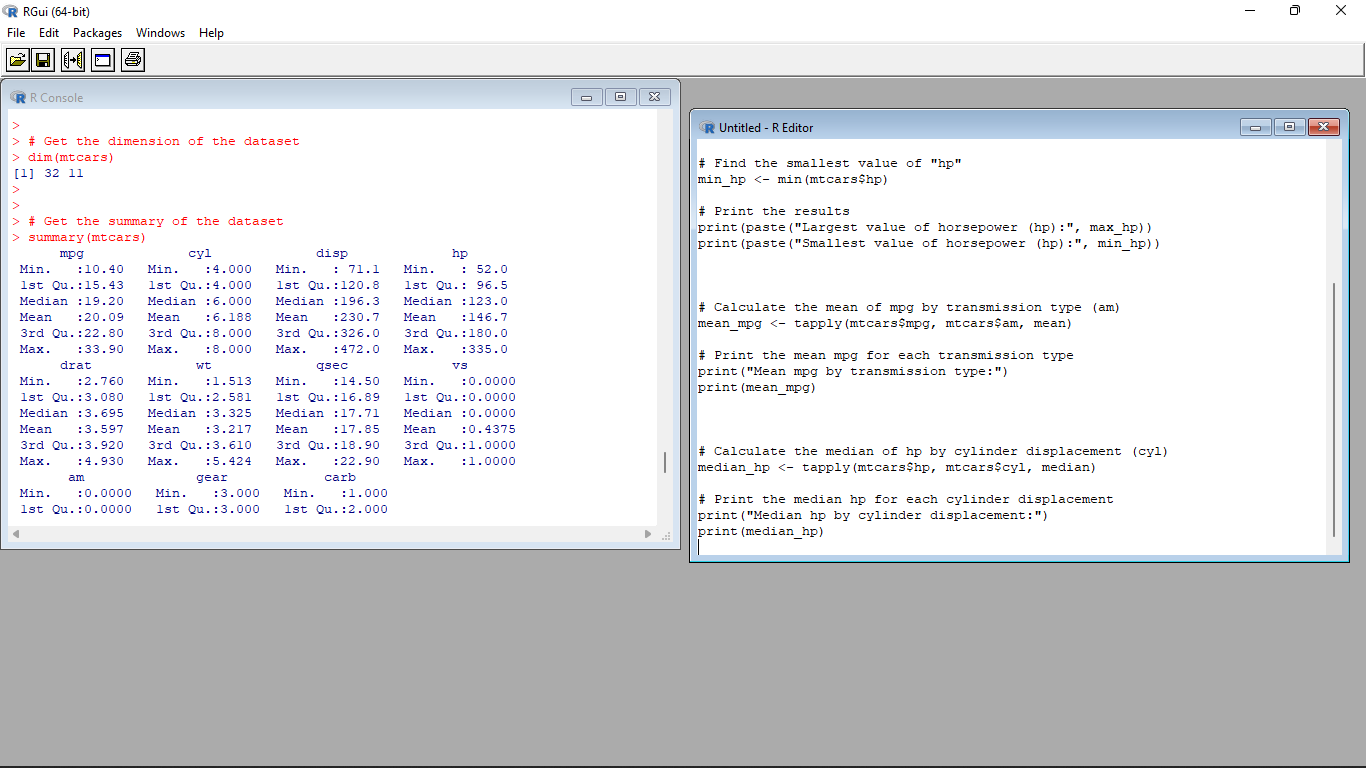
# Calculate the median of hp by cylinder displacement (cyl)

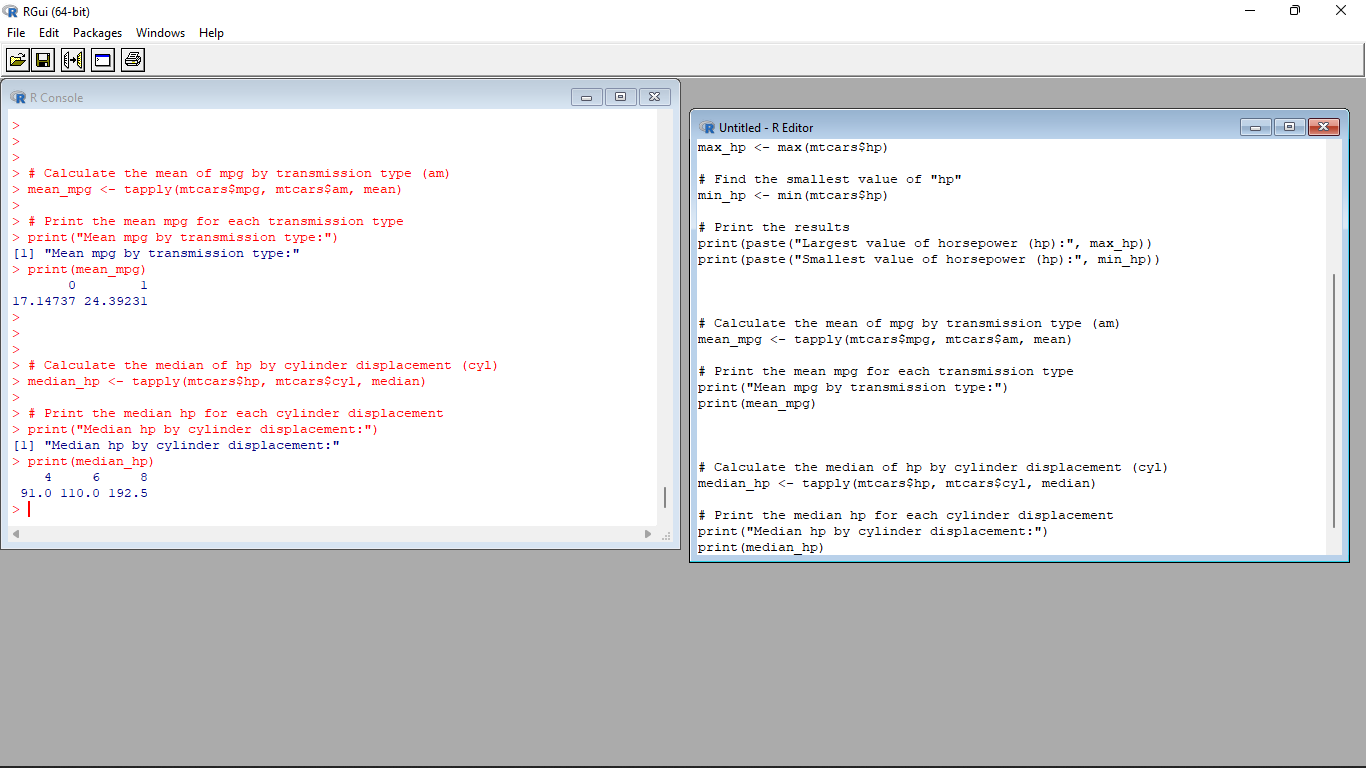
median\_hp <- tapply(mtcars$hp, mtcars$cyl, median)

# Print the median hp for each cylinder displacement

print("Median hp by cylinder displacement:")

print(median\_hp)





3.(i)Create Scatter plot mpg vs hp, grouped by transmission model (feature named as ‘am’)

(ii)Create Box plot for mpg with respect to transmission model (feature named as ‘am’)

(iii)Create histogram plot which shows statistical distribution of hp

(iv)Draw the Bar Chart to show car distribution with respect to number of gears grouped by

cylinder.(Grouped or multiple bar chart)

(v)Draw Pie chart which shows the percentage of distribution by number of gears.

**Input**:

# Load the mtcars dataset

data(mtcars)

# Scatter plot of mpg vs hp, grouped by transmission model

plot(mtcars$hp, mtcars$mpg, col = mtcars$am, pch = 19, xlab = "Horsepower (hp)", ylab = "Miles per Gallon (mpg)")

legend("topright", legend = c("Automatic", "Manual"), col = c(1, 2), pch = 19, title = "Transmission")

# Box plot of mpg with respect to transmission model

boxplot(mpg ~ am, data = mtcars, xlab = "Transmission", ylab = "Miles per Gallon (mpg)", main = "Box Plot of MPG by Transmission")

# Histogram plot of hp

hist(mtcars$hp, breaks = 10, col = "skyblue", xlab = "Horsepower (hp)", ylab = "Frequency", main = "Histogram of HP")

# Bar chart of car distribution by number of gears, grouped by cylinder

barplot(table(mtcars$gear, mtcars$cyl), beside = TRUE, col = c("skyblue", "orange", "green"), xlab = "Number of Gears", ylab = "Frequency", main = "Car Distribution by Number of Gears and Cylinder")

legend("topright", legend = c("4 Cylinder", "6 Cylinder", "8 Cylinder"), fill = c("skyblue", "orange", "green"))

# Pie chart of percentage distribution by number of gears

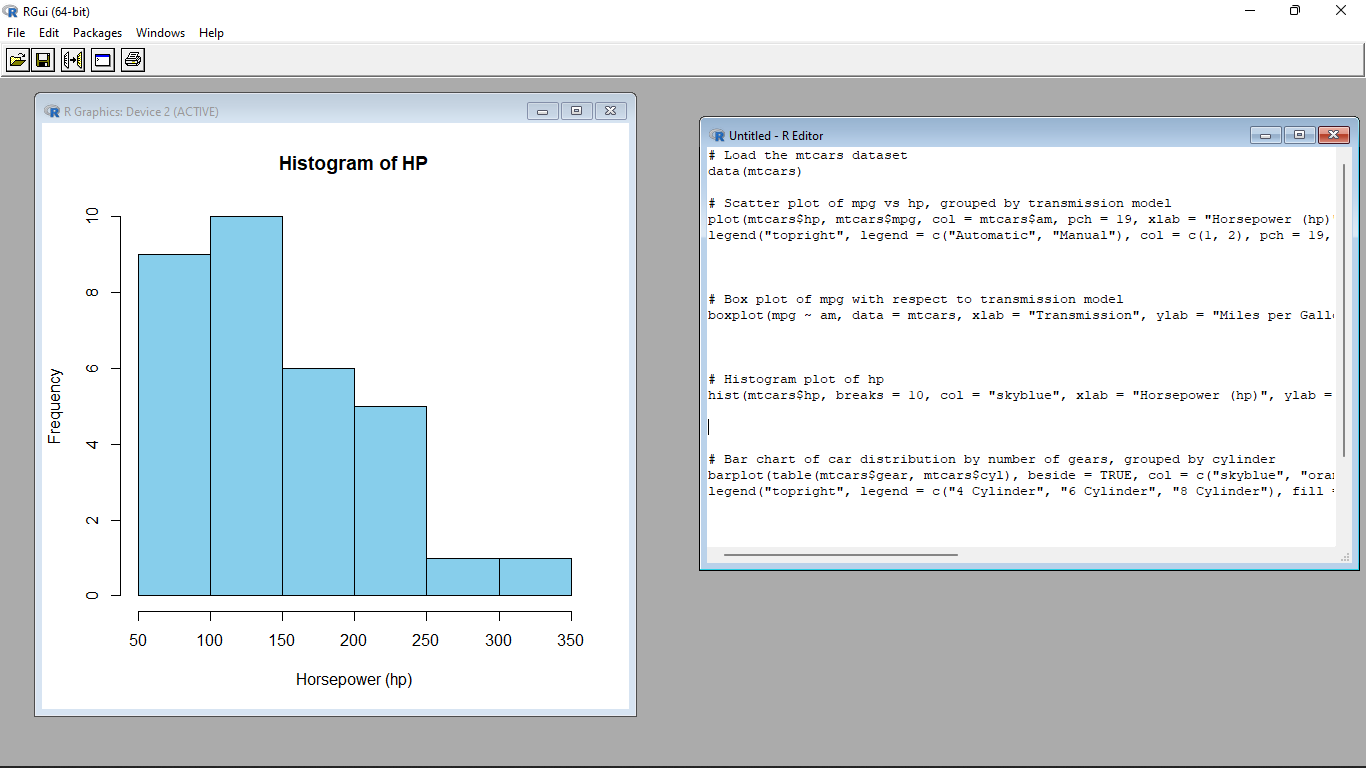
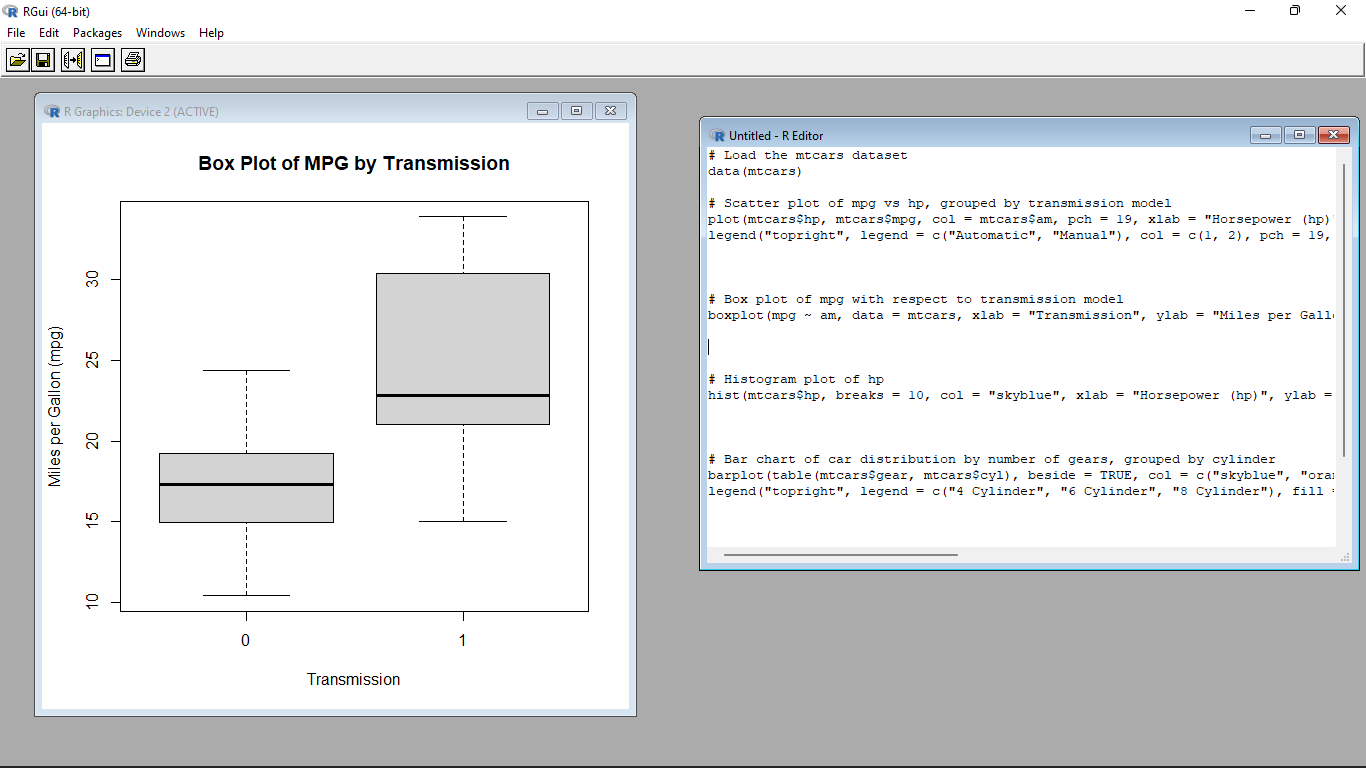
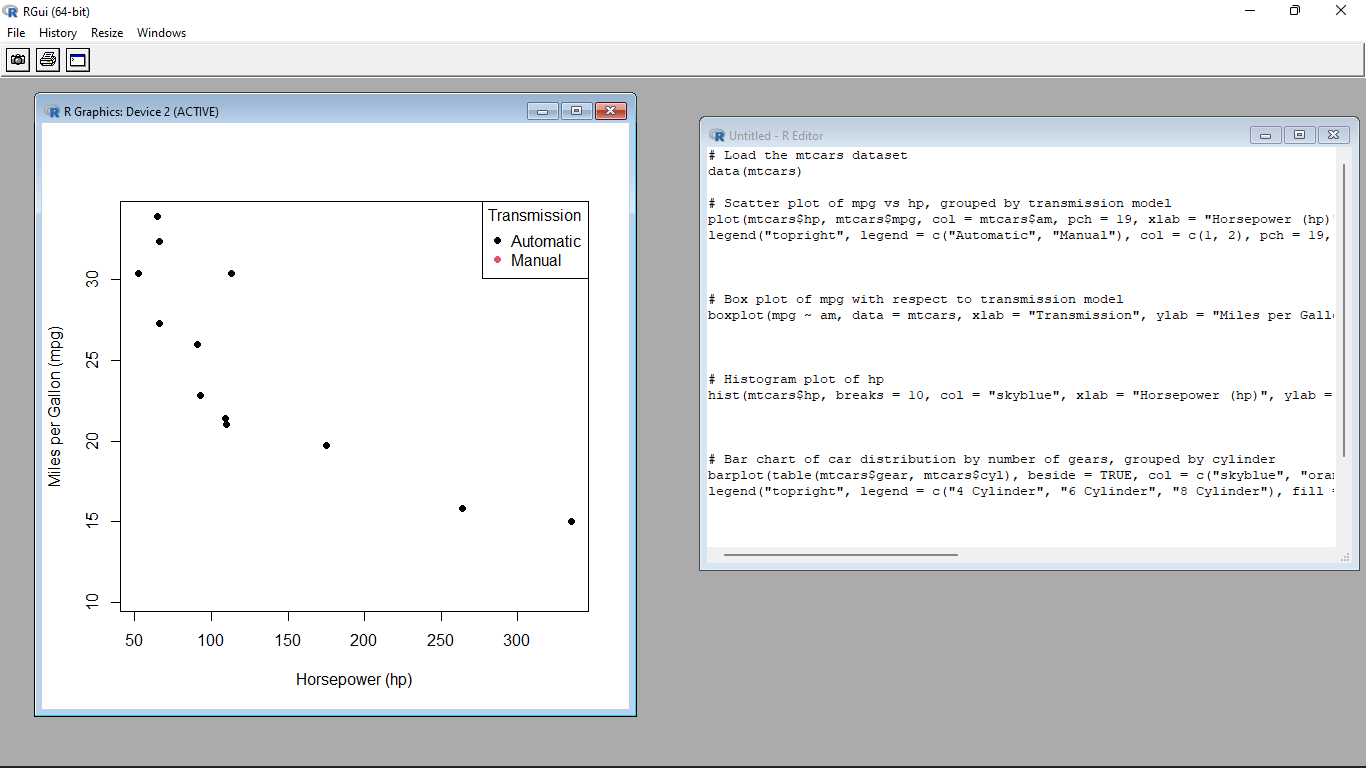
gear\_counts <- table(mtcars$gear)

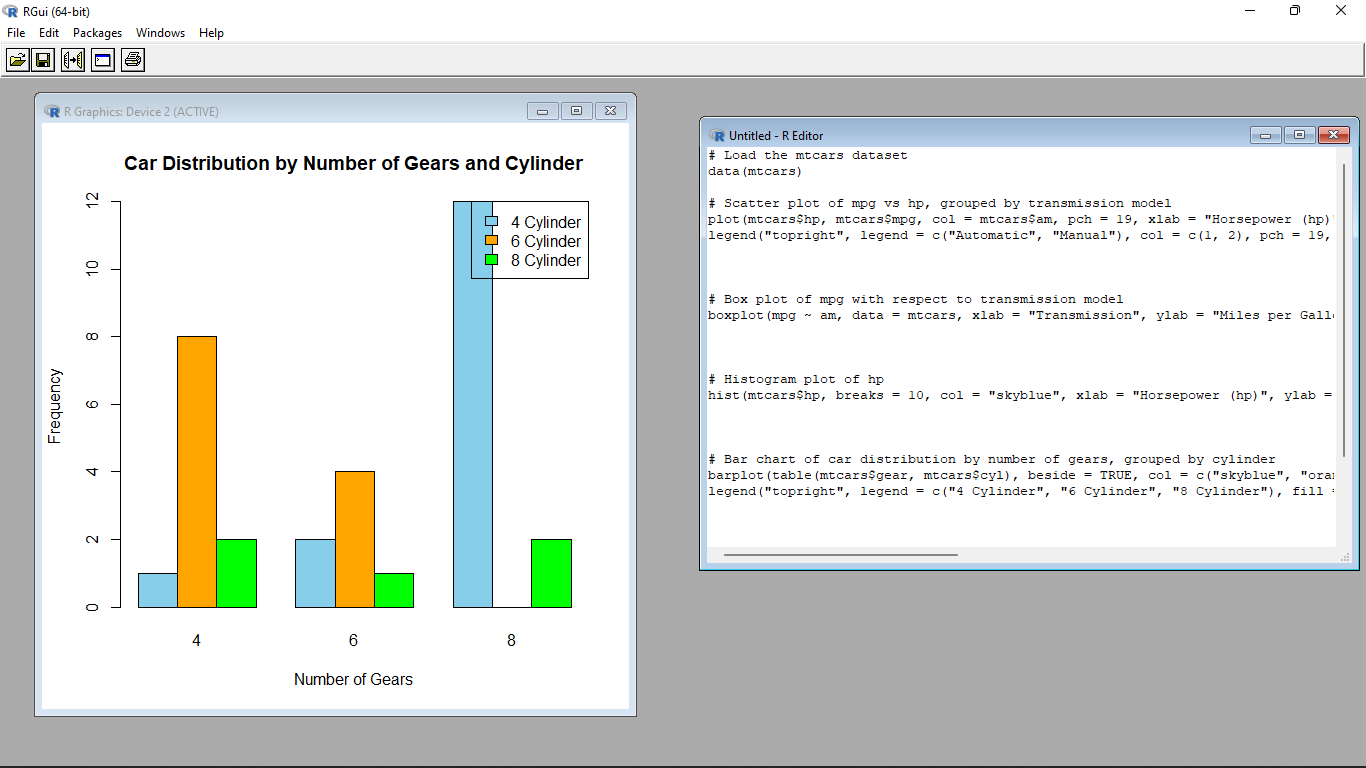
labels <- paste(names(gear\_counts), "Gears")

percentages <- round(gear\_counts / sum(gear\_counts) \* 100, 1)

pie(gear\_counts, labels = labels, main = "Percentage Distribution by Number of Gears")

legend("topright", legend = paste(labels, "-", percentages, "%"), cex = 0.8, fill = rainbow(length(gear\_counts)))





4. (i)Generate a multiple regression model using the built-in dataset mtcars. Establish the relationship

between "mpg" as a response variable with "disp","hp" and "wt" as predictor variables .

(ii)Plot the multiple regression line model with above model parameters.

(iii) Predict the mileage of the car with dsp=221, hp=102 and wt=2.91

**Input**:

# Load the mtcars dataset

data(mtcars)

# Create the multiple regression model

reg\_model <- lm(mpg ~ disp + hp + wt, data = mtcars)

# Summary of the regression model

summary(reg\_model)

# Scatter plot of actual mpg vs. predicted mpg

plot(mtcars$mpg, predict(reg\_model), xlab = "Actual MPG", ylab = "Predicted MPG", main = "Actual vs. Predicted MPG")

abline(0, 1, col = "red", lwd = 2) # Add a reference line with slope 1

# Create a data frame with predictor variables for prediction

new\_data <- data.frame(disp = 221, hp = 102, wt = 2.91)

# Predict the mileage for the new data using the regression model

predicted\_mpg <- predict(reg\_model, newdata = new\_data)

# Print the predicted mileage

print(paste("Predicted MPG:", predicted\_mpg))

