```
library(MASS)

vehicles <- read.csv("/Users/zissimilstein/Downloads/auto-mpg.csv")
str(vehicles)

# Convert horsepower to an integer
vehicles$horsepower <- as.integer(vehicles$horsepower)

# Split the data into training and testing
vehicles_train <- vehicles[1:300,]
vehicles_test <- vehicles[301:398,]



# Using the first 300 samples in the auto-mpg.csv, run a simple linear regression and multiple linear regression
# to determine the relationship between mpg and appropriate independent variable/(s).
# Report all the appropriate information regarding your regression.

full_model <- lm(mpg ~ ., data=vehicles_train)
summary(full_model)

# Multiple R-squared: .9789
# Adjusted R-squared:.9022
# Complete Linear Regression equation: Including the car names has the best R value but is very complicated to create a linear model
# since each car has it's own value. Additionally the testing set will have other car names that can't be predicted based on these.
# This is without the car names:

full_model <- lm(mpg ~ cylinder+displacement+horsepower+weight+acceleration+model.year+origin, data=vehicles_train)
summary(full_model)

# Multiple R-squared: .823
# Adjusted R-squared:.8187
# Complete Linear Regression equation: Y = 5.8118427 + -.4562389*B1 + 0.0101272*B2 + -0.0172493*B3 + -0.0053282*B4 + -0.0278409*B5 + 0.4439943*B6 + 0.9931335*B7

backward_model <- stepAIC(full_model, direction = "backward")
summary(backward_model)

# Multiple R-squared: .823
```

```
# Adjusted R-squared:.8193
# Complete Linear Regression equation: Y = 5.8118427 + -0.448111*B1 + 0.010352*B2 +
-0.015202*B3 + -0.005406*B4 + 0.444528*B5 + 0.996595*B6

my_model <- lm(mpg ~ weight+model.year+origin, data=vehicles_train)
summary(my_model)

# Multiple R-squared: .8206
# Adjusted R-squared:.8188
# Complete Linear Regression equation: Y = 3.2289898 + -0.0056852*B1 + 0.4585332*B2 +
0.8495008*B3


# For the remaining 98 samples in the dataset, use your best linear model(s) to predict each
automobile’s mpg and
# report how your predictions compare to the car’s actual reported mpg.

# Use my model to take in the values in the dataset and predict what the mpg would be:
predictions <- predict(my_model, vehicles_test)
# Store the actual mpg in actual outcomes
actual_outcomes <- vehicles_test$mpg

# Compare the two:
residuals <- actual_outcomes - predictions

summary(residuals)
# The closer to zero the better the model. Here our median residual is 3.995
summary(actual_outcomes)
summary(predictions)

#Residual Plot:

# Checks the distribution of the residuals

plot(residuals,
    main = "Residuals Plot",
    xlab = "Observation Number",
    ylab = "Residuals",
    col = "blue",
    pch = 16)
abline(h = 0, col = "red", lty = 2)

# It doesn't look to great, there are many above the zero line and not many below it
```

```
# Compares the residuals to the predictions:

plot(x = predictions, y = residuals,
    main = "Residuals vs. Fitted Values",
    xlab = "Fitted Values",
    ylab = "Residuals",
    col = "blue",
    pch = 16)
abline(h = 0, col = "red", lty = 2)


# Histogram:

hist(residuals,
    main = "Histogram of Residuals",
    xlab = "Residuals",
    ylab = "Frequency",
    col = "lightblue",
    border = "black")
# The histogram shows that majority of the residuals are between 0 and 5. While not perfect, it is a good average.
# As we saw in the chart there are many more towards the positive side than towards the negative.

# QQ plot

qqnorm(residuals)
qqline(residuals)

# The points of the residuls fit the line but start shifting off by the 1 point. There might be some outliers that are throwing the model off.
```