Q1:
We can build a linear regression model (M1) to estimate horsepower from cylinders in R, by using the lm() function. The output of the summary(M1) will provide information about the parameters of the model, including coefficients and statistical measures like R-squared, p-values, and more.

The linear regression model M1 provides valuable insights into the relationship between the number of cylinders in a car's engine and its horsepower. The model has two parameters: the intercept and the "Cylinders" coefficient. Firstly, the intercept (approximately 0.3822) represents the expected horsepower when a car has zero cylinders, which is a theoretical scenario. Secondly, the "Cylinders" coefficient (about 19.0220) signifies that for each additional cylinder added to the engine, the estimated horsepower is anticipated to increase by approximately 19.0220 units. These coefficients have been estimated with high precision, as indicated by their low standard errors and high t-values. The extremely low p-values associated with both parameters reinforce their statistical significance, highlighting that the number of cylinders indeed plays a crucial role in determining a car's horsepower.

Moreover, the model's overall performance is assessed through various statistics. The R-squared value of approximately 0.7106 implies that the model can explain around 71.06% of the variance in horsepower, indicating a strong fit. Additionally, the F-statistic, with a high value of 957.7 and an extremely low p-value, underscores the model's overall significance. When applying this model to estimate the horsepower of a car with 10 cylinders, we find an estimated value of approximately 190.6022. This showcases the practical utility of the model, allowing us to make informed predictions about a car's horsepower based on its engine's cylinder count.

```
> data <- read.csv("C:/Users/faiya/OneDrive - Texas Tech University/Texas tech course/fall 23/software analytics//HW1/auto.csv")
> # Summary of the model
> # Build the linear regression model M1
> M1 <- lm(horsepower ~ cylinders, data = data)
> # Summary of the model
> summary(M1)

Call:
lm(formula = horsepower ~ cylinders, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-62.558 -12.558  -2.558  11.530  77.442

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.3822     3.5226   0.108    0.914
cylinders    19.0220     0.6147  30.947   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.73 on 390 degrees of freedom
Multiple R-squared:  0.7106,    Adjusted R-squared:  0.7099
F-statistic: 957.7 on 1 and 390 DF,  p-value: < 2.2e-16

> |
```

```
>
> # Define the number of cylinders for the car
> new_data <- data.frame(cylinders = 10)
> # Predict the horsepower for a car with 10 cylinders using the model
> predicted_horsepower <- predict(M1, newdata = new_data)
> # Print the estimated horsepower
> cat("Estimated Horsepower for a Car with 10 Cylinders:", predicted_horsepower)
Estimated Horsepower for a Car with 10 Cylinders: 190.6022
```

Q2:

We can build a linear regression model (M2) to estimate horsepower from displacement in R, by using the lm() function. The output of the summary(M2) will provide you with information about the parameters of the model, including coefficients and statistical measures like R-squared, p-values, and more.

The linear regression model M2, represented by the equation Horsepower = 40.306108 + 0.330038 * Displacement, encompasses several important parameters. The intercept (40.306108) signifies the estimated horsepower when engine displacement is virtually zero, though this doesn't hold practical significance since real-world cars don't have zero displacement. It rather serves as a reference point for negligible displacement. The coefficient for the "displacement" variable (0.330038) holds greater importance. It denotes the anticipated change in estimated horsepower for each additional cubic inch of engine displacement. In simpler terms, for every extra cubic inch, you can expect a horsepower increase of approximately 0.330038 units.

The statistical aspects of this model further reveal its robustness. The low p-value (<<0.001) associated with the "displacement" coefficient implies its high statistical significance. A high t-value (40.13) corroborates this significance, indicating that the relationship between engine displacement and horsepower is unlikely due to random chance. The R-squared value (0.8051) reflects the model's goodness of fit, indicating that approximately 80.51% of the variance in horsepower can be explained by the model, signifying a strong explanatory capability.

To estimate the horsepower of a car with a 200 cubic inch engine displacement using this model, simply apply the formula: Horsepower = 40.306108 + (0.330038 * 200). This calculation yields the estimated horsepower for such a car, bridging the gap between statistical modeling and practical automotive insights.

```
> # Build the linear regression model
> model_M2 <- lm(horsepower ~ displacement, data = data)
> # Summarize the model
> summary(model_M2)

Call:
lm(formula = horsepower ~ displacement, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-50.819 -10.695  -0.819   8.676  64.742

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  40.306108   1.815099   22.21   <2e-16 ***
displacement  0.330038   0.008223   40.13   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.02 on 390 degrees of freedom
Multiple R-squared:  0.8051,    Adjusted R-squared:  0.8046
F-statistic:  1611 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
> #q2
> # Given engine displacement
> displacement_200 <- 200
> # Estimate horsepower
> estimated_horsepower <- 40.306108 + (0.330038 * displacement_200)
> # Print the estimated horsepower
> print(estimated_horsepower)
[1] 106.3137
```

Q3:

In R, `summary(model)` provides a summary of the linear regression model `model`. `$r.squared` extracts the R-squared value from the summary, which quantifies the proportion of variance in the dependent variable (horsepower) explained by the independent variable (cylinders).

To compare the goodness of fit between models M1 and M2, we can look at their respective R-squared values. The R-squared value indicates the proportion of the variance in the dependent variable (horsepower) that is explained by the model. A higher R-squared value suggests a better goodness of fit because it means that a larger portion of the variance is accounted for by the model.

Here are the R-squared values for both models:

M1 (Horsepower ~ Cylinders):

- Multiple R-squared: 0.7106

M2 (Horsepower ~ Displacement):

- Multiple R-squared: 0.8051

Comparing these R-squared values, we can see that model M2 (Horsepower ~ Displacement) has a higher R-squared value (0.8051) compared to model M1 (0.7106). This indicates that M2 explains a larger proportion of the variance in horsepower, suggesting that it has a better goodness of fit for predicting horsepower based on engine displacement.

```
> # Extract R-squared values
> R_squared_M1 <- summary(M1)$r.squared


> R_squared_M2 <- summary(model_M2)$r.squared
> # Compare R-squared values
> if (R_squared_M2 > R_squared_M1) {
+   cat("Model M2 has a better goodness of fit (higher R-squared) than Model M1.\n")
+ } else if (R_squared_M1 > R_squared_M2) {
+   cat("Model M1 has a better goodness of fit (higher R-squared) than Model M2.\n")
+ } else {
+   cat("Both models have the same R-squared value and provide similar goodness of fit.\n")
+ }
Model M2 has a better goodness of fit (higher R-squared) than Model M1.
```

In the analysis, a seed was set for reproducibility using the `set.seed(123)` function to ensure that random sampling remains consistent across runs. Then, 20 random row indices were sampled from the dataset using `sample(1:nrow(data), 20)` to create a representative sample of 20 data points.

Next, the values of cylinders and displacement were extracted from this sample using `sample_data$cylinders` and `sample_data$displacement`. These extracted values represented the independent variables for models M1 and M2.

Using the pre-trained linear regression models M1 and M2, the estimated horsepower for the 20 sampled data points was calculated. This was done separately for each model by applying the `predict()` function with the appropriate independent variables.

To compare the performance of the two models, the absolute errors were calculated for both models. The absolute error for each data point was obtained by taking the absolute difference between the estimated and actual horsepower values, thus quantifying the modeling errors.

Finally, a paired t-test was conducted to statistically compare the absolute errors between models M1 and M2. The `t.test()` function with the `paired = TRUE` argument was used to perform this analysis. The results of the paired t-test were printed, including the p-value, which determined whether there was a significant difference in modeling errors between the two models based on the sampled data.
Paired t-test results:

- t-statistic: 2.1018
- Degrees of freedom: 19
- p-value: 0.04914
- 95% confidence interval for the difference in means: 0.016 to 7.611
- Mean of the absolute error differences: 3.8135

There is statistically significant evidence to suggest that model M2 performed differently from model M1 in terms of predicting horsepower. On average, model M2 had an absolute error that was about 3.8135 units different from that of model M1. However, the practical significance of this difference would require further consideration.

```
> # Set a seed for reproducibility
> set.seed(123)
> # Randomly sample 20 rows from the dataset
> sample_indices <- sample(1:nrow(data), 20)
> sample_data <- data[sample_indices, ]
> # Extract cylinders and displacement values from the sample
> sample_cylinders <- sample_data$cylinders
> sample_displacement <- sample_data$displacement
> # Use models M1 and M2 to estimate horsepower for the samples
> estimated_horsepower_M1 <- predict(M1, newdata = data.frame(cylinders = sample_cylinders))
> estimated_horsepower_M2 <- predict(model_M2, newdata = data.frame(displacement = sample_displacement))
> # Calculate the absolute errors for both models
> error_M1 <- abs(estimated_horsepower_M1 - sample_data$horsepower)
> error_M2 <- abs(estimated_horsepower_M2 - sample_data$horsepower)
> # Perform a paired t-test to compare the errors
> t_test_errors <- t.test(error_M1, error_M2, paired = TRUE)
> # Print the t-test results
> cat("Paired T-Test Results for Error Comparison:\n")
Paired T-Test Results for Error Comparison:
> print(t_test_errors)

        Paired t-test

data:  error_M1 and error_M2
t = 2.1018, df = 19, p-value = 0.04914
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.01598068 7.61103410
sample estimates:
mean of the differences
               3.813507
```

Q4:

We can build a linear regression model (M3) to estimate mpg from cylinders in R, by using the lm() function. The output of the summary(M3) will provide you with information about the parameters of the model, including coefficients and statistical measures like R-squared, p-values, and more.

Model M3, a linear regression model, consists of an intercept (42.9155) and a coefficient for the "cylinders" variable (-3.5581). The intercept represents an impractical baseline mpg when there are no cylinders. The coefficient for "cylinders" signifies that for each additional cylinder in a car's engine, the estimated mpg is expected to decrease by approximately 3.5581 units. Both coefficients are highly statistically significant with low p-values. The model explains around 60.47% of the variance in mpg. To estimate the mpg of a car with 10 cylinders, we can use the formula mpg = Intercept + (Cylinders * 10), yielding the estimated miles per gallon for a 10-cylinder car. In this case, the Estimated MPG for a Car with 10 Cylinders: 7.3345

```
> # Build linear regression model M3
> model_M3 <- lm(formula = mpg ~ cylinders, data = data)
> # Print model summary
> summary(model_M3)

Call:
lm(formula = mpg ~ cylinders, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-14.2413  -3.1832  -0.6332   2.5491  17.9168

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  42.9155     0.8349   51.40   <2e-16 ***
cylinders    -3.5581     0.1457  -24.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.914 on 390 degrees of freedom
Multiple R-squared:  0.6047,    Adjusted R-squared:  0.6037
F-statistic: 596.6 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
> # Define the number of cylinders (10 in this case)
> cylinders <- 10
> # Use model M3 to predict mpg
> estimated_mpg <- 42.9155 + (-3.5581 * cylinders)
> # Print the estimated mpg
> cat("Estimated MPG for a Car with 10 Cylinders:", estimated_mpg, "\n")
Estimated MPG for a Car with 10 Cylinders: 7.3345
>
```

Q5:

We can build a linear regression model (M4) to estimate mpg from weight in R, by using the lm() function.

The linear regression model M4 is represented by the formula:
mpg = 46.216524 - 0.007647 * weight
In this model, the intercept (46.216524) represents the estimated miles per gallon (mpg) when the weight of the car is zero, which is not practically meaningful. The coefficient for "weight" (-0.007647) signifies the linear relationship between the weight of the car and its miles per gallon. For each additional pound of weight, the estimated mpg is expected to decrease by approximately 0.007647 units. Both the intercept and the weight coefficient are highly statistically significant, with very low p-values (< 0.001). The model explains approximately 69.26% of the variance in mpg, indicating a reasonably good fit to the data.

Comparing models M3 and M4 based on their goodness of fit statistics, M4 appears to be a better model for predicting mpg compared to M3. This conclusion is drawn from the higher multiple R-squared value

of M4 (0.6926) compared to M3 (0.6047). A higher R-squared value indicates that M4 explains a larger proportion of the variance in mpg, suggesting that it provides a better fit to the data.

The reason why using the weight of cars to predict mpg might be better than using engine size (cylinders) can be attributed to the fact that weight is a more direct and continuous measure of a car's physical characteristics that can impact fuel efficiency. Weight affects factors such as aerodynamics, rolling resistance, and overall engine load, all of which can significantly influence a car's mpg. In contrast, while the number of cylinders (cylinders) is a relevant indicator of engine size, it might not capture the full spectrum of factors affecting fuel efficiency. Therefore, M4, which uses weight as a predictor, may provide a better fit due to its ability to capture a broader range of variables that influence mpg.

```
> #q5
> # Build linear regression model M4
> model_M4 <- lm(mpg ~ weight, data = data)
> # Print the summary of the model
> summary(model_M4)

Call:
lm(formula = mpg ~ weight, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-11.9736  -2.7556  -0.3358   2.1379  16.5194

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.216524   0.798673   57.87   <2e-16 ***
weight      -0.007647   0.000258  -29.64   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.333 on 390 degrees of freedom
Multiple R-squared:  0.6926,    Adjusted R-squared:  0.6918
F-statistic: 878.8 on 1 and 390 DF,  p-value: < 2.2e-16
```