Week 9 - Evaluation of Regression Models

Introduction:

Welcome to this unit on methods of evaluating regression models. Regression models are essential in mathematical modeling and data analysis, and their performance needs to be assessed rigorously. Without proper evaluation, researchers may misinterpret the relationships between variables. They might erroneously conclude that certain predictors have a significant impact on the outcome when they do not or vice versa. Inadequately evaluated regression models may produce inaccurate predictions. If the model does not capture the underlying relationships between variables effectively, its predictive performance may be poor, leading to unreliable forecasts or estimates. In this unit, we will explore various evaluation methods and metrics used to assess the quality and accuracy of regression models.

Importance of Model Evaluation:

Model evaluation is critical to determine how well a regression model fits the data and makes predictions. It helps in selecting the best model among competing models and guides model refinement. Accurate model evaluation is essential for making informed decisions in various fields, including economics, engineering, and the sciences.

Common Evaluation Metrics:

Several metrics are commonly used to evaluate regression models:

1. Mean Squared Error (MSE):

MSE measures the average squared difference between predicted and actual values. It penalizes larger errors more than smaller ones, making it sensitive to outliers. The formula for MSE is:

$$MSE = \frac{\sum(y_i - \bar{y})^2}{n}$$

where y_i is the actual value, $\bar{y}$ is the mean of actual values, and n is the number of data points.

2. Root Mean Squared Error (RMSE):

RMSE is the square root of MSE and is in the same units as the dependent variable. It provides a more interpretable measure of error.

$$RMSE = \sqrt{MSE}$$

3. Mean Absolute Error (MAE):

MAE measures the average absolute difference between predicted and actual values. It is less sensitive to outliers compared to MSE.

$$MAE = \frac{\sum |y_i - \bar{y}|}{n}$$

where ŷ is the predicted value.

4. R-squared (R^2):

R^2 represents the proportion of variance in the dependent variable explained by the model. It ranges from 0 to 1, where a higher value indicates a better fit.

% Calculate SSE and SST

$$R^2 = 1 - (SSE/SST)$$

where SSE is the sum of squared errors and SST is the total sum of squares.

5. Adjusted R-squared (Adjusted R^2):

Adjusted R^2 adjusts R^2 for the number of predictors in the model. It penalizes the addition of unnecessary variables.

$$AdjustedR^2 = 1 - [(1 - R^2) * ((n - 1)/(n - p - 1))]$$

where n is the number of data points and p is the number of predictors.

6. Multiple R squared.

Model Validation Techniques in R:

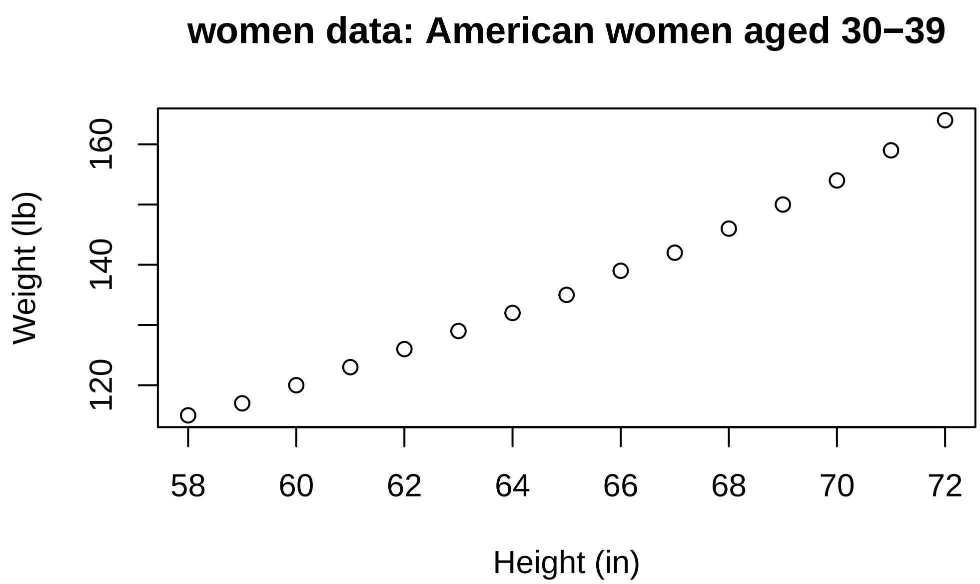
Summary Statistics:

In this example, we will do an exploration of a dataset which includes American women's height and weight. The question we are asking, is can we predict a woman's height given her weight?

```
 # You may need to install the datasets library (by using install.packages('datasets'))
library(datasets)
```

In this example, it's worth plotting the data to get a sense of whether there may be a trend or not.

```
plot(women, xlab = "Height (in)", ylab = "Weight (lb)",
     main = "women data: American women aged 30-39")
```



The summary() function provides a summary of the regression model, including coefficients, standard errors, and metrics for evaluating the effectiveness of the model.

Example:

```
summary(lm_model)
```

Below we have created a linear regression model, where we use weight as the dependent variable and height as the independent variable.

```
lmHeight = lm(height~weight, data = women) #Create the linear regression
summary(lmHeight) #Review the results
```

```
Call:
lm(formula = height ~ weight, data = women)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.83233 -0.26249  0.08314  0.34353  0.49790 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 25.723456   1.043746   24.64 2.68e-12 ***
weight       0.287249   0.007588   37.85 1.09e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.44 on 13 degrees of freedom
Multiple R-squared:  0.991,	Adjusted R-squared:  0.9903 
F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

The most important results are, our Multiple R-Squared and Adjusted R

Coefficient of Determination (R-squared):

R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables in the model. Higher values indicate better fit.

If we just want to isolate this statistic, we can call the following function.

Example:

```
summary(lm_model)$r.squared
```

Adjusted R-squared:

Adjusted R-squared penalizes the inclusion of unnecessary predictors in the model and provides a more conservative measure of model fit, especially for models with multiple predictors.

If we just want to isolate this statistic, we can call the following function.

Example:

```
summary(lm_model)$adj.r.squared
```

However, these statistics are included in the summary report given above.

Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE):

These metrics quantify the average difference between observed and predicted values. Lower values indicate better predictive accuracy.

Example

```
RMSE: sqrt(mean(residuals(lm_model)^2))
```

```r
rmse = sqrt(mean(residuals(lmHeight)^2))
rmse
```

```
[1] 0.4096541
```

Multiple Linear Regression

Generally speaking, modelling is going to be concerned with looking at more than one factor. In this case, what we can do is use multiple linear regression to assess whether multiple factors are more useful to predict a value.

In this case, we are looking at predicting a cars price based on some characteristics of the car.

```r
automobile <- read.csv("~/automobile.csv")
```

For example, we could look at whether city.mpg is a good predictor of price but doing a simple linear regression model.

```r
model = lm(price ~ `city.mpg`, data = automobile) #Create the linear regression
summary(model) #Review the results
```

```

Call:
lm(formula = price ~ city.mpg, data = automobile)

Residuals:
   Min     1Q Median     3Q    Max 
 -7623  -3302  -1698   1632  22078 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 35802.44    1690.14   21.18   <2e-16 ***
city.mpg     -891.42      64.79  -13.76   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5739 on 193 degrees of freedom
Multiple R-squared:  0.4951,	Adjusted R-squared:  0.4925 
F-statistic: 189.3 on 1 and 193 DF,  p-value: < 2.2e-16
```

Not so great with a multiple R squared of 0.4993 and an adjusted R squared of 0.4967. Maybe we can use another variable and see if this helps the model. Lets add highway.mpg. Note, all we do to add another variable is use the + sign and then the next variable.

```r
model = lm(price ~ `city.mpg` + `highway.mpg`, data = automobile) #Create the linear regre
summary(model) #Review the results
```

```

Call:
lm(formula = price ~ city.mpg + highway.mpg, data = automobile)

Residuals:
   Min     1Q Median     3Q    Max 
 -8441  -3476  -1111   1108  20397 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  39172.6     2015.2  19.439  < 2e-16 ***
```

```
city.mpg      -129.8      266.7  -0.487  0.62687
highway.mpg   -735.3      250.0  -2.941  0.00368 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5628 on 192 degrees of freedom
Multiple R-squared:  0.5169,    Adjusted R-squared:  0.5119
F-statistic: 102.7 on 2 and 192 DF,  p-value: < 2.2e-16
```

This has improved our multiple R squared to 0.5183 and adjusted R squared to 0.5132. Maybe we can add horsepower to our multiple linear regression model.

```r
model = lm(price ~ `city.mpg` + `highway.mpg` + `horsepower`, data = automobile) #Create t
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower, data = automobile)

Residuals:
    Min      1Q  Median      3Q     Max
-5480.2  -752.9   -13.0   387.1  7002.2

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   14649.69    2473.37   5.923 2.45e-08 ***
city.mpg       -189.58     151.42  -1.252 0.212705
highway.mpg     -22.40     132.05  -0.170 0.865567
horsepower100  1060.15    1934.18   0.548 0.584511
horsepower101  5849.27    1776.15   3.293 0.001262 **
horsepower102   -39.57    1673.98  -0.024 0.981178
horsepower106 13354.16    2520.69   5.298 4.60e-07 ***
horsepower110  4700.79    1540.68   3.051 0.002742 **
horsepower111  3399.03    1725.39   1.970 0.050870 .
horsepower112   346.95    2029.65   0.171 0.864524
horsepower114  6650.92    1612.53   4.125 6.43e-05 ***
horsepower115  6705.52    2426.64   2.763 0.006515 **
horsepower116  1287.99    1507.62   0.854 0.394429
horsepower120  7837.08    2355.60   3.327 0.001129 **
horsepower121 12086.39    1764.20   6.851 2.32e-10 ***
horsepower123 18475.04    1780.77  10.375  < 2e-16 ***
horsepower134 10762.91    2400.95   4.483 1.55e-05 ***
```

```
horsepower140 12896.14    2472.68   5.215 6.67e-07 ***
horsepower142  7450.31    2382.32   3.127 0.002158 **
horsepower143 11575.08    2355.60   4.914 2.53e-06 ***
horsepower145  3032.89    1673.70   1.812 0.072179 .
horsepower152  3013.72    1824.08   1.652 0.100803
horsepower154  6034.69    2359.63   2.557 0.011640 *
horsepower155 23406.76    2137.49  10.951  < 2e-16 ***
horsepower156  5304.68    1981.65   2.677 0.008345 **
horsepower160  7803.85    1639.14   4.761 4.87e-06 ***
horsepower161  5862.68    1981.65   2.958 0.003647 **
horsepower162  7750.94    2003.05   3.870 0.000168 ***
horsepower175  5992.89    2389.66   2.508 0.013323 *
horsepower176 22519.58    2074.71  10.854  < 2e-16 ***
horsepower182 25116.56    1856.97  13.526  < 2e-16 ***
horsepower184 31542.81    2182.14  14.455  < 2e-16 ***
horsepower200  8787.33    2396.02   3.667 0.000350 ***
horsepower207 23661.13    1800.60  13.141  < 2e-16 ***
horsepower262 24195.62    2517.45   9.611  < 2e-16 ***
horsepower48    598.70    2982.21   0.201 0.841189
horsepower52   1280.10    2193.68   0.584 0.560495
horsepower55   2100.34    2903.89   0.723 0.470747
horsepower56    947.73    2174.11   0.436 0.663588
horsepower58   2328.26    3091.47   0.753 0.452677
horsepower60  -1105.91    2652.10  -0.417 0.677341
horsepower62  -1131.42    1681.93  -0.673 0.502286
horsepower68   -699.62    1607.03  -0.435 0.663997
horsepower69   -915.30    1621.31  -0.565 0.573313
horsepower70   -101.05    1659.60  -0.061 0.951537
horsepower72  10444.83    2414.78   4.325 2.93e-05 ***
horsepower73   -307.21    1835.50  -0.167 0.867325
horsepower76  -1124.30    1744.26  -0.645 0.520290
horsepower78  -2665.21    2369.89  -1.125 0.262733
horsepower82   -910.19    1698.67  -0.536 0.592954
horsepower84    881.14    1632.75   0.540 0.590309
horsepower85   -547.82    1772.11  -0.309 0.757695
horsepower86    240.62    1699.44   0.142 0.887615
horsepower88   -307.13    1574.46  -0.195 0.845629
horsepower90   1424.12    1786.63   0.797 0.426783
horsepower92   1562.21    1731.51   0.902 0.368532
horsepower94    969.15    1934.18   0.501 0.617134
horsepower95   6533.21    1640.59   3.982 0.000111 ***
horsepower97   2079.74    1623.08   1.281 0.202249
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1907 on 136 degrees of freedom
Multiple R-squared:  0.9607,    Adjusted R-squared:  0.944
F-statistic: 57.35 on 58 and 136 DF,  p-value: < 2.2e-16
```

We have improved this more dramatically this time, and our multiple R squared value rises to 0.6803 and adjusted R squared value rises to 0.6753.

We have dealt so far with just numerical data, lets see what happens if we use some categorical data by adding make to our linear regression.

```r
model = lm(price ~ `city.mpg` + `highway.mpg` + horsepower + make, data = automobile) #Cre
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower + make,
    data = automobile)

Residuals:
    Min      1Q  Median      3Q     Max
-5504.8  -496.6     0.0   358.1  6123.4

Coefficients: (5 not defined because of singularities)
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       14007.84    2418.31   5.792 5.69e-08 ***
city.mpg           -256.50     150.46  -1.705 0.090830 .
highway.mpg          47.96     125.02   0.384 0.701944
horsepower100     -1286.45    3267.63  -0.394 0.694505
horsepower101      1200.92    3999.71   0.300 0.764504
horsepower102       651.57    2820.66   0.231 0.817709
horsepower106     12088.96    3563.28   3.393 0.000938 ***
horsepower110      2054.42    2810.73   0.731 0.466254
horsepower111      5081.33    1670.00   3.043 0.002881 **
horsepower112       389.20    2882.23   0.135 0.892812
horsepower114      5127.48    3216.33   1.594 0.113522
horsepower115      2322.90    3329.43   0.698 0.486724
horsepower116      1955.80    2707.41   0.722 0.471461
horsepower120     10619.17    3358.39   3.162 0.001985 **
horsepower121      6234.17    4285.37   1.455 0.148348
horsepower123     18830.25    1556.11  12.101  < 2e-16 ***
```

horsepower134	9243.78	3609.56	2.561	0.011679	*
horsepower140	8587.31	3350.85	2.563	0.011622	*
horsepower142	4661.28	3391.26	1.374	0.171849	
horsepower143	11588.83	2039.23	5.683	9.45e-08	***
horsepower145	5539.82	2870.07	1.930	0.055941	.
horsepower152	3107.97	2803.65	1.109	0.269843	
horsepower154	6118.78	2042.77	2.995	0.003331	**
horsepower155	23852.94	1892.07	12.607	< 2e-16	***
horsepower156	5263.73	2964.68	1.775	0.078354	.
horsepower160	6964.37	2744.54	2.538	0.012445	*
horsepower161	5821.73	2964.68	1.964	0.051878	.
horsepower162	6235.24	3436.49	1.814	0.072111	.
horsepower175	6217.70	2072.58	3.000	0.003284	**
horsepower176	22828.48	1836.04	12.434	< 2e-16	***
horsepower182	19328.42	4338.77	4.455	1.90e-05	***
horsepower184	31995.85	1947.11	16.432	< 2e-16	***
horsepower200	8836.97	3124.00	2.829	0.005479	**
horsepower207	23681.74	1578.70	15.001	< 2e-16	***
horsepower262	24511.39	2227.27	11.005	< 2e-16	***
horsepower48	214.89	3538.71	0.061	0.951678	
horsepower52	-1246.34	3214.01	-0.388	0.698862	
horsepower55	2124.15	3370.47	0.630	0.529746	
horsepower56	779.74	2968.31	0.263	0.793242	
horsepower58	-99.44	4412.26	-0.023	0.982056	
horsepower60	-3425.47	4109.80	-0.833	0.406226	
horsepower62	-1291.67	2743.21	-0.471	0.638595	
horsepower68	1340.49	2823.16	0.475	0.635778	
horsepower69	-612.37	2524.82	-0.243	0.808775	
horsepower70	-330.04	2743.93	-0.120	0.904462	
horsepower72	13185.69	3381.24	3.900	0.000159	***
horsepower73	1723.99	2297.85	0.750	0.454566	
horsepower76	-3459.28	3731.64	-0.927	0.355780	
horsepower78	-5126.08	3836.87	-1.336	0.184075	
horsepower82	2175.20	2146.58	1.013	0.312941	
horsepower84	3779.89	2962.96	1.276	0.204521	
horsepower85	-2899.20	2994.19	-0.968	0.334855	
horsepower86	-2181.86	3747.71	-0.582	0.561534	
horsepower88	1293.11	2804.39	0.461	0.645561	
horsepower90	-863.08	3052.90	-0.283	0.777887	
horsepower92	1458.60	2768.79	0.527	0.599306	
horsepower94	3933.06	2331.64	1.687	0.094236	.
horsepower95	3892.47	3046.55	1.278	0.203834	
horsepower97	424.45	2796.43	0.152	0.879613	

```
makeaudi           4681.22    2618.14   1.788 0.076300 .
makebmw            5977.49    4012.84   1.490 0.138956
makechevrolet       442.07    3005.76   0.147 0.883320
makedodge         -2341.33    2586.02  -0.905 0.367078
makehonda          2549.45    3510.03   0.726 0.469048
makeisuzu          2668.50    3319.86   0.804 0.423104
makejaguar              NA         NA      NA       NA
makemazda         -2768.35    2657.86  -1.042 0.299705
makemercedes-benz       NA         NA      NA       NA
makemercury             NA         NA      NA       NA
makemitsubishi    -2256.97    2513.42  -0.898 0.371001
makenissan          111.68    2300.07   0.049 0.961354
makepeugot         2946.91    2794.62   1.054 0.293775
makeplymouth      -2298.36    2596.65  -0.885 0.377859
makeporsche             NA         NA      NA       NA
makerenault             NA         NA      NA       NA
makesaab           1429.11    2515.69   0.568 0.571043
makesubaru        -2831.79    1713.68  -1.652 0.101055
maketoyota          299.22    2475.04   0.121 0.903975
makevolkswagen     2408.00    2640.16   0.912 0.363564
makevolvo          1747.37    2946.61   0.593 0.554289
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1643 on 120 degrees of freedom
Multiple R-squared:  0.9743,    Adjusted R-squared:  0.9584
F-statistic: 61.38 on 74 and 120 DF,  p-value: < 2.2e-16
```

So some things to note with this output. Firstly, our multiple R squared value rises again to 0.9179, and our adjusted R squared value rises to 0.908. But our output looks different. We can see that each make of car is represented, and that some have some * next to their entries, which denotes significance (we can see that these models are a lot better (significant) than others).

Lets check to see if this is right. Remember, we can create a subset of values by using the which command. Lets put all mercedez cars into a smaller dataframe. To compare, lets also put all hondas into a smaller dataframe.

```
# This code takes all rows where make is mercedes-benz and puts it into a new dataframe ca
mercedes <- automobile[which(automobile$make == 'mercedes-benz'),]
# This code takes all rows where make is honda and puts it into a new dataframe called hon
honda <- automobile[which(automobile$make == 'honda'),]
```

Now lets run our linear regression model, just for all mercedes cars and all honda cars.

```r
model = lm(price ~ `city.mpg` + `highway.mpg` + horsepower, data = mercedes) #Create the l
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower, data = mercedes)

Residuals:
   59    60    61    62    63    64    65    66 
-2842  -146  -218  3206  -436   436 -2220  2220 

Coefficients: (2 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)      64192       4479  14.331 2.98e-05 ***
city.mpg        -23734       9725  -2.440   0.0586 .  
highway.mpg      19454       8549   2.276   0.0719 .  
horsepower155       NA         NA      NA       NA    
horsepower184       NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2394 on 5 degrees of freedom
Multiple R-squared:  0.9112,    Adjusted R-squared:  0.8757 
F-statistic: 25.65 on 2 and 5 DF,  p-value: 0.002351
```

```r
model = lm(price ~ `city.mpg` + `highway.mpg` + horsepower, data = honda) #Create the line
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower, data = honda)

Residuals:
   Min     1Q Median     3Q    Max 
 -1138      0      0     67   1262 

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)    25852.7    10375.5   2.492   0.0471 *
city.mpg        -806.0      620.3  -1.299   0.2415
highway.mpg      149.8      310.1   0.483   0.6463
horsepower101   2243.3     1272.7   1.763   0.1284
horsepower58   12033.7     9498.1   1.267   0.2521
horsepower60    3884.7     5116.7   0.759   0.4765
horsepower76     297.8     1798.8   0.166   0.8740
horsepower86        NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 744.3 on 6 degrees of freedom
Multiple R-squared:  0.9348,    Adjusted R-squared:  0.8697
F-statistic: 14.34 on 6 and 6 DF,  p-value: 0.002504
```

We can see that the model is better for mercedes cars 0.9112 and 0.8757, than for honda cars 0.8722 and 0.8297.

Practical Questions

The main purpose of this week, is to see how we evaluate linear regression models. In our dataset, we have a number of different variables described below.

make: alfa-romero, audi, bmw, chevrolet, dodge, honda, isuzu, jaguar, mazda, mercedes-benz, mercury, mitsubishi, nissan, peugot, plymouth, porsche, renault, saab, subaru, toyota, volkswagen, volvo fuel-type: diesel, gas. aspiration: std, turbo. num-of-doors: four, two. body-style: hardtop, wagon, sedan, hatchback, convertible. drive-wheels: 4wd, fwd, rwd. engine-location: front, rear. wheel-base: continuous from 86.6 120.9. length: continuous from 141.1 to 208.1. width: continuous from 60.3 to 72.3. height: continuous from 47.8 to 59.8. curb-weight: continuous from 1488 to 4066. engine-type: dohc, dohcv, l, ohc, ohcf, ohcv, rotor. num-of-cylinders: eight, five, four, six, three, twelve, two. engine-size: continuous from 61 to 326. fuel-system: 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi. bore: continuous from 2.54 to 3.94. stroke: continuous from 2.07 to 4.17. compression-ratio: continuous from 7 to 23. horsepower: continuous from 48 to 288. peak-rpm: continuous from 4150 to 6600. city-mpg: continuous from 13 to 49. highway-mpg: continuous from 16 to 54. price: continuous from 5118 to 45400.

Create models to answer the following questions in a R Markdown document.

1. What is the best single variable for predicting price?
2. Rank, in order, each variable according to Adjusted R squared and also Root Mean Squared. Are the orders different?

3. Is there a combination of variables that improves our model? HINT: Combine the single variables that scored best in various ways to see if you can improve the model.
4. Now that you have found your best combination, answer the following questions:
5. Does the model perform better for certain makes of car?
6. Does the model work better for standard or turbo cars?
7. Does the model work for cars with two or four doors, or is there no difference?

Conclusion:

Evaluating regression models is essential to determine their accuracy, reliability, and suitability for a given problem. Metrics like MSE, RMSE, MAE, R^2, and adjusted R^2 provide valuable insights into a model's performance. Model validation techniques and residual analysis help ensure robust and interpretable regression models.