```
> summary(model_1)

Call:
lm(formula = price ~ highwaympg + citympg + peakrpm + horsepower +
    compressionratio + stroke + bore + enginesize + curbweight +
    height + width + length + wheelbase)

Residuals:
     Min       1Q   Median       3Q      Max
-11674.3  -1620.6      4.4   1543.8  13809.5

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -63633.132  16432.763  -3.872 0.000151 ***
highwaympg          287.178    164.689   1.744 0.082918 .
citympg            -305.769    182.951  -1.671 0.096407 .
peakrpm               2.117      0.677   3.127 0.002063 **
horsepower           37.280     18.211   2.047 0.042109 *
compressionratio    247.104     86.596   2.854 0.004834 **
stroke            -2962.854    797.179  -3.717 0.000270 ***
bore               -838.703   1217.519  -0.689 0.491802
enginesize          127.760     15.208   8.401 1.33e-14 ***
curbweight            1.639      1.739   0.942 0.347275
height              326.301    143.014   2.282 0.023691 *
width               632.908    258.021   2.453 0.015128 *
length              -84.382     57.919  -1.457 0.146894
wheelbase            60.048    104.790   0.573 0.567342
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3168 on 179 degrees of freedom
Multiple R-squared:  0.857,	Adjusted R-squared:  0.8466
F-statistic: 82.52 on 13 and 179 DF,  p-value: < 2.2e-16
```

#it can be seen that p-value of the F-statistics is < 2.26-16, which is highly significant. This means that, at least, one of the 13 explanatory variables is significantly related to the responsive variable

#Multiple R-squared is 0.857, which indicate a pretty well fit to the data

#based on the coefficients table we can see that

```
      the coefficient with *** are statistically significant
      the coefficient with **  are somewhat statistically significant
      the coefficient with *   are not very statistically significant
      the coefficient with     are not statistically significant
```

in summary, changing in engine size and stroke are significantly associated to changes in price, changing in compression ratio and peak rpm are somewhat associate to changes in price while changing in rest variables are not significantly associated with price.

```
> model2 <- lm(price ~ stroke + enginesize, data = df)
```

```
> summary(model2)

Call:
lm(formula = price ~ stroke + enginesize, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-13553.2  -1770.5   -260.7   1128.9  14074.0 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1126.508   2730.551  -0.413  0.68040    
stroke      -2544.482    853.327  -2.982  0.00324 ** 
enginesize    177.001      6.472  27.350  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3643 on 190 degrees of freedom
Multiple R-squared:  0.7993,	Adjusted R-squared:  0.7972 
F-statistic: 378.4 on 2 and 190 DF,  p-value: < 2.2e-16

#based on this model, we can tell that engine size have more significantly effect on the price



> model3 <- lm(price ~ enginesize, data = df)
> summary(model3)

Call:
lm(formula = price ~ enginesize, data = df)

Residuals:
   Min     1Q Median     3Q    Max 
-11490  -2031   -193   1460  14050 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8862.79     868.66   -10.2   <2e-16 ***
enginesize    172.86       6.45    26.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3717 on 191 degrees of freedom
Multiple R-squared:  0.7899,	Adjusted R-squared:  0.7888 
F-statistic: 718.2 on 1 and 191 DF,  p-value: < 2.2e-16
```

```
> boot.control <- trainControl(method = "boot", number = 500)
> model_1_boot <- train(price ~., data = df, method = "lm", trControl = boot.control)
> print(model_1_boot)
```

```
Linear Regression

193 samples
 13 predictor

No pre-processing
Resampling: Bootstrapped (500 reps)
Summary of sample sizes: 193, 193, 193, 193, 193, 193, ...
Resampling results:

  RMSE      Rsquared  MAE
  3513.489  0.821008  2458.846

Tuning parameter 'intercept' was held constant at a value of TRUE
```

The output shows the average model_1 performance across 500 resamples

RMSE(root mean squared error) measures the model prediction error, the lower the better the model.

MAE(Mean Absolute Error) measures the model prediction error, the lower the better the model.

The R squared represent the proportion of variation in the outcome explained by the predictor variables included in the model. The higher the better.

```
> boot.control <- trainControl(method = "boot", number = 500)
> model_3_boot <- train(price ~ enginesize, data = df, method = "lm",
trControl = boot.control)
> print(model_2_boot)
Linear Regression

193 samples
  2 predictor

No pre-processing
Resampling: Bootstrapped (500 reps)
Summary of sample sizes: 193, 193, 193, 193, 193, 193, ...
Resampling results:

  RMSE      Rsquared   MAE
  3740.456  0.7973457  2582.375

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
boot.control <- trainControl(method = "boot", number = 500)
```

```
model_3_boot <- train(price ~ enginesize, data = df, method = "lm",
trControl = boot.control)
print(model_3_boot)
> print(model_3_boot)
Linear Regression

193 samples
  1 predictor

No pre-processing
Resampling: Bootstrapped (500 reps)
Summary of sample sizes: 193, 193, 193, 193, 193, 193, ...
Resampling results:

  RMSE     Rsquared   MAE
  3786.08  0.7873644  2671.926

Tuning parameter 'intercept' was held constant at a value of TRUE
```