

DATA603Project: Aircraft Price Prediction

Anitha Joseph

Jincy Thomas

Joshua Quartey

Megha Radhakrishnan Sanitha

Prichelle Lal

Riya Chevli

2025-03-30

Aircraft Price Prediction using Multiple Linear Regression

The aircraft price varies by their engine type, power, fuel capacity, speed, weight, distance range and other parameters. Prices for 518 different aircraft are given in aircraft_price.csv file. The dataset includes 517 aircraft records with 14 quantitative variables and 2 qualitative variables. Quantitative Variables are measured on a continuous ratio scale (e.g., speed in knots, weight in pounds, price in dollars) and Qualitative Variables are measured on a nominal scale (e.g., engine type as piston or other). The dataset contains a variety of aircraft performance and specification data, which could be useful for price prediction.

The explained variable is:

1. price: in US dollars (Quantitative variable)

The explanatory variables are:

1. model_name: Name of the aircraft model.
2. engine_type: Type of the engine. (Qualitative variable)
3. engine_power: Power of the engine. (hp)
4. max_speed: Maximum speed of the aircraft. (Knots(kt))
5. cruise_speed: Cruise speed of the aircraft. (Knots(kt))
6. stall_speed: Minimum speed of the aircraft to prevent stalling.(Knots(kt))
7. fuel_tank: Fuel tank capacity of the aircraft. (gallons(gal))
8. all_eng_roc: All Engine Rate of Climb. The maximum altitude of the aircraft at full power. (feet/min)
9. out_eng_roc: Out Engine Rate of Climb. The maximum altitude of the aircraft at out power. (feet/min)
10. takeoff_distance: The minimum distance required for an aircraft to take off. (feet)
11. landing_distance: The minimum distance required for an aircraft to land. (feet)
12. empty_weight: Empty weight of the aircraft. (pounds(lbs))
13. length: Length of the aircraft. (inch)

14. wing_span: Wingspan of the aircraft. (inch)

15. range: Range of the aircraft. (nmi)

```
# Read the aircraft data from the csv file
```

```
Aircraft = read.csv("/Users/anithajoseph/Documents/UofC/DATA603/project/aircraft_price.csv")  
print(head(Aircraft))
```

```
##           model_name engine_type engine_power max_speed  
## 1      100 Darter (S.L. Industries)      Piston      145      104  
## 2           7 CCM Champ      Piston       85       89  
## 3      100 Darter (S.L. Industries)      Piston       90       90  
## 4           7 AC Champ      Piston       85       88  
## 5      100 Darter (S.L. Industries)      Piston       65       83  
## 6 PA-60-700P Aerostar (preliminary)      Piston       65       78  
##   cruise_speed stall_speed fuel_tank all_eng_roc out_eng_roc takeoff_distance  
## 1           91         46        36        450        900          1300  
## 2           83         44        15        600        720           800  
## 3           78         37        19        650        475           850  
## 4           78         37        19        620        500           850  
## 5           74         33        14        370        632           885  
## 6           72         33        15        360        583           880  
##   landing_distance empty_weight length wing_span range   price  
## 1           2050         1180    303     449   370 1300000  
## 2           1350         820    247     433   190 1230000  
## 3           1300         810    257     420   210 1600000  
## 4           1300         800    257     420   210 1300000  
## 5           1220         740    257     420   175 1250000  
## 6           1250         786    244     433   180 1100000
```

1. Cleaning the Aircraft data

Check the Aircraft data for any missing values and then remove rows with missing data to create a cleaned version of the data set. It ensures that the cleaned Aircraft data has complete information in every row so you can analyze it without any gaps in data.

```
# Check for missing values  
sum(is.na(Aircraft))
```

```
## [1] 10
```

```
# Remove rows with any missing values  
AircraftData <- na.omit(Aircraft)  
print(head(AircraftData))
```

```
##           model_name engine_type engine_power max_speed
## 1      100 Darter (S.L. Industries)      Piston         145      104
## 2              7 CCM Champ      Piston          85       89
## 3      100 Darter (S.L. Industries)      Piston          90       90
## 4              7 AC Champ      Piston          85       88
## 5      100 Darter (S.L. Industries)      Piston          65       83
## 6 PA-60-700P Aerostar (preliminary)      Piston          65       78
##   cruise_speed stall_speed fuel_tank all_eng_roc out_eng_roc takeoff_distance
## 1           91          46         36         450         900           1300
## 2           83          44         15         600         720           800
## 3           78          37         19         650         475           850
## 4           78          37         19         620         500           850
## 5           74          33         14         370         632           885
## 6           72          33         15         360         583           880
##   landing_distance empty_weight length wing_span range   price
## 1           2050         1180    303      449   370 1300000
## 2           1350          820    247      433   190 1230000
## 3           1300          810    257      420   210 1600000
## 4           1300          800    257      420   210 1300000
## 5           1220          740    257      420   175 1250000
## 6           1250          786    244      433   180 1100000
```

2. Create first order model

First order model for predicting the Aircraft price is generated with the all the explanatory variables excluding the model name.

```
firstordermodel = lm(price ~factor(engine_type)+engine_power +
                      max_speed + cruise_speed +stall_speed +
                      fuel_tank + all_eng_roc + out_eng_roc +
                      takeoff_distance + landing_distance +
                      empty_weight + length + wing_span + range,
                      data = AircraftData)
summary(firstordermodel)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##      cruise_speed + stall_speed + fuel_tank + all_eng_roc + out_eng_roc +
##      takeoff_distance + landing_distance + empty_weight + length +
##      wing_span + range, data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -935282 -228459  -47954  191710 1777937
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.653e+04  2.464e+05  -0.067 0.946532
## factor(engine_type)Piston  -4.274e+05  1.255e+05  -3.406 0.000714 ***
## factor(engine_type)Propjet -3.642e+05  1.246e+05  -2.923 0.003629 **
## engine_power      8.070e+01  4.566e+01   1.767 0.077814 .
## max_speed        1.794e+03  3.013e+02   5.954 5e-09 ***
## cruise_speed     4.828e+03  4.782e+02  10.096 < 2e-16 ***
## stall_speed     4.041e+03  2.019e+03   2.001 0.045938 *
## fuel_tank        1.166e+01  1.986e+01   0.587 0.557400
## all_eng_roc      8.712e+00  1.766e+01   0.493 0.622036
## out_eng_roc     -6.528e+01  3.569e+01  -1.829 0.067983 .
## takeoff_distance  8.075e+01  5.007e+01   1.613 0.107475
## landing_distance -8.968e+01  1.923e+01  -4.664 4e-06 ***
## empty_weight     6.497e+01  3.208e+01   2.025 0.043403 *
## length          1.533e+03  5.108e+02   3.001 0.002827 **
## wing_span        1.412e+03  4.032e+02   3.503 0.000502 ***
## range           1.825e+02  5.045e+01   3.617 0.000329 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 348000 on 491 degrees of freedom
## Multiple R-squared:  0.8868, Adjusted R-squared:  0.8833
## F-statistic: 256.3 on 15 and 491 DF, p-value: < 2.2e-16
```

3. Checking the Regression Assumptions

3.1. Multicollinearity

Multicollinearity occurs when independent variables are highly correlated, which can distort the coefficients in a regression model. We are testing for Multicollinearity with the following techniques:

1. Pairs plot
2. Variance Inflation Factors (VIF). (VIF > 5 → Moderate multicollinearity. VIF > 10 → Severe multicollinearity.)

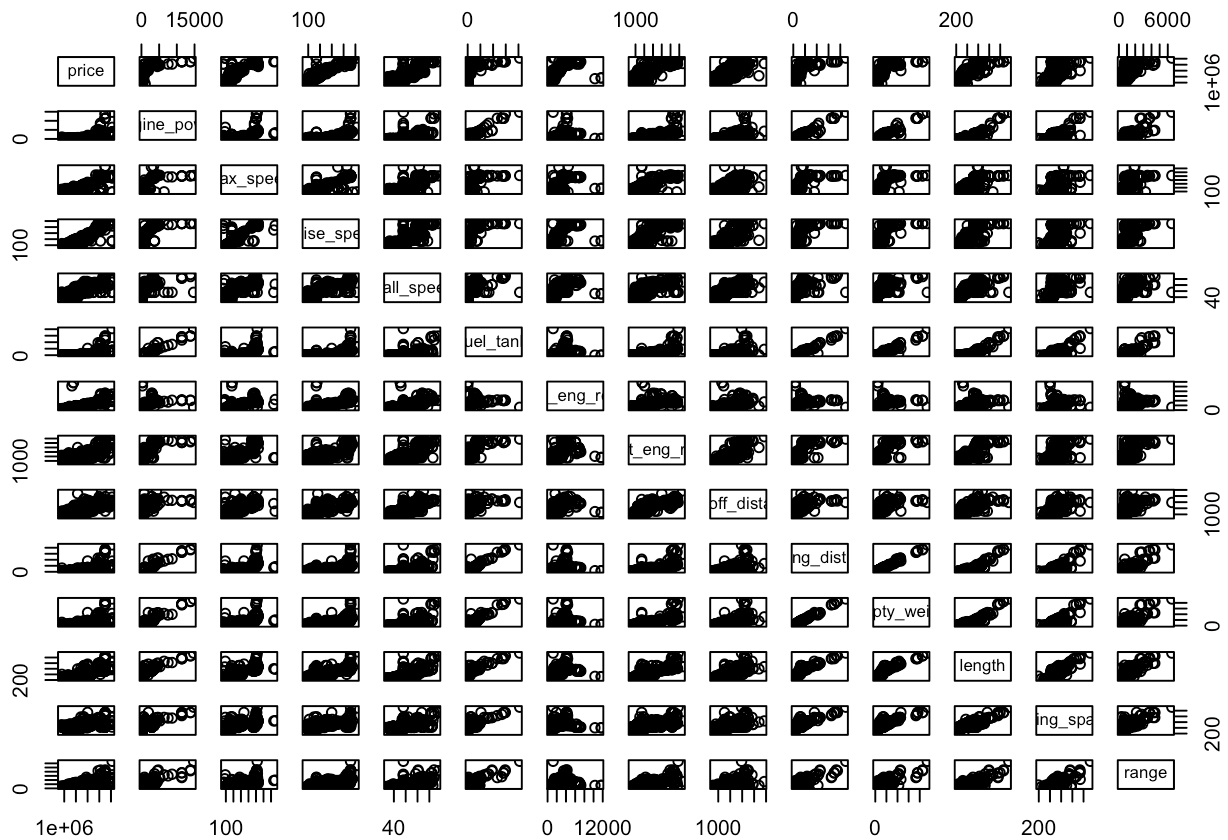
3.1.1 Multicollinearity: Pairs plot

Let's plot the most correlated predictors:

```

pairs(~ price + engine_power + max_speed + cruise_speed + stall_speed +
      fuel_tank + all_eng_roc + out_eng_roc + takeoff_distance + landing_distance +
      empty_weight + length + wing_span + range, data = AircraftData)

```



From the above output we can see that there exists correlation for engine_power with landing_distance, empty_weight and fuel_tank respectively. Also there exists a correlation between wing_span and length.

3.1.2 Multicollinearity: Variance Inflation Factors

So let's find the VIF for the variables.

```

#2. Variance Inflation Factors (VIF)
imcdiag(firstordermodel, method="VIF")

```

```
##
## Call:
## imcdiag(mod = firstordermodel, method = "VIF")
##
## VIF Multicollinearity Diagnostics
##
## VIF detection
## factor(engine_type)Piston 12.5070 1
## factor(engine_type)Propjet 6.5843 0
## engine_power 27.9138 1
## max_speed 4.9745 0
## cruise_speed 10.0485 1
## stall_speed 4.5704 0
## fuel_tank 30.5244 1
## all_eng_roc 2.7720 0
## out_eng_roc 6.7501 0
## takeoff_distance 5.2539 0
## landing_distance 165.2845 1
## empty_weight 138.6177 1
## length 21.4960 1
## wing_span 7.1323 0
## range 5.1477 0
##
## Multicollinearity may be due to factor(engine_type)Piston engine_power cruise_speed fuel_tank
landing_distance empty_weight length regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
vif(firstordermodel)
```

```
## GVIF Df GVIF^(1/(2*Df))
## factor(engine_type) 16.141293 2 2.004401
## engine_power 27.913786 1 5.283350
## max_speed 4.974526 1 2.230365
## cruise_speed 10.048520 1 3.169940
## stall_speed 4.570406 1 2.137851
## fuel_tank 30.524423 1 5.524891
## all_eng_roc 2.772011 1 1.664936
## out_eng_roc 6.750085 1 2.598093
## takeoff_distance 5.253891 1 2.292137
## landing_distance 165.284479 1 12.856301
## empty_weight 138.617698 1 11.773602
## length 21.496040 1 4.636382
## wing_span 7.132325 1 2.670641
## range 5.147730 1 2.268861
```

From the output we can see that landing_distance has highest VIF. According to the VIF values, we can list the highest multicollinearity items as below:

Severe Multicollinearity (VIF/GVIF > 10):

1. landing_distance (VIF=165.28, GVIF=165.28)
2. empty_weight (VIF=138.62, GVIF=138.62)
3. fuel_tank (VIF=30.52, GVIF=30.52)
4. engine_power (VIF=27.91, GVIF=27.91)
5. length (VIF=21.50, GVIF=21.50)
6. factor(engine_type)Piston (VIF=12.51, GVIF=16.14)
7. cruise_speed (VIF=10.05, GVIF=10.05)

Moderate/Low Collinearity (VIF/GVIF < 10): All other variables (e.g., max_speed, wing_span, range) are safe to retain. GVIF (Generalized Variance Inflation Factor) adjusts for categorical predictors like engine_type. For categorical variables with Df > 1, e.g., engine_type has 2 levels, $GVIF^{(1/(2 \cdot Df))}$ is interpreted. Here, factor(engine_type) has $GVIF^{(1/4)} = 2.00 \rightarrow$ Acceptable (under common thresholds).

So remove the most correlated variables Priority Order:

First: landing_distance (VIF=165.28)

Second: empty_weight (VIF=138.62)

Third: fuel_tank (VIF=30.52)

Fourth: length (VIF=21.50)

Since engine_power, Cruise_speed and engine_type are statistically and theoretically significant, we cannot remove them. Below code removes the variables in the highest collinearity order.

```
#firstordermodel
model1 = lm(price ~factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
              fuel_tank + all_eng_roc + out_eng_roc + takeoff_distance + landing_distance +
              empty_weight + length + wing_span + range, data = AircraftData)
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##      cruise_speed + stall_speed + fuel_tank + all_eng_roc + out_eng_roc +
##      takeoff_distance + landing_distance + empty_weight + length +
##      wing_span + range, data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -935282 -228459  -47954  191710 1777937
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.653e+04  2.464e+05  -0.067 0.946532
## factor(engine_type)Piston  -4.274e+05  1.255e+05  -3.406 0.000714 ***
## factor(engine_type)Propjet -3.642e+05  1.246e+05  -2.923 0.003629 **
## engine_power      8.070e+01  4.566e+01   1.767 0.077814 .
## max_speed        1.794e+03  3.013e+02   5.954 5e-09 ***
## cruise_speed     4.828e+03  4.782e+02  10.096 < 2e-16 ***
## stall_speed     4.041e+03  2.019e+03   2.001 0.045938 *
## fuel_tank        1.166e+01  1.986e+01   0.587 0.557400
## all_eng_roc      8.712e+00  1.766e+01   0.493 0.622036
## out_eng_roc     -6.528e+01  3.569e+01  -1.829 0.067983 .
## takeoff_distance  8.075e+01  5.007e+01   1.613 0.107475
## landing_distance -8.968e+01  1.923e+01  -4.664 4e-06 ***
## empty_weight     6.497e+01  3.208e+01   2.025 0.043403 *
## length          1.533e+03  5.108e+02   3.001 0.002827 **
## wing_span        1.412e+03  4.032e+02   3.503 0.000502 ***
## range           1.825e+02  5.045e+01   3.617 0.000329 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 348000 on 491 degrees of freedom
## Multiple R-squared:  0.8868, Adjusted R-squared:  0.8833
## F-statistic: 256.3 on 15 and 491 DF, p-value: < 2.2e-16
```

```
imcdiag(model1, method="VIF")
```



```
##
## Call:
## imcdiag(mod = model1, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 12.5070 1
## factor(engine_type)Propjet 6.5843 0
## engine_power 27.9138 1
## max_speed 4.9745 0
## cruise_speed 10.0485 1
## stall_speed 4.5704 0
## fuel_tank 30.5244 1
## all_eng_roc 2.7720 0
## out_eng_roc 6.7501 0
## takeoff_distance 5.2539 0
## landing_distance 165.2845 1
## empty_weight 138.6177 1
## length 21.4960 1
## wing_span 7.1323 0
## range 5.1477 0
##
## Multicollinearity may be due to factor(engine_type)Piston engine_power cruise_speed fuel_tank
landing_distance empty_weight length regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
#removed Landing_distance
model2 = lm(price ~factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
            fuel_tank + all_eng_roc + out_eng_roc + takeoff_distance +
            empty_weight + length + wing_span + range, data = AircraftData)
summary(model2)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##      cruise_speed + stall_speed + fuel_tank + all_eng_roc + out_eng_roc +
##      takeoff_distance + empty_weight + length + wing_span + range,
##      data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -922389 -230292  -50931   200933 1993660
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      57568.57   251044.88    0.229 0.818719
## factor(engine_type)Piston  -448395.55  128035.27   -3.502 0.000504 ***
## factor(engine_type)Propjet -394540.95  127027.84   -3.106 0.002006 **
## engine_power           46.97      46.03    1.020 0.308014
## max_speed            1640.00     305.71    5.365 1.25e-07 ***
## cruise_speed          4772.82     488.07    9.779 < 2e-16 ***
## stall_speed           4845.70    2053.93    2.359 0.018703 *
## fuel_tank             -19.76     19.07   -1.036 0.300587
## all_eng_roc            8.86      18.03    0.491 0.623346
## out_eng_roc           -54.06     36.35   -1.487 0.137661
## takeoff_distance       57.17     50.86    1.124 0.261460
## empty_weight          -60.34     17.90   -3.370 0.000809 ***
## length              1420.64     520.91    2.727 0.006615 **
## wing_span            1302.44     410.90    3.170 0.001621 **
## range                185.46      51.50    3.601 0.000349 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 355300 on 492 degrees of freedom
## Multiple R-squared:  0.8817, Adjusted R-squared:  0.8784
## F-statistic: 262 on 14 and 492 DF, p-value: < 2.2e-16
```

```
imcdiag(model2, method="VIF")
```

```
##
## Call:
## imcdiag(mod = model2, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 12.4909 1
## factor(engine_type)Propjet 6.5664 0
## engine_power 27.2139 1
## max_speed 4.9151 0
## cruise_speed 10.0423 1
## stall_speed 4.5370 0
## fuel_tank 27.0119 1
## all_eng_roc 2.7720 0
## out_eng_roc 6.7194 0
## takeoff_distance 5.2004 0
## empty_weight 41.4192 1
## length 21.4482 1
## wing_span 7.1080 0
## range 5.1469 0
##
## Multicollinearity may be due to factor(engine_type)Piston engine_power cruise_speed fuel_tank
empty_weight length regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
#removed empty_weight
model3 = lm(price ~factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
              fuel_tank + all_eng_roc + out_eng_roc + takeoff_distance + length + wing_
span + range,
              data = AircraftData)
summary(model3)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##      cruise_speed + stall_speed + fuel_tank + all_eng_roc + out_eng_roc +
##      takeoff_distance + length + wing_span + range, data = AircraftData)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -1010722  -228791   -53279    207544   1884293
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      569816.12   201901.75    2.822  0.00496 **
## factor(engine_type)Piston  -613204.07  119567.06   -5.129 4.20e-07 ***
## factor(engine_type)Propjet -588560.93  114419.57   -5.144 3.89e-07 ***
## engine_power          -27.66      40.78   -0.678  0.49794
## max_speed            1494.70     305.82    4.888 1.38e-06 ***
## cruise_speed         4620.14     491.04    9.409 < 2e-16 ***
## stall_speed          2996.44    1999.99    1.498  0.13471
## fuel_tank            -50.73      16.89   -3.004  0.00280 **
## all_eng_roc           14.36      18.14    0.791  0.42916
## out_eng_roc          -37.31      36.39   -1.025  0.30571
## takeoff_distance      41.29      51.17    0.807  0.42006
## length              921.06     504.59    1.825  0.06855 .
## wing_span            894.18     396.75    2.254  0.02465 *
## range               207.97      51.60    4.031 6.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 359000 on 493 degrees of freedom
## Multiple R-squared:  0.879, Adjusted R-squared:  0.8758
## F-statistic: 275.5 on 13 and 493 DF, p-value: < 2.2e-16
```

```
imcdiag(model3, method="VIF")
```

```
##
## Call:
## imcdiag(mod = model3, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 10.6691 1
## factor(engine_type)Propjet 5.2179 0
## engine_power 20.9170 1
## max_speed 4.8173 0
## cruise_speed 9.9558 0
## stall_speed 4.2133 0
## fuel_tank 20.7414 1
## all_eng_roc 2.7493 0
## out_eng_roc 6.5939 0
## takeoff_distance 5.1557 0
## length 19.7116 1
## wing_span 6.4903 0
## range 5.0603 0
##
## Multicollinearity may be due to factor(engine_type)Piston engine_power fuel_tank length regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
#removed fuel_tank
model4 = lm(price ~factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
            all_eng_roc + out_eng_roc + takeoff_distance + length + wing_span + range,
            data = AircraftData)
summary(model4)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##      cruise_speed + stall_speed + all_eng_roc + out_eng_roc +
##      takeoff_distance + length + wing_span + range, data = AircraftData)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -1310103  -225685   -48704    196659   1939134
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      695908.97   199089.37     3.495 0.000516 ***
## factor(engine_type)Piston  -641222.25  120167.30    -5.336 1.45e-07 ***
## factor(engine_type)Propjet -560763.09  114967.61    -4.878 1.45e-06 ***
## engine_power          -126.51      24.28    -5.211 2.76e-07 ***
## max_speed            1467.85     308.16     4.763 2.51e-06 ***
## cruise_speed         4637.80     494.98     9.370 < 2e-16 ***
## stall_speed          3901.87    1993.15     1.958 0.050834 .
## all_eng_roc           18.96      18.22     1.040 0.298814
## out_eng_roc          -60.79      35.83    -1.697 0.090357 .
## takeoff_distance       73.95      50.40     1.467 0.142949
## length               954.72     508.55     1.877 0.061062 .
## wing_span            622.99     389.47     1.600 0.110330
## range                154.44      48.82     3.164 0.001654 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 361900 on 494 degrees of freedom
## Multiple R-squared:  0.8768, Adjusted R-squared:  0.8738
## F-statistic: 293 on 12 and 494 DF, p-value: < 2.2e-16
```

```
imcdiag(model4, method="VIF")
```

```
##
## Call:
## imcdiag(mod = model4, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 10.6042      1
## factor(engine_type)Propjet 5.1838      0
## engine_power              7.2957      0
## max_speed                 4.8132      0
## cruise_speed              9.9544      0
## stall_speed               4.1176      0
## all_eng_roc                2.7298      0
## out_eng_roc                6.2897      0
## takeoff_distance           4.9230      0
## length                    19.7019      1
## wing_span                  6.1544      0
## range                      4.4570      0
##
## Multicollinearity may be due to factor(engine_type)Piston length regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
#removed Length
model5 = lm(price ~factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
              all_eng_roc + out_eng_roc + takeoff_distance + wing_span + range, data =
AircraftData)
summary(model5)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##      cruise_speed + stall_speed + all_eng_roc + out_eng_roc +
##      takeoff_distance + wing_span + range, data = AircraftData)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -1291854  -224274   -51822   188629   2036117
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      723802.03   199039.77    3.636 0.000305 ***
## factor(engine_type)Piston  -670177.36   119476.86   -5.609 3.38e-08 ***
## factor(engine_type)Propjet -547492.38   115042.31   -4.759 2.56e-06 ***
## engine_power          -107.82      22.20   -4.858 1.60e-06 ***
## max_speed            1522.64     307.56    4.951 1.02e-06 ***
## cruise_speed         4702.09     495.05    9.498 < 2e-16 ***
## stall_speed         4695.48     1952.76    2.405 0.016559 *
## all_eng_roc           21.87       18.20    1.202 0.230105
## out_eng_roc          -44.91       34.90   -1.287 0.198788
## takeoff_distance       75.07       50.53    1.486 0.137965
## wing_span            1094.87     298.26    3.671 0.000268 ***
## range                166.44      48.52    3.430 0.000653 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 362800 on 495 degrees of freedom
## Multiple R-squared:  0.8759, Adjusted R-squared:  0.8732
## F-statistic: 317.7 on 11 and 495 DF, p-value: < 2.2e-16
```

```
imcdiag(model5, method="VIF")
```



```
##
## Call:
## imcdiag(mod = model5, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 10.4295      1
## factor(engine_type)Propjet 5.1642      0
## engine_power              6.0686      0
## max_speed                 4.7700      0
## cruise_speed              9.9067      0
## stall_speed               3.9324      0
## all_eng_roc                2.7099      0
## out_eng_roc                5.9389      0
## takeoff_distance           4.9223      0
## wing_span                  3.5910      0
## range                      4.3806      0
##
## Multicollinearity may be due to factor(engine_type)Piston regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
# removed insignifocant terms:all_eng_roc, out_eng_roc & takeoff_distance from model5
model6 = lm(price ~factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
             wing_span + range, data = AircraftData)
summary(model6)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##     cruise_speed + stall_speed + wing_span + range, data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1305416  -227626   -54056   193438  2121608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      693659.32   186978.43    3.710 0.000231 ***
## factor(engine_type)Piston  -661350.47   111533.27   -5.930 5.68e-09 ***
## factor(engine_type)Propjet -527735.55   110064.68   -4.795 2.15e-06 ***
## engine_power        -115.72      21.27   -5.441 8.31e-08 ***
## max_speed           1571.61     305.17    5.150 3.76e-07 ***
## cruise_speed        4963.33     474.50   10.460 < 2e-16 ***
## stall_speed        5401.47     1817.88    2.971 0.003109 **
## wing_span          1104.12     288.70    3.825 0.000148 ***
## range              159.60      47.52    3.359 0.000843 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 363200 on 498 degrees of freedom
## Multiple R-squared:  0.8749, Adjusted R-squared:  0.8729
## F-statistic: 435.3 on 8 and 498 DF,  p-value: < 2.2e-16
```

```
imcdiag(model6, method="VIF")
```

```
##
## Call:
## imcdiag(mod = model6, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 9.0677      0
## factor(engine_type)Propjet 4.7161      0
## engine_power              5.5579      0
## max_speed                 4.6856      0
## cruise_speed              9.0803      0
## stall_speed               3.4000      0
## wing_span                 3.3566      0
## range                     4.1923      0
##
## NOTE: VIF Method Failed to detect multicollinearity
##
##
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
# removed cruise_speed: even if the multicollinearity =0 , VIF>5, so removed from model5
model7 = lm(price ~factor(engine_type)+engine_power + max_speed + stall_speed + all_eng_roc + ou
t_eng_roc +
              takeoff_distance + wing_span + range, data = AircraftData)
summary(model7)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##     stall_speed + all_eng_roc + out_eng_roc + takeoff_distance +
##     wing_span + range, data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1352143  -252725   -53828    229130   1554800
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.390e+06  2.023e+05   6.868 1.96e-11 ***
## factor(engine_type)Piston -1.207e+06  1.144e+05 -10.551 < 2e-16 ***
## factor(engine_type)Propjet -1.019e+06  1.127e+05  -9.035 < 2e-16 ***
## engine_power      -1.130e+02  2.410e+01  -4.686 3.60e-06 ***
## max_speed         2.389e+03  3.191e+02   7.487 3.24e-13 ***
## stall_speed       1.040e+04  2.018e+03   5.151 3.74e-07 ***
## all_eng_roc       6.346e+01  1.919e+01   3.306 0.00101 **
## out_eng_roc      -6.729e+01  3.782e+01  -1.779 0.07587 .
## takeoff_distance   1.503e+02  5.420e+01   2.774 0.00575 **
## wing_span         9.717e+02  3.237e+02   3.002 0.00282 **
## range            2.791e+02  5.111e+01   5.460 7.53e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 394100 on 496 degrees of freedom
## Multiple R-squared:  0.8533, Adjusted R-squared:  0.8503
## F-statistic: 288.5 on 10 and 496 DF, p-value: < 2.2e-16
```

```
#imcdiag(model7, method="VIF")
```

```
# removed insignifocant terms:out_eng_roc from model6
```

```
model8 = lm(price ~factor(engine_type)+engine_power + max_speed + stall_speed +
             all_eng_roc + takeoff_distance + wing_span + range, data = AircraftData)
summary(model8)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##     stall_speed + all_eng_roc + takeoff_distance + wing_span +
##     range, data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1380009  -248912   -51757   236662  1520550
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.282e+06  1.934e+05   6.625 9.02e-11 ***
## factor(engine_type)Piston -1.144e+06  1.091e+05 -10.489 < 2e-16 ***
## factor(engine_type)Propjet -9.659e+05  1.090e+05  -8.860 < 2e-16 ***
## engine_power      -1.240e+02  2.333e+01  -5.316 1.61e-07 ***
## max_speed         2.383e+03  3.197e+02   7.453 4.08e-13 ***
## stall_speed       1.020e+04  2.020e+03   5.051 6.19e-07 ***
## all_eng_roc       6.615e+01  1.918e+01   3.449 0.00061 ***
## takeoff_distance  1.001e+02  4.636e+01   2.159 0.03134 *
## wing_span         1.036e+03  3.224e+02   3.213 0.00140 **
## range            2.709e+02  5.101e+01   5.310 1.65e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 395000 on 497 degrees of freedom
## Multiple R-squared:  0.8524, Adjusted R-squared:  0.8497
## F-statistic: 318.8 on 9 and 497 DF,  p-value: < 2.2e-16
```

```
imcdiag(model7, method="VIF")
```

```
##
## Call:
## imcdiag(mod = model7, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 8.0988 0
## factor(engine_type)Propjet 4.2041 0
## engine_power 6.0650 0
## max_speed 4.3507 0
## stall_speed 3.5607 0
## all_eng_roc 2.5532 0
## out_eng_roc 5.9119 0
## takeoff_distance 4.8012 0
## wing_span 3.5842 0
## range 4.1191 0
##
## NOTE: VIF Method Failed to detect multicollinearity
##
##
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

Proceeding with model 5, since the multicollinearity removed.

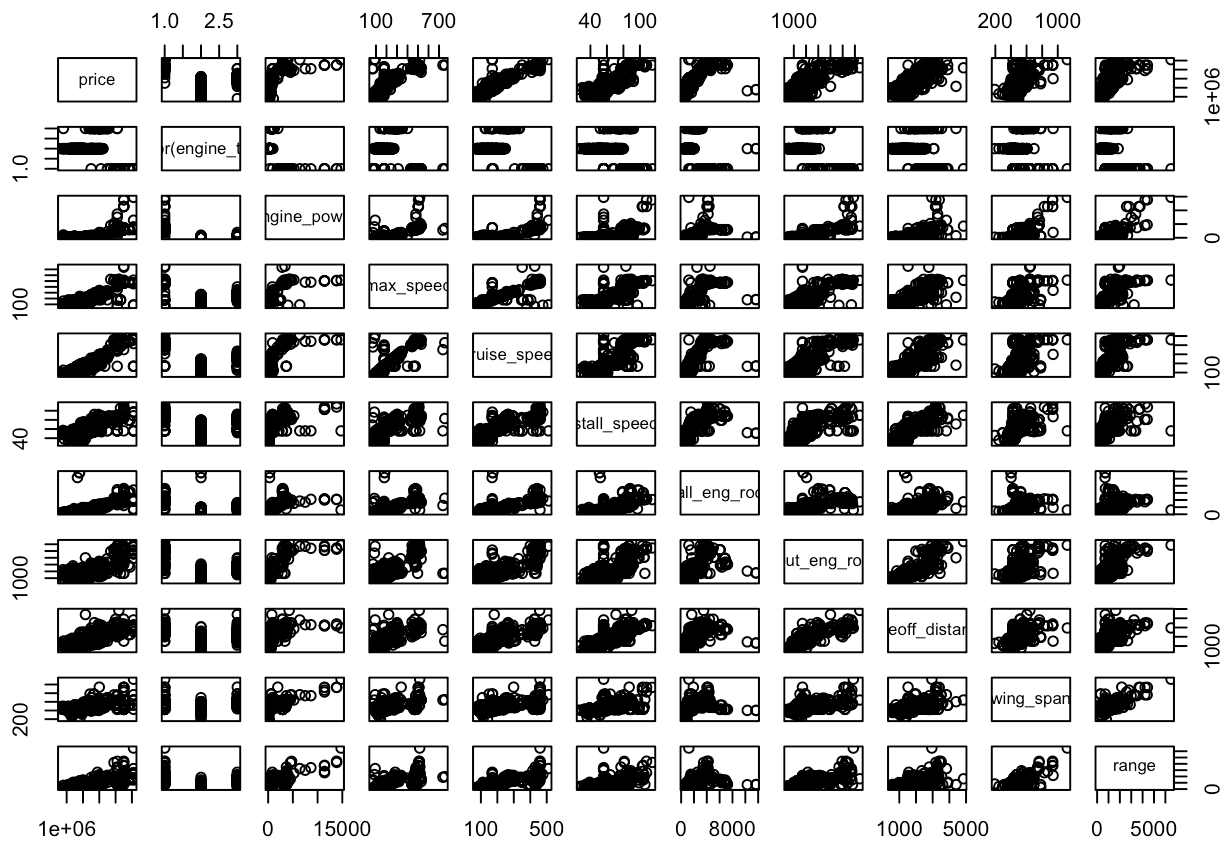
```
model5 = lm(price ~factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
              all_eng_roc + out_eng_roc + takeoff_distance + wing_span + range, data =
AircraftData)
summary(model5)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##      cruise_speed + stall_speed + all_eng_roc + out_eng_roc +
##      takeoff_distance + wing_span + range, data = AircraftData)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -1291854  -224274   -51822   188629   2036117
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      723802.03   199039.77    3.636 0.000305 ***
## factor(engine_type)Piston  -670177.36   119476.86   -5.609 3.38e-08 ***
## factor(engine_type)Propjet -547492.38   115042.31   -4.759 2.56e-06 ***
## engine_power          -107.82      22.20   -4.858 1.60e-06 ***
## max_speed             1522.64     307.56    4.951 1.02e-06 ***
## cruise_speed           4702.09     495.05    9.498 < 2e-16 ***
## stall_speed           4695.48     1952.76    2.405 0.016559 *
## all_eng_roc            21.87       18.20    1.202 0.230105
## out_eng_roc           -44.91       34.90   -1.287 0.198788
## takeoff_distance        75.07       50.53    1.486 0.137965
## wing_span             1094.87      298.26    3.671 0.000268 ***
## range                 166.44       48.52    3.430 0.000653 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 362800 on 495 degrees of freedom
## Multiple R-squared:  0.8759, Adjusted R-squared:  0.8732
## F-statistic: 317.7 on 11 and 495 DF, p-value: < 2.2e-16
```

```
imcdiag(model5, method="VIF")
```

```
##
## Call:
## imcdiag(mod = model5, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##
## VIF detection
## factor(engine_type)Piston 10.4295      1
## factor(engine_type)Propjet 5.1642      0
## engine_power              6.0686      0
## max_speed                 4.7700      0
## cruise_speed              9.9067      0
## stall_speed               3.9324      0
## all_eng_roc                2.7099      0
## out_eng_roc                5.9389      0
## takeoff_distance           4.9223      0
## wing_span                  3.5910      0
## range                      4.3806      0
##
## Multicollinearity may be due to factor(engine_type)Piston regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
plot(~price +factor(engine_type)+engine_power + max_speed + cruise_speed +stall_speed +
      all_eng_roc + out_eng_roc + takeoff_distance + wing_span + range, data =
AircraftData)
```

4. MODEL SELECTION PROCEDURES

4.1. Stepwise Selection Procedure

```
fullmodel = lm(price ~ engine_type + engine_power + max_speed + cruise_speed + stall_speed +
               all_eng_roc + out_eng_roc + takeoff_distance + wing_span + range, data =
AircraftData)
```

```
stepmod=ols_step_both_p(fullmodel,p_enter = 0.05, p_remove = 0.1, details=TRUE)
```

```

## Stepwise Selection Method
## -----
##
## Candidate Terms:
##
## 1. engine_type
## 2. engine_power
## 3. max_speed
## 4. cruise_speed
## 5. stall_speed
## 6. all_eng_roc
## 7. out_eng_roc
## 8. takeoff_distance
## 9. wing_span
## 10. range
##
##
## Step    => 0
## Model   => price ~ 1
## R2      => 0
##
## Initiating stepwise selection...
##
## Step    => 1
## Selected => cruise_speed
## Model   => price ~ cruise_speed
## R2      => 0.84
##
## Step    => 2
## Selected => max_speed
## Model   => price ~ cruise_speed + max_speed
## R2      => 0.853
##
## Step    => 3
## Selected => wing_span
## Model   => price ~ cruise_speed + max_speed + wing_span
## R2      => 0.859
##
## Step    => 4
## Selected => engine_power
## Model   => price ~ cruise_speed + max_speed + wing_span + engine_power
## R2      => 0.863
##
## Step    => 5
## Selected => engine_type
## Model   => price ~ cruise_speed + max_speed + wing_span + engine_power + engine_type
## R2      => 0.87
##
## Step    => 6
## Selected => range
## Model   => price ~ cruise_speed + max_speed + wing_span + engine_power + engine_type + range
e

```

```
## R2      => 0.873
##
## Step    => 7
## Selected => stall_speed
## Model   => price ~ cruise_speed + max_speed + wing_span + engine_power + engine_type + range + stall_speed
## R2      => 0.875
##
##
## No more variables to be added or removed.
```

```
summary(stepmod$model)
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1305416  -227626   -54056   193438  2121608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    693659.32   186978.43    3.710 0.000231 ***
## cruise_speed     4963.33     474.50   10.460 < 2e-16 ***
## max_speed       1571.61     305.17    5.150 3.76e-07 ***
## wing_span       1104.12     288.70    3.825 0.000148 ***
## engine_power    -115.72      21.27   -5.441 8.31e-08 ***
## engine_typePiston -661350.47  111533.27   -5.930 5.68e-09 ***
## engine_typePropjet -527735.55  110064.68   -4.795 2.15e-06 ***
## range           159.60       47.52    3.359 0.000843 ***
## stall_speed      5401.47     1817.88    2.971 0.003109 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 363200 on 498 degrees of freedom
## Multiple R-squared:  0.8749, Adjusted R-squared:  0.8729
## F-statistic: 435.3 on 8 and 498 DF,  p-value: < 2.2e-16
```

```
stepmod$metrics$adj_r2
```

```
## [1] 0.8400379 0.8519476 0.8576611 0.8620161 0.8684412 0.8708709 0.8728655
```

Best additive model using the Stepwise selection procedure is as follows:

$$\widehat{\text{price}} = 693659.32 - 661350.47X_{\text{engine_typePiston}} - 527735.55X_{\text{engine_typePropjet}} - 115.72X_{\text{engine_power}} + 1571.61X_{\text{max_speed}} + 4963.33X_{\text{cruise_speed}} + 5401.47X_{\text{stall_speed}} + 1104.12X_{\text{wing_span}} + 159.60X_{\text{range}}$$

4.2. Forward Selection Procedure

```
ExecPriceFor=ols_step_forward_p(fullmodel,p_val = 0.05, details=TRUE)
```

```

## Forward Selection Method
## -----
##
## Candidate Terms:
##
## 1. engine_type
## 2. engine_power
## 3. max_speed
## 4. cruise_speed
## 5. stall_speed
## 6. all_eng_roc
## 7. out_eng_roc
## 8. takeoff_distance
## 9. wing_span
## 10. range
##
##
## Step    => 0
## Model   => price ~ 1
## R2      => 0
##
## Initiating stepwise selection...
##
##                               Selection Metrics Table
## -----
## Predictor          Pr(>|t|)    R-Squared    Adj. R-Squared    AIC
## -----
## cruise_speed        0.00000      0.840        0.840      14541.306
## engine_type          0.00000      0.734        0.733      14801.855
## max_speed            0.00000      0.728        0.727      14811.548
## stall_speed          0.00000      0.621        0.620      14979.453
## takeoff_distance     0.00000      0.611        0.610      14993.017
## out_eng_roc          0.00000      0.595        0.594      15013.720
## range                0.00000      0.535        0.534      15082.938
## all_eng_roc          0.00000      0.513        0.512      15106.338
## engine_power         0.00000      0.454        0.453      15164.527
## wing_span            0.00000      0.374        0.372      15234.377
## -----
##
## Step      => 1
## Selected   => cruise_speed
## Model      => price ~ cruise_speed
## R2         => 0.84
##
##                               Selection Metrics Table
## -----
## Predictor          Pr(>|t|)    R-Squared    Adj. R-Squared    AIC
## -----
## max_speed          0.00000      0.853        0.852      14503.075
## engine_type        0.00000      0.853        0.852      14503.419
## stall_speed        0.00000      0.849        0.848      14515.385
## wing_span          0.00000      0.848        0.848      14516.785

```

```

## takeoff_distance      0.00000      0.848      0.847      14518.994
## range                 2e-05      0.846      0.845      14525.373
## out_eng_roc           0.00747      0.843      0.842      14536.101
## all_eng_roc           0.08229      0.841      0.841      14540.266
## engine_power          0.86979      0.840      0.840      14543.279
## -----
##
## Step      => 2
## Selected  => max_speed
## Model     => price ~ cruise_speed + max_speed
## R2        => 0.853
##
##                               Selection Metrics Table
## -----
## Predictor      Pr(>|t|)    R-Squared    Adj. R-Squared    AIC
## -----
## engine_type    0.00000      0.862      0.861      14474.722
## wing_span      1e-05      0.859      0.858      14484.115
## stall_speed    2e-05      0.858      0.857      14486.964
## takeoff_distance 3e-05      0.857      0.857      14487.716
## range          0.00126      0.856      0.855      14494.583
## out_eng_roc    0.13339      0.853      0.852      14502.802
## engine_power   0.26912      0.853      0.852      14503.842
## all_eng_roc    0.36159      0.853      0.852      14504.235
## -----
##
## Step      => 3
## Selected  => wing_span
## Model     => price ~ cruise_speed + max_speed + wing_span
## R2        => 0.859
##
##                               Selection Metrics Table
## -----
## Predictor      Pr(>|t|)    R-Squared    Adj. R-Squared    AIC
## -----
## engine_power    5e-05      0.863      0.862      14469.351
## engine_type     5e-05      0.864      0.863      14468.015
## stall_speed     0.00097      0.862      0.860      14475.119
## takeoff_distance 0.00211      0.861      0.860      14476.560
## all_eng_roc     0.05289      0.860      0.858      14482.326
## range          0.41669      0.859      0.858      14485.448
## out_eng_roc     0.69134      0.859      0.857      14485.955
## -----
##
## Step      => 4
## Selected  => engine_power
## Model     => price ~ cruise_speed + max_speed + wing_span + engine_power
## R2        => 0.863
##
##                               Selection Metrics Table
## -----
## Predictor      Pr(>|t|)    R-Squared    Adj. R-Squared    AIC

```

```

## -----
## engine_type      0.00000      0.870      0.868      14447.152
## takeoff_distance 0.00970      0.865      0.864      14464.574
## range            0.01788      0.865      0.863      14465.670
## stall_speed      0.02123      0.865      0.863      14465.975
## all_eng_roc      0.04258      0.864      0.863      14467.186
## out_eng_roc      0.06035      0.864      0.863      14467.777
## -----
##
## Step            => 5
## Selected        => engine_type
## Model           => price ~ cruise_speed + max_speed + wing_span + engine_power + engine_type
## R2              => 0.87
##
##                               Selection Metrics Table
## -----
## Predictor          Pr(>|t|)    R-Squared    Adj. R-Squared    AIC
## -----
## range              0.00134      0.873      0.871      14438.686
## stall_speed         0.00498      0.872      0.870      14441.132
## takeoff_distance    0.02453      0.871      0.870      14444.009
## all_eng_roc         0.24900      0.870      0.869      14447.800
## out_eng_roc         0.56433      0.870      0.868      14448.814
## -----
##
## Step              => 6
## Selected          => range
## Model             => price ~ cruise_speed + max_speed + wing_span + engine_power + engine_type + rang
e
## R2                => 0.873
##
##                               Selection Metrics Table
## -----
## Predictor          Pr(>|t|)    R-Squared    Adj. R-Squared    AIC
## -----
## stall_speed         0.00311      0.875      0.873      14431.776
## takeoff_distance    0.05944      0.874      0.872      14437.065
## all_eng_roc         0.12034      0.873      0.871      14438.227
## out_eng_roc         0.93557      0.873      0.871      14440.679
## -----
##
## Step              => 7
## Selected          => stall_speed
## Model             => price ~ cruise_speed + max_speed + wing_span + engine_power + engine_type + rang
e + stall_speed
## R2                => 0.875
##
##                               Selection Metrics Table
## -----
## Predictor          Pr(>|t|)    R-Squared    Adj. R-Squared    AIC
## -----
## all_eng_roc         0.20827      0.875      0.873      14432.159

```

```
## takeoff_distance      0.35603      0.875      0.873      14432.906
## out_eng_roc           0.50131      0.875      0.873      14433.315
## -----
##
##
## No more variables to be added.
##
## Variables Selected:
##
## => cruise_speed
## => max_speed
## => wing_span
## => engine_power
## => engine_type
## => range
## => stall_speed
```

```
summary(ExecPriceFor$model)
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1305416  -227626   -54056   193438   2121608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    693659.32   186978.43    3.710 0.000231 ***
## cruise_speed     4963.33     474.50   10.460 < 2e-16 ***
## max_speed       1571.61     305.17    5.150 3.76e-07 ***
## wing_span       1104.12     288.70    3.825 0.000148 ***
## engine_power    -115.72      21.27   -5.441 8.31e-08 ***
## engine_typePiston -661350.47  111533.27   -5.930 5.68e-09 ***
## engine_typePropjet -527735.55  110064.68   -4.795 2.15e-06 ***
## range           159.60      47.52    3.359 0.000843 ***
## stall_speed      5401.47     1817.88    2.971 0.003109 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 363200 on 498 degrees of freedom
## Multiple R-squared:  0.8749, Adjusted R-squared:  0.8729
## F-statistic: 435.3 on 8 and 498 DF,  p-value: < 2.2e-16
```

```
ExecPriceFor$metrics$adj_r2
```

```
## [1] 0.8400379 0.8519476 0.8576611 0.8620161 0.8684412 0.8708709 0.8728655
```


Best additive model using the Forward selection procedure is as follows:

$$\widehat{\text{price}} = 693659.32 - 661350.47X_{\text{engine_typePiston}} - 527735.55X_{\text{engine_typePropjet}} - 115.72X_{\text{engine_power}} + 1571.61X_{\text{max_speed}} + 4963.33X_{\text{cruise_speed}} + 5401.47X_{\text{stall_speed}} + 1104.12X_{\text{wing_span}} + 159.60X_{\text{range}}$$

4.3. Backwards Elimination

```
ExecPriceBack=ols_step_backward_p(fullmodel, p_val = 0.05, details=TRUE)
```

```

## Backward Elimination Method
## -----
##
## Candidate Terms:
##
## 1. engine_type
## 2. engine_power
## 3. max_speed
## 4. cruise_speed
## 5. stall_speed
## 6. all_eng_roc
## 7. out_eng_roc
## 8. takeoff_distance
## 9. wing_span
## 10. range
##
##
## Step    => 0
## Model   => price ~ engine_type + engine_power + max_speed + cruise_speed + stall_speed + all_e
ng_roc + out_eng_roc + takeoff_distance + wing_span + range
## R2      => 0.876
##
## Initiating stepwise selection...
##
## Step    => 1
## Removed => all_eng_roc
## Model   => price ~ engine_type + engine_power + max_speed + cruise_speed + stall_speed + out
_eng_roc + takeoff_distance + wing_span + range
## R2      => 0.87555
##
## Step    => 2
## Removed => out_eng_roc
## Model   => price ~ engine_type + engine_power + max_speed + cruise_speed + stall_speed + tak
eoff_distance + wing_span + range
## R2      => 0.87509
##
## Step    => 3
## Removed => takeoff_distance
## Model   => price ~ engine_type + engine_power + max_speed + cruise_speed + stall_speed + win
g_span + range
## R2      => 0.87488
##
##
## No more variables to be removed.
##
## Variables Removed:
##
## => all_eng_roc
## => out_eng_roc
## => takeoff_distance

```

```
summary(ExecPriceBack$model)
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(c(include, cterms), collapse = " + ")),
##     data = l)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1305416  -227626   -54056   193438  2121608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    693659.32   186978.43     3.710 0.000231 ***
## engine_typePiston  -661350.47   111533.27    -5.930 5.68e-09 ***
## engine_typePropjet -527735.55   110064.68    -4.795 2.15e-06 ***
## engine_power      -115.72      21.27    -5.441 8.31e-08 ***
## max_speed         1571.61      305.17     5.150 3.76e-07 ***
## cruise_speed       4963.33      474.50    10.460 < 2e-16 ***
## stall_speed        5401.47     1817.88     2.971 0.003109 **
## wing_span          1104.12      288.70     3.825 0.000148 ***
## range              159.60       47.52     3.359 0.000843 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 363200 on 498 degrees of freedom
## Multiple R-squared:  0.8749, Adjusted R-squared:  0.8729
## F-statistic: 435.3 on 8 and 498 DF,  p-value: < 2.2e-16
```

Best additive model using the Backwards elimination procedure is as follows:

$\widehat{\text{price}} = 693659.32 - 661350.47X_{\text{engine_typePiston}} - 527735.55X_{\text{engine_typePropjet}} - 115.72X_{\text{engine_power}} + 1571.61X_{\text{max_speed}} + 4963.33X_{\text{cruise_speed}} + 5401.47X_{\text{stall_speed}} + 1104.12X_{\text{wing_span}} + 159.60X_{\text{range}}$

```
ExecPriceBack$metrics
```

```
##   step      variable    r2   adj_r2    aic    sbc    sbic
## 1     1    all_eng_roc 0.8755545 0.8730456 14433.02 14483.76 12992.64
## 2     2    out_eng_roc 0.8750900 0.8728280 14432.91 14479.42 12992.42
## 3     3 takeoff_distance 0.8748755 0.8728655 14431.78 14474.06 12991.22
##  mallows_cp    rmse
## 1    11.44377 359021.5
## 2    11.29699 359691.0
## 3    10.15265 359999.7
```

```
ExecPriceBack$metrics$adj_r2
```

```
## [1] 0.8730456 0.8728280 0.8728655
```

4.4. All-Possible-Regressions Selection Procedure

```
#option 1  
ExecSubsets=ols_step_best_subset(fullmodel, details=TRUE)  
ExecSubsets$metrics
```

```

##      mindex  n
## 1         1  1
## 2         2  2
## 3         3  3
## 4         4  4
## 5         5  5
## 6         6  6
## 7         7  7
## 8         8  8
## 9         9  9
## 10        10 10
##
predictors
## 1
cruise_speed
## 2
engine_type cruise_speed
## 3
type max_speed cruise_speed
## 4
engine_type max_speed
d cruise_speed stall_speed
## 5
engine_type engine_power max_sp
eed cruise_speed wing_span
## 6
engine_type engine_power max_speed cr
uise_speed wing_span range
## 7
engine_type engine_power max_speed cruise_speed s
tall_speed wing_span range
## 8
engine_type engine_power max_speed cruise_speed stall_speed a
ll_eng_roc wing_span range
## 9
engine_type engine_power max_speed cruise_speed stall_speed out_eng_roc takeof
f_distance wing_span range
## 10 engine_type engine_power max_speed cruise_speed stall_speed all_eng_roc out_eng_roc takeof
f_distance wing_span range
##      rsquare      adjr   predrsq      cp      aic      sbic      sbc
## 1  0.8403540 0.8400379 0.8392919 133.86724 14541.31 13101.60 14553.99
## 2  0.8530137 0.8521370 0.8454943  87.36483 14503.42 13061.77 14524.56
## 3  0.8616489 0.8605465 0.8517118  54.91667 14474.72 13033.25 14500.09
## 4  0.8656081 0.8642669 0.8547768  41.12261 14462.00 13020.65 14491.60
## 5  0.8700012 0.8684412 0.8593856  25.59755 14447.15 13006.08 14480.98
## 6  0.8726573 0.8708709 0.8608337  17.00173 14438.69 12997.85 14476.74
## 7  0.8748755 0.8728655 0.8623799  10.15265 14431.78 12991.22 14474.06
## 8  0.8752739 0.8730153 0.8616052  10.56334 14432.16 12991.70 14478.67
## 9  0.8755545 0.8730456 0.8612297  11.44377 14433.02 12992.64 14483.76
## 10 0.8759165 0.8731590 0.8605288  12.00000 14433.54 12993.27 14488.51
##
##      msep      fpe      apc      hsp
## 1  8.416750e+13 166665713962 0.1609105 329386585
## 2  7.764723e+13 154362233774 0.1487362 305077973
## 3  7.323115e+13 145868543974 0.1405515 288300231
## 4  7.127750e+13 142254925506 0.1370690 281169096
## 5  6.908544e+13 138149327661 0.1331126 273067092
## 6  6.780952e+13 135862223961 0.1309083 268561043
## 7  6.676211e+13 134023891075 0.1291365 264943708

```

```
## 8 6.668344e+13 134125896389 0.1292343 265163966
## 9 6.666754e+13 134353777160 0.1294533 265635198
## 10 6.660795e+13 134493319232 0.1295873 265933906
```

Using option1(olsrr package): The 7th model is the best model, with CP = 10.15265 and AIC = 14431.78(smallest AIC) and predictors are: engine_type, engine_power, max_speed, cruise_speed, stall_speed, wing_span & range.

```
#option 1 Continued
rsquare=c((ExecSubsets$metrics)$rsquare)
AdjustedR=c((ExecSubsets$metrics)$adjr)
cp=c((ExecSubsets$metrics)$cp)
AIC=c((ExecSubsets$metrics)$aic)
cbind(rsquare,AdjustedR,cp,AIC)
```

```
##          rsquare AdjustedR          cp          AIC
## [1,] 0.8403540 0.8400379 133.86724 14541.31
## [2,] 0.8530137 0.8521370 87.36483 14503.42
## [3,] 0.8616489 0.8605465 54.91667 14474.72
## [4,] 0.8656081 0.8642669 41.12261 14462.00
## [5,] 0.8700012 0.8684412 25.59755 14447.15
## [6,] 0.8726573 0.8708709 17.00173 14438.69
## [7,] 0.8748755 0.8728655 10.15265 14431.78
## [8,] 0.8752739 0.8730153 10.56334 14432.16
## [9,] 0.8755545 0.8730456 11.44377 14433.02
## [10,] 0.8759165 0.8731590 12.00000 14433.54
```

```
#option 2: using Leaps package
best.subsetFullModel=regsubsets(price ~ engine_type + engine_power + max_speed + cruise_speed +s
tall_speed +
                                all_eng_roc + out_eng_roc + takeoff_distance + wing_span + range, data= A
ircraftData, nv=10 )
summary(best.subsetFullModel)
```

```

## Subset selection object
## Call: regsubsets.formula(price ~ engine_type + engine_power + max_speed +
##      cruise_speed + stall_speed + all_eng_roc + out_eng_roc +
##      takeoff_distance + wing_span + range, data = AircraftData,
##      nv = 10)
## 11 Variables (and intercept)
##
##      Forced in Forced out
## engine_typePiston      FALSE      FALSE
## engine_typePropjet     FALSE      FALSE
## engine_power           FALSE      FALSE
## max_speed              FALSE      FALSE
## cruise_speed           FALSE      FALSE
## stall_speed            FALSE      FALSE
## all_eng_roc            FALSE      FALSE
## out_eng_roc            FALSE      FALSE
## takeoff_distance       FALSE      FALSE
## wing_span              FALSE      FALSE
## range                  FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##
##      engine_typePiston engine_typePropjet engine_power max_speed
## 1 ( 1 ) " " " " " " " "
## 2 ( 1 ) "*" " " " " " "
## 3 ( 1 ) "*" " " " " "*"
## 4 ( 1 ) "*" " " " " "*"
## 5 ( 1 ) "*" " " "*" "*"
## 6 ( 1 ) "*" "*" "*" "*"
## 7 ( 1 ) "*" "*" "*" "*"
## 8 ( 1 ) "*" "*" "*" "*"
## 9 ( 1 ) "*" "*" "*" "*"
## 10 ( 1 ) "*" "*" "*" "*"
##
##      cruise_speed stall_speed all_eng_roc out_eng_roc takeoff_distance
## 1 ( 1 ) "*" " " " " " "
## 2 ( 1 ) "*" " " " " " "
## 3 ( 1 ) "*" " " " " " "
## 4 ( 1 ) "*" "*" " " " "
## 5 ( 1 ) "*" " " " " " "
## 6 ( 1 ) "*" " " " " " "
## 7 ( 1 ) "*" " " " " " "
## 8 ( 1 ) "*" "*" " " " "
## 9 ( 1 ) "*" "*" "*" " "
## 10 ( 1 ) "*" "*" " " "*"
##
##      wing_span range
## 1 ( 1 ) " " " "
## 2 ( 1 ) " " " "
## 3 ( 1 ) " " " "
## 4 ( 1 ) " " " "
## 5 ( 1 ) "*" " "
## 6 ( 1 ) "*" " "
## 7 ( 1 ) "*" "*"
## 8 ( 1 ) "*" "*"

```

```
## 9 ( 1 ) "*"      "*"
## 10 ( 1 ) "*"     "*"

```

```
reg.summary=summary(best.subsetFullModel)
rsquare=c(reg.summary$rsq)
cp=c(reg.summary$cp)
AdjustedR=c(reg.summary$adjr2)
RSS=c(reg.summary$rss)
BIC=c(reg.summary$bic)
cbind(rsquare,cp,BIC,RSS,AdjustedR)

```

##		rsquare	cp	BIC	RSS	AdjustedR
##	[1,]	0.8403540	133.86724	-917.7849	8.383547e+13	0.8400379
##	[2,]	0.8525548	87.19535	-951.8640	7.742844e+13	0.8519697
##	[3,]	0.8614890	53.55476	-977.3263	7.273681e+13	0.8606629
##	[4,]	0.8646943	42.76774	-982.9685	7.105356e+13	0.8636162
##	[5,]	0.8667015	36.76087	-984.3171	6.999956e+13	0.8653711
##	[6,]	0.8700012	25.59755	-990.7969	6.826677e+13	0.8684412
##	[7,]	0.8726573	17.00173	-995.0346	6.687197e+13	0.8708709
##	[8,]	0.8748755	10.15265	-997.7155	6.570710e+13	0.8728655
##	[9,]	0.8752739	10.56334	-993.1039	6.549789e+13	0.8730153
##	[10,]	0.8755545	11.44377	-988.0174	6.535051e+13	0.8730456

Using the option2(with leaps package): The model with AdjustedR2 = 0.8728655, MallowsCP = 10.15265, BIC = -997.7155 is the best predictor and predictors are: engine_type, engine_power, max_speed, cruise_speed, stall_speed, wing_span & range.

Thus we can conclude that our best additive model as: price ~ engine_type + engine_power + max_speed + cruise_speed + stall_speed + wing_span + range

5. Interaction model building

In the next step, we will build interaction models based on best additive model.

```
interactPriceFull = lm(price ~ (factor(engine_type) + engine_power + max_speed +
                                cruise_speed + stall_speed + wing_span + range)^2, data = Air
craftData)

summary(interactPriceFull)

```



```
##
## Call:
## lm(formula = price ~ (factor(engine_type) + engine_power + max_speed +
##   cruise_speed + stall_speed + wing_span + range)^2, data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -973549 -180573  -26201  139343  853490
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.256e+06  1.302e+06   3.268 0.001164
## factor(engine_type)Piston      -3.258e+06  9.645e+05  -3.378 0.000791
## factor(engine_type)Propjet     -1.990e+06  8.984e+05  -2.215 0.027236
## engine_power                   1.847e+03  3.102e+02   5.952 5.17e-09
## max_speed                     -1.218e+04  5.504e+03  -2.213 0.027346
## cruise_speed                   7.187e+03  4.248e+03   1.692 0.091361
## stall_speed                   -2.160e+04  1.887e+04  -1.145 0.252856
## wing_span                     -8.295e+03  2.848e+03  -2.913 0.003752
## range                         2.116e+03  6.499e+02   3.256 0.001212
## factor(engine_type)Piston:engine_power -1.317e+03  3.456e+02  -3.813 0.000156
## factor(engine_type)Propjet:engine_power -2.442e+03  3.324e+02  -7.348 8.93e-13
## factor(engine_type)Piston:max_speed   4.888e+03  2.405e+03   2.032 0.042677
## factor(engine_type)Propjet:max_speed   5.496e+03  1.766e+03   3.111 0.001976
## factor(engine_type)Piston:cruise_speed 7.405e+02  2.432e+03   0.304 0.760915
## factor(engine_type)Propjet:cruise_speed -4.113e+03  1.839e+03  -2.236 0.025807
## factor(engine_type)Piston:stall_speed  4.935e+03  9.977e+03   0.495 0.621117
## factor(engine_type)Propjet:stall_speed  2.555e+03  9.092e+03   0.281 0.778792
## factor(engine_type)Piston:wing_span   8.378e+03  2.152e+03   3.893 0.000113
## factor(engine_type)Propjet:wing_span   8.132e+03  1.868e+03   4.353 1.65e-05
## factor(engine_type)Piston:range      -1.349e+03  3.530e+02  -3.822 0.000150
## factor(engine_type)Propjet:range      -6.499e+02  2.909e+02  -2.234 0.025943
## engine_power:max_speed              -2.107e+00  5.115e-01  -4.119 4.49e-05
## engine_power:cruise_speed           -3.480e+00  7.785e-01  -4.471 9.79e-06
## engine_power:stall_speed              9.480e-03  1.875e+00   0.005 0.995967
## engine_power:wing_span               1.013e+00  3.409e-01   2.971 0.003119
## engine_power:range                   -2.007e-02  3.182e-02  -0.631 0.528524
## max_speed:cruise_speed               1.565e+01  8.910e+00   1.756 0.079692
## max_speed:stall_speed                 2.958e+00  1.872e+01   0.158 0.874498
## max_speed:wing_span                  2.484e+01  9.414e+00   2.638 0.008613
## max_speed:range                      -1.977e+00  1.340e+00  -1.475 0.140817
## cruise_speed:stall_speed              7.221e+01  3.182e+01   2.269 0.023694
## cruise_speed:wing_span               -1.060e+01  1.153e+01  -0.920 0.358169
## cruise_speed:range                   -1.439e+00  1.469e+00  -0.979 0.327940
## stall_speed:wing_span                 -1.562e+01  2.883e+01  -0.542 0.588139
## stall_speed:range                     8.615e-01  5.092e+00   0.169 0.865721
## wing_span:range                      -6.797e-01  5.239e-01  -1.297 0.195189
##
## (Intercept)                **
## factor(engine_type)Piston   ***
## factor(engine_type)Propjet  *
## engine_power                ***
```

```

## max_speed *
## cruise_speed .
## stall_speed
## wing_span **
## range **
## factor(engine_type)Piston:engine_power ***
## factor(engine_type)Propjet:engine_power ***
## factor(engine_type)Piston:max_speed *
## factor(engine_type)Propjet:max_speed **
## factor(engine_type)Piston:cruise_speed
## factor(engine_type)Propjet:cruise_speed *
## factor(engine_type)Piston:stall_speed
## factor(engine_type)Propjet:stall_speed
## factor(engine_type)Piston:wing_span ***
## factor(engine_type)Propjet:wing_span ***
## factor(engine_type)Piston:range ***
## factor(engine_type)Propjet:range *
## engine_power:max_speed ***
## engine_power:cruise_speed ***
## engine_power:stall_speed
## engine_power:wing_span **
## engine_power:range
## max_speed:cruise_speed .
## max_speed:stall_speed
## max_speed:wing_span **
## max_speed:range
## cruise_speed:stall_speed *
## cruise_speed:wing_span
## cruise_speed:range
## stall_speed:wing_span
## stall_speed:range
## wing_span:range
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 282100 on 471 degrees of freedom
## Multiple R-squared:  0.9286, Adjusted R-squared:  0.9233
## F-statistic: 175.1 on 35 and 471 DF,  p-value: < 2.2e-16

```

#insignificant terms are removed and new model is created

```

interacModel = lm(price ~ factor(engine_type) + engine_power + max_speed +
                    cruise_speed + stall_speed + wing_span + range +
                    factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                    factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                    engine_power * max_speed + engine_power * cruise_speed + engine_power *
                    wing_span +
                    max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

summary(interacModel)

```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##     cruise_speed + stall_speed + wing_span + range + factor(engine_type) *
##     engine_power + factor(engine_type) * max_speed + factor(engine_type) *
##     cruise_speed + factor(engine_type) * wing_span + engine_power *
##     max_speed + engine_power * cruise_speed + engine_power *
##     wing_span + max_speed * wing_span + cruise_speed * stall_speed,
##     data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -962876 -184373  -32016  139816  837474
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.205e+06  1.079e+06   3.898 0.000111
## factor(engine_type)Piston      -3.179e+06  7.852e+05  -4.048 6.01e-05
## factor(engine_type)Propjet     -1.889e+06  6.321e+05  -2.989 0.002945
## engine_power       1.726e+03  2.151e+02   8.022 7.86e-15
## max_speed        -2.126e+03  2.212e+03  -0.961 0.336890
## cruise_speed      3.563e+03  1.572e+03   2.267 0.023841
## stall_speed      -2.058e+04  3.907e+03  -5.266 2.10e-07
## wing_span        -6.681e+03  2.198e+03  -3.040 0.002493
## range            7.327e+01  4.460e+01   1.643 0.101044
## factor(engine_type)Piston:engine_power  -9.805e+02  2.683e+02  -3.655 0.000286
## factor(engine_type)Propjet:engine_power -2.217e+03  2.761e+02  -8.029 7.48e-15
## factor(engine_type)Piston:max_speed     1.097e+03  1.634e+03   0.671 0.502226
## factor(engine_type)Propjet:max_speed     4.679e+03  1.149e+03   4.074 5.40e-05
## factor(engine_type)Piston:cruise_speed   2.308e+03  1.984e+03   1.163 0.245364
## factor(engine_type)Propjet:cruise_speed -3.238e+03  1.443e+03  -2.244 0.025284
## factor(engine_type)Piston:wing_span      6.354e+03  1.488e+03   4.272 2.34e-05
## factor(engine_type)Propjet:wing_span     6.060e+03  1.267e+03   4.784 2.28e-06
## engine_power:max_speed     -1.553e+00  2.889e-01  -5.376 1.19e-07
## engine_power:cruise_speed  -2.895e+00  3.857e-01  -7.506 2.95e-13
## engine_power:wing_span      4.083e-01  1.060e-01   3.851 0.000133
## max_speed:wing_span        1.048e+01  4.561e+00   2.298 0.022004
## cruise_speed:stall_speed    6.309e+01  1.275e+01   4.949 1.03e-06
##
## (Intercept) ***
## factor(engine_type)Piston ***
## factor(engine_type)Propjet **
## engine_power ***
## max_speed
## cruise_speed *
## stall_speed ***
## wing_span **
## range
## factor(engine_type)Piston:engine_power ***
## factor(engine_type)Propjet:engine_power ***
## factor(engine_type)Piston:max_speed
## factor(engine_type)Propjet:max_speed ***
```

```
## factor(engine_type)Piston:cruise_speed
## factor(engine_type)Propjet:cruise_speed *
## factor(engine_type)Piston:wing_span      ***
## factor(engine_type)Propjet:wing_span      ***
## engine_power:max_speed                    ***
## engine_power:cruise_speed                 ***
## engine_power:wing_span                    ***
## max_speed:wing_span                       *
## cruise_speed:stall_speed                  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 284300 on 485 degrees of freedom
## Multiple R-squared:  0.9253, Adjusted R-squared:  0.9221
## F-statistic: 286.2 on 21 and 485 DF,  p-value: < 2.2e-16
```

#insignificant terms removed from the above interaction model: range

```
bestInteracModel = lm(price ~ factor(engine_type) + engine_power + max_speed +
                        cruise_speed + stall_speed + wing_span +
                        factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                        factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                        engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                        max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

summary(bestInteracModel)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + max_speed +
##     cruise_speed + stall_speed + wing_span + factor(engine_type) *
##     engine_power + factor(engine_type) * max_speed + factor(engine_type) *
##     cruise_speed + factor(engine_type) * wing_span + engine_power *
##     max_speed + engine_power * cruise_speed + engine_power *
##     wing_span + max_speed * wing_span + cruise_speed * stall_speed,
##     data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -962455 -183993  -29856  147763  842716
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.096e+06  1.079e+06   3.798 0.000165
## factor(engine_type)Piston      -3.084e+06  7.845e+05  -3.931 9.68e-05
## factor(engine_type)Propjet     -1.852e+06  6.328e+05  -2.927 0.003585
## engine_power        1.733e+03  2.154e+02   8.045 6.66e-15
## max_speed         -2.131e+03  2.216e+03  -0.962 0.336611
## cruise_speed        3.856e+03  1.564e+03   2.465 0.014048
## stall_speed       -1.973e+04  3.880e+03  -5.086 5.24e-07
## wing_span         -6.532e+03  2.200e+03  -2.970 0.003128
## factor(engine_type)Piston:engine_power  -1.041e+03  2.662e+02  -3.912 0.000105
## factor(engine_type)Propjet:engine_power -2.191e+03  2.762e+02  -7.935 1.47e-14
## factor(engine_type)Piston:max_speed    1.254e+03  1.634e+03   0.768 0.443107
## factor(engine_type)Propjet:max_speed    4.892e+03  1.143e+03   4.279 2.27e-05
## factor(engine_type)Piston:cruise_speed  2.194e+03  1.987e+03   1.105 0.269874
## factor(engine_type)Propjet:cruise_speed -3.253e+03  1.445e+03  -2.251 0.024860
## factor(engine_type)Piston:wing_span    6.155e+03  1.485e+03   4.144 4.02e-05
## factor(engine_type)Propjet:wing_span    5.861e+03  1.263e+03   4.640 4.49e-06
## engine_power:max_speed      -1.591e+00  2.885e-01  -5.513 5.74e-08
## engine_power:cruise_speed    -2.918e+00  3.861e-01  -7.559 2.04e-13
## engine_power:wing_span       4.369e-01  1.048e-01   4.171 3.60e-05
## max_speed:wing_span         1.071e+01  4.567e+00   2.346 0.019390
## cruise_speed:stall_speed      6.071e+01  1.269e+01   4.785 2.27e-06
##
## (Intercept)          ***
## factor(engine_type)Piston          ***
## factor(engine_type)Propjet          **
## engine_power          ***
## max_speed
## cruise_speed          *
## stall_speed          ***
## wing_span             **
## factor(engine_type)Piston:engine_power ***
## factor(engine_type)Propjet:engine_power ***
## factor(engine_type)Piston:max_speed
## factor(engine_type)Propjet:max_speed ***
## factor(engine_type)Piston:cruise_speed
## factor(engine_type)Propjet:cruise_speed *
```

```
## factor(engine_type)Piston:wing_span      ***
## factor(engine_type)Propjet:wing_span      ***
## engine_power:max_speed                    ***
## engine_power:cruise_speed                 ***
## engine_power:wing_span                    ***
## max_speed:wing_span                       *
## cruise_speed:stall_speed                   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 284800 on 486 degrees of freedom
## Multiple R-squared:  0.9249, Adjusted R-squared:  0.9218
## F-statistic: 299.3 on 20 and 486 DF,  p-value: < 2.2e-16
```

```
bestInteracModel => price ~ factor(engine_type) + engine_power + max_speed + cruise_speed + stall_speed +
wing_span + factor(engine_type) * engine_power + factor(engine_type) * max_speed + factor(engine_type) *
cruise_speed + factor(engine_type) * wing_span +
engine_power * max_speed + engine_power * cruise_speed + engine_power * wing_span + max_speed *
wing_span + cruise_speed * stall_speed
```

We have to test which model is to be taken as the best interaction model. So we can perform an ANOVA test. H_0 : the reduced model is the best H_1 : the full interaction model is the best

```
anova(bestInteracModel, interactPriceFull)
```

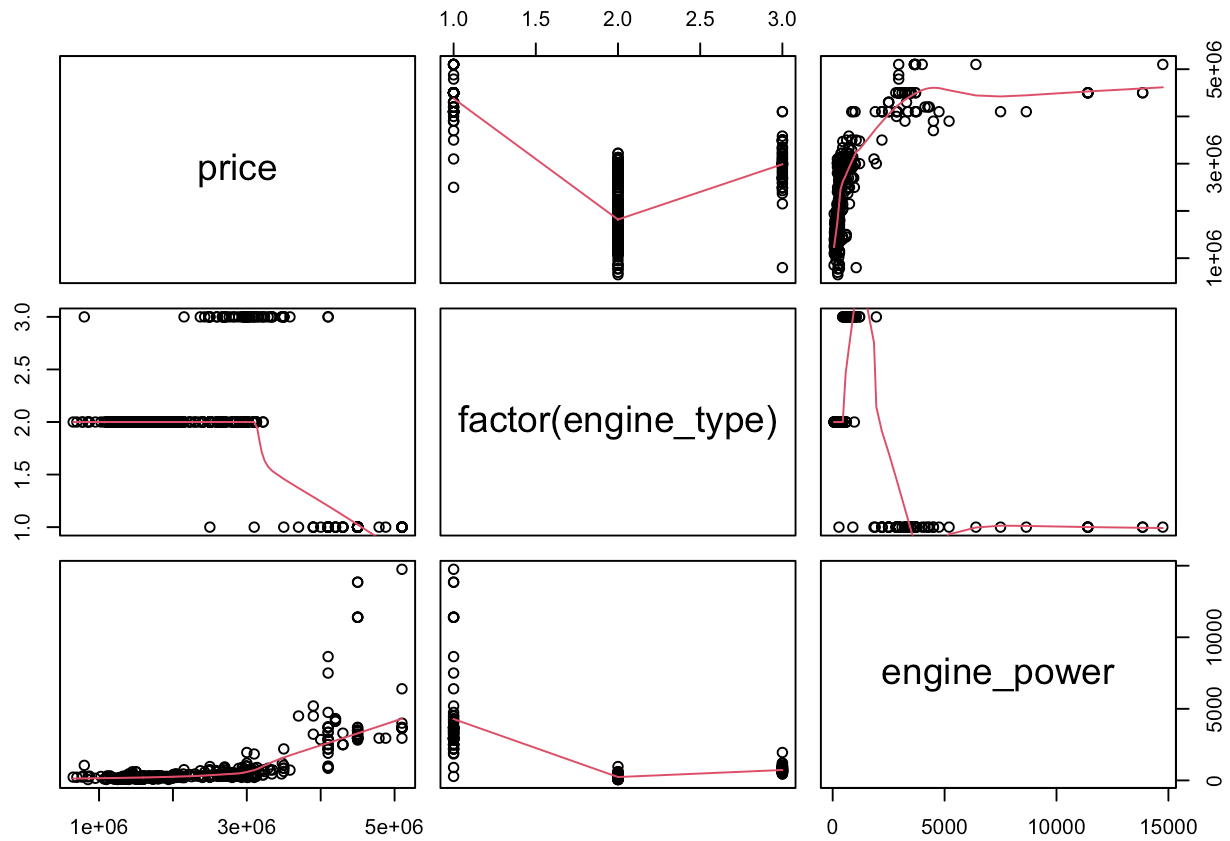
```
## Analysis of Variance Table
##
## Model 1: price ~ factor(engine_type) + engine_power + max_speed + cruise_speed +
##   stall_speed + wing_span + factor(engine_type) * engine_power +
##   factor(engine_type) * max_speed + factor(engine_type) * cruise_speed +
##   factor(engine_type) * wing_span + engine_power * max_speed +
##   engine_power * cruise_speed + engine_power * wing_span +
##   max_speed * wing_span + cruise_speed * stall_speed
## Model 2: price ~ (factor(engine_type) + engine_power + max_speed + cruise_speed +
##   stall_speed + wing_span + range)^2
##   Res.Df      RSS Df Sum of Sq    F Pr(>F)
## 1      486 3.9428e+13
## 2      471 3.7475e+13 15 1.9531e+12 1.6365 0.06086 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the above output, we can see that the p-value = 0.06086 > alpha(0.05). So we fail to reject the null hypothesis. There for we can say that the reduced model is the best.

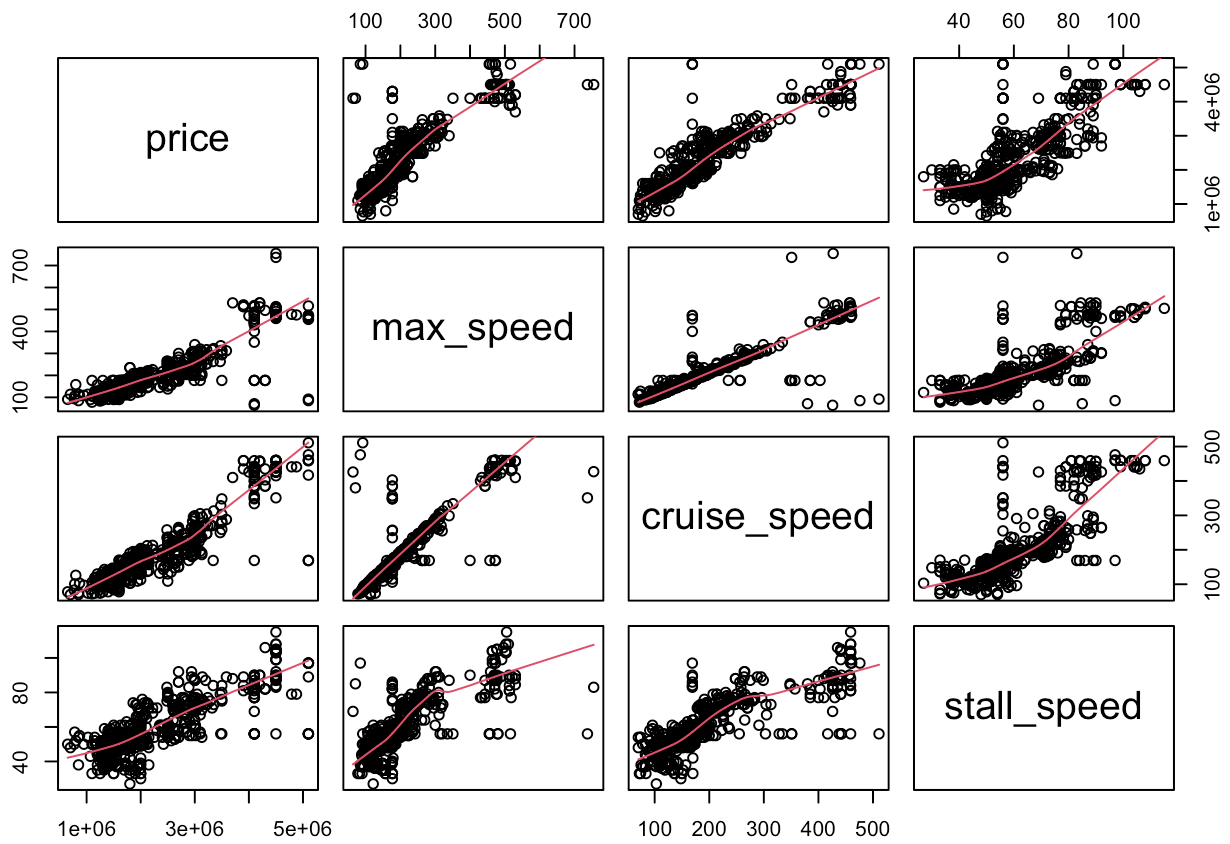
6. Introduce higher order terms in best interaction model

We can check if there is a possibility for higher order for the predictors using the pairs plot.

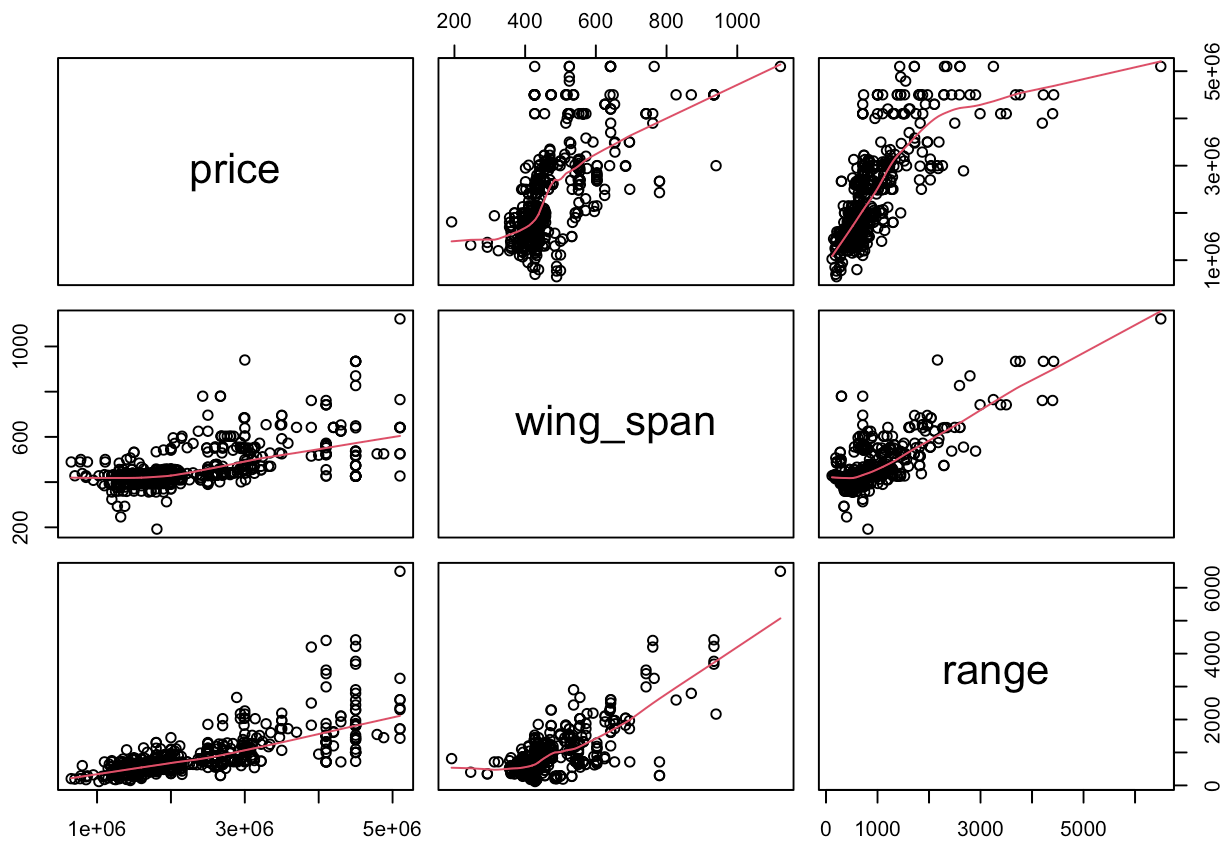
```
pairs(~price + factor(engine_type) +engine_power, data = AircraftData , panel = panel.smooth)
```



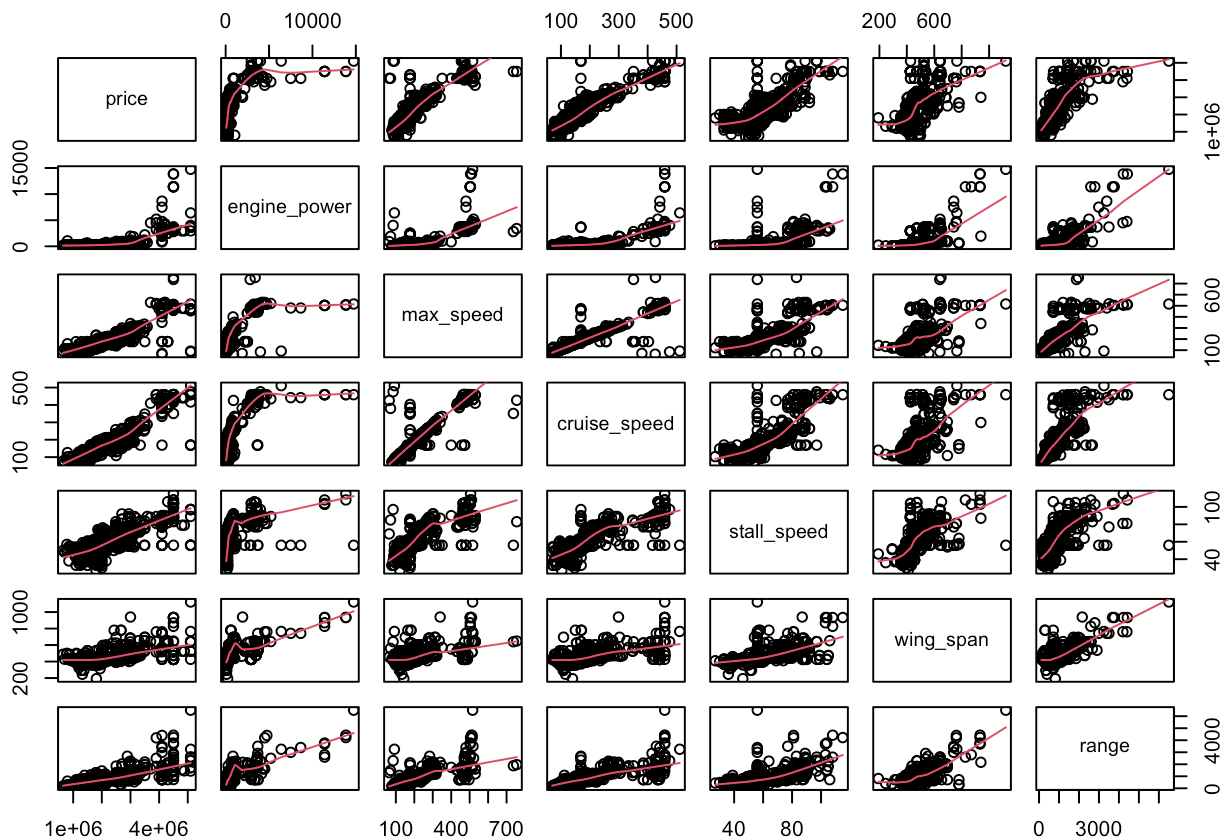
```
pairs(~price + max_speed + cruise_speed + stall_speed, data = AircraftData, panel = panel.smooth)
```



```
pairs(~price + wing_span + range ,data = AircraftData, panel = panel.smooth)
```

```
pairs(~price + engine_power + max_speed + cruise_speed + stall_speed + wing_span + range ,data = AircraftData, panel = panel.smooth)
```



From the above output, we can see there might be a possibility for higher orders for the variables: engine_power, max_speed and cruise_speed.

6.1. Checking higher order term for engine_power

```
# Checking I(engine_power^2)
higherOrder1 = lm(price ~ factor(engine_type) + engine_power + I(engine_power^2) +
                  max_speed + cruise_speed + stall_speed + wing_span +
                  factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                  factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                  engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                  max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)
summary(higherOrder1)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + I(engine_power^2) +
##     max_speed + cruise_speed + stall_speed + wing_span + factor(engine_type) *
##     engine_power + factor(engine_type) * max_speed + factor(engine_type) *
##     cruise_speed + factor(engine_type) * wing_span + engine_power *
##     max_speed + engine_power * cruise_speed + engine_power *
##     wing_span + max_speed * wing_span + cruise_speed * stall_speed,
##     data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -963496 -186337  -30832  146997  840584
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   3.923e+06  1.140e+06   3.442 0.000626
## factor(engine_type)Piston      -2.905e+06  8.709e+05  -3.336 0.000916
## factor(engine_type)Propjet     -1.707e+06  7.031e+05  -2.428 0.015544
## engine_power                   1.774e+03  2.320e+02   7.644 1.13e-13
## I(engine_power^2)              4.984e-03  1.050e-02   0.475 0.635092
## max_speed                     -2.120e+03  2.218e+03  -0.956 0.339733
## cruise_speed                   4.120e+03  1.662e+03   2.480 0.013486
## stall_speed                   -1.964e+04  3.888e+03  -5.052 6.21e-07
## wing_span                     -6.349e+03  2.235e+03  -2.841 0.004691
## factor(engine_type)Piston:engine_power -1.032e+03  2.671e+02  -3.863 0.000127
## factor(engine_type)Propjet:engine_power -2.166e+03  2.817e+02  -7.688 8.37e-14
## factor(engine_type)Piston:max_speed   1.118e+03  1.661e+03   0.673 0.500984
## factor(engine_type)Propjet:max_speed   4.750e+03  1.183e+03   4.017 6.84e-05
## factor(engine_type)Piston:cruise_speed 1.971e+03  2.043e+03   0.965 0.335254
## factor(engine_type)Propjet:cruise_speed -3.426e+03  1.492e+03  -2.297 0.022064
## factor(engine_type)Piston:wing_span   5.939e+03  1.555e+03   3.821 0.000150
## factor(engine_type)Propjet:wing_span   5.673e+03  1.324e+03   4.283 2.22e-05
## engine_power:max_speed              -1.649e+00  3.140e-01  -5.252 2.25e-07
## engine_power:cruise_speed            -2.968e+00  4.001e-01  -7.418 5.36e-13
## engine_power:wing_span               3.600e-01  1.929e-01   1.866 0.062612
## max_speed:wing_span                 1.105e+01  4.624e+00   2.389 0.017288
## cruise_speed:stall_speed             6.003e+01  1.278e+01   4.698 3.42e-06
##
## (Intercept) ***
## factor(engine_type)Piston ***
## factor(engine_type)Propjet *
## engine_power ***
## I(engine_power^2)
## max_speed
## cruise_speed *
## stall_speed ***
## wing_span **
## factor(engine_type)Piston:engine_power ***
## factor(engine_type)Propjet:engine_power ***
## factor(engine_type)Piston:max_speed
## factor(engine_type)Propjet:max_speed ***
```

```

## factor(engine_type)Piston:cruise_speed
## factor(engine_type)Propjet:cruise_speed *
## factor(engine_type)Piston:wing_span      ***
## factor(engine_type)Propjet:wing_span      ***
## engine_power:max_speed                    ***
## engine_power:cruise_speed                 ***
## engine_power:wing_span                    .
## max_speed:wing_span                       *
## cruise_speed:stall_speed                  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 285100 on 485 degrees of freedom
## Multiple R-squared:  0.925, Adjusted R-squared:  0.9217
## F-statistic: 284.6 on 21 and 485 DF, p-value: < 2.2e-16

```

Since $l(\text{engine_power}^2)$ is not significant with the p-value, we are moving to next step.

6.2. Checking higher order term for max_speed

```
# Checking I(max_speed^2)
higherOrderMaxSpeed2 = lm(price ~ factor(engine_type) + engine_power + I(max_speed^2) +
                           max_speed + cruise_speed + stall_speed + wing_span +
                           factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                           factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                           engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                           max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

# Checking I(max_speed^3)
higherOrderMaxSpeed3 = lm(price ~ factor(engine_type) + engine_power + I(max_speed^2) + I(max_sp
eed^3) +
                           max_speed + cruise_speed + stall_speed + wing_span +
                           factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                           factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                           engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                           max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

# Checking I(max_speed^4)
higherOrderMaxSpeed4 = lm(price ~ factor(engine_type) + engine_power + I(max_speed^2) + I(max_sp
eed^3) + I(max_speed^4) +
                           max_speed + cruise_speed + stall_speed + wing_span +
                           factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                           factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                           engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                           max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

summary(higherOrderMaxSpeed2)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + I(max_speed^2) +
##     max_speed + cruise_speed + stall_speed + wing_span + factor(engine_type) *
##     engine_power + factor(engine_type) * max_speed + factor(engine_type) *
##     cruise_speed + factor(engine_type) * wing_span + engine_power *
##     max_speed + engine_power * cruise_speed + engine_power *
##     wing_span + max_speed * wing_span + cruise_speed * stall_speed,
##     data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -878011 -179243  -33321  135051  877066
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.234e+06  1.073e+06   3.945 9.16e-05
## factor(engine_type)Piston      -3.288e+06  7.836e+05  -4.196 3.23e-05
## factor(engine_type)Propjet     -2.126e+06  6.376e+05  -3.335 0.000918
## engine_power                   1.738e+03  2.141e+02   8.118 3.95e-15
## I(max_speed^2)                 3.679e+00  1.398e+00   2.632 0.008755
## max_speed                     -3.740e+03  2.286e+03  -1.636 0.102456
## cruise_speed                   3.790e+03  1.555e+03   2.438 0.015147
## stall_speed                   -1.935e+04  3.860e+03  -5.014 7.49e-07
## wing_span                     -6.065e+03  2.193e+03  -2.765 0.005905
## factor(engine_type)Piston:engine_power -1.053e+03  2.646e+02  -3.980 7.96e-05
## factor(engine_type)Propjet:engine_power -2.182e+03  2.745e+02  -7.949 1.33e-14
## factor(engine_type)Piston:max_speed   2.477e+03  1.690e+03   1.466 0.143211
## factor(engine_type)Propjet:max_speed   5.720e+03  1.179e+03   4.851 1.66e-06
## factor(engine_type)Piston:cruise_speed 2.373e+03  1.976e+03   1.201 0.230342
## factor(engine_type)Propjet:cruise_speed -2.945e+03  1.441e+03  -2.043 0.041580
## factor(engine_type)Piston:wing_span   6.040e+03  1.477e+03   4.090 5.05e-05
## factor(engine_type)Propjet:wing_span   5.948e+03  1.256e+03   4.736 2.86e-06
## engine_power:max_speed             -1.491e+00  2.893e-01  -5.153 3.73e-07
## engine_power:cruise_speed           -2.962e+00  3.841e-01  -7.713 7.05e-14
## engine_power:wing_span              4.321e-01  1.041e-01   4.149 3.94e-05
## max_speed:wing_span                 8.488e+00  4.617e+00   1.838 0.066636
## cruise_speed:stall_speed             5.985e+01  1.261e+01   4.744 2.76e-06
##
## (Intercept) ***
## factor(engine_type)Piston ***
## factor(engine_type)Propjet ***
## engine_power ***
## I(max_speed^2) **
## max_speed
## cruise_speed *
## stall_speed ***
## wing_span **
## factor(engine_type)Piston:engine_power ***
## factor(engine_type)Propjet:engine_power ***
## factor(engine_type)Piston:max_speed
## factor(engine_type)Propjet:max_speed ***
```

```
## factor(engine_type)Piston:cruise_speed
## factor(engine_type)Propjet:cruise_speed *
## factor(engine_type)Piston:wing_span      ***
## factor(engine_type)Propjet:wing_span      ***
## engine_power:max_speed                    ***
## engine_power:cruise_speed                 ***
## engine_power:wing_span                    ***
## max_speed:wing_span                       .
## cruise_speed:stall_speed                  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 283100 on 485 degrees of freedom
## Multiple R-squared:  0.926, Adjusted R-squared:  0.9228
## F-statistic: 288.9 on 21 and 485 DF, p-value: < 2.2e-16
```

```
summary(higherOrderMaxSpeed3)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + I(max_speed^2) +
##     I(max_speed^3) + max_speed + cruise_speed + stall_speed +
##     wing_span + factor(engine_type) * engine_power + factor(engine_type) *
##     max_speed + factor(engine_type) * cruise_speed + factor(engine_type) *
##     wing_span + engine_power * max_speed + engine_power * cruise_speed +
##     engine_power * wing_span + max_speed * wing_span + cruise_speed *
##     stall_speed, data = AircraftData)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-832249	-176376	-34849	133272	861647

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.030e+06	1.104e+06	4.555	6.65e-06
factor(engine_type)Piston	-3.717e+06	7.936e+05	-4.683	3.67e-06
factor(engine_type)Propjet	-2.303e+06	6.365e+05	-3.619	0.000327
engine_power	1.598e+03	2.186e+02	7.311	1.11e-12
I(max_speed^2)	2.490e+01	7.806e+00	3.190	0.001515
I(max_speed^3)	-1.708e-02	6.181e-03	-2.763	0.005951
max_speed	-1.065e+04	3.378e+03	-3.153	0.001717
cruise_speed	3.005e+03	1.570e+03	1.913	0.056313
stall_speed	-1.871e+04	3.840e+03	-4.872	1.50e-06
wing_span	-5.586e+03	2.185e+03	-2.556	0.010890
factor(engine_type)Piston:engine_power	-8.734e+02	2.708e+02	-3.226	0.001342
factor(engine_type)Propjet:engine_power	-2.152e+03	2.729e+02	-7.886	2.10e-14
factor(engine_type)Piston:max_speed	3.816e+03	1.747e+03	2.185	0.029386
factor(engine_type)Propjet:max_speed	5.694e+03	1.171e+03	4.862	1.58e-06
factor(engine_type)Piston:cruise_speed	3.559e+03	2.009e+03	1.772	0.077078
factor(engine_type)Propjet:cruise_speed	-1.834e+03	1.487e+03	-1.233	0.218060
factor(engine_type)Piston:wing_span	5.581e+03	1.476e+03	3.780	0.000176
factor(engine_type)Propjet:wing_span	5.698e+03	1.251e+03	4.556	6.62e-06
engine_power:max_speed	-1.453e+00	2.876e-01	-5.051	6.23e-07
engine_power:cruise_speed	-2.725e+00	3.911e-01	-6.968	1.06e-11
engine_power:wing_span	4.370e-01	1.034e-01	4.224	2.87e-05
max_speed:wing_span	7.619e+00	4.597e+00	1.657	0.098103
cruise_speed:stall_speed	5.579e+01	1.262e+01	4.422	1.21e-05

```
##
## (Intercept) ***
## factor(engine_type)Piston ***
## factor(engine_type)Propjet ***
## engine_power ***
## I(max_speed^2) **
## I(max_speed^3) **
## max_speed **
## cruise_speed .
## stall_speed ***
## wing_span *
## factor(engine_type)Piston:engine_power **
## factor(engine_type)Propjet:engine_power ***
```



```
## factor(engine_type)Piston:max_speed      *
## factor(engine_type)Propjet:max_speed     ***
## factor(engine_type)Piston:cruise_speed   .
## factor(engine_type)Propjet:cruise_speed
## factor(engine_type)Piston:wing_span      ***
## factor(engine_type)Propjet:wing_span      ***
## engine_power:max_speed                   ***
## engine_power:cruise_speed               ***
## engine_power:wing_span                   ***
## max_speed:wing_span                      .
## cruise_speed:stall_speed                 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 281200 on 484 degrees of freedom
## Multiple R-squared:  0.9271, Adjusted R-squared:  0.9238
## F-statistic: 279.9 on 22 and 484 DF, p-value: < 2.2e-16
```

```
summary(higherOrderMaxSpeed4)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + I(max_speed^2) +
##     I(max_speed^3) + I(max_speed^4) + max_speed + cruise_speed +
##     stall_speed + wing_span + factor(engine_type) * engine_power +
##     factor(engine_type) * max_speed + factor(engine_type) * cruise_speed +
##     factor(engine_type) * wing_span + engine_power * max_speed +
##     engine_power * cruise_speed + engine_power * wing_span +
##     max_speed * wing_span + cruise_speed * stall_speed, data = AircraftData)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -830583 -175564  -35477   134014   863405
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.982e+06  1.116e+06   4.465 1.00e-05
## factor(engine_type)Piston      -3.589e+06  8.919e+05  -4.024 6.64e-05
## factor(engine_type)Propjet     -2.196e+06  7.225e+05  -3.040 0.002497
## engine_power                   1.596e+03  2.189e+02   7.291 1.27e-12
## I(max_speed^2)                 3.368e+01  2.901e+01   1.161 0.246156
## I(max_speed^3)                -3.651e-02  6.216e-02  -0.587 0.557173
## I(max_speed^4)                 1.329e-05  4.230e-05   0.314 0.753434
## max_speed                     -1.190e+04  5.213e+03  -2.282 0.022915
## cruise_speed                   3.080e+03  1.590e+03   1.937 0.053332
## stall_speed                   -1.874e+04  3.845e+03  -4.874 1.49e-06
## wing_span                     -5.431e+03  2.243e+03  -2.421 0.015829
## factor(engine_type)Piston:engine_power -8.605e+02  2.741e+02  -3.139 0.001798
## factor(engine_type)Propjet:engine_power -2.143e+03  2.747e+02  -7.798 3.89e-14
## factor(engine_type)Piston:max_speed   3.599e+03  1.880e+03   1.915 0.056143
## factor(engine_type)Propjet:max_speed   5.532e+03  1.281e+03   4.319 1.90e-05
## factor(engine_type)Piston:cruise_speed 3.505e+03  2.018e+03   1.737 0.083014
## factor(engine_type)Propjet:cruise_speed -1.948e+03  1.532e+03  -1.272 0.204152
## factor(engine_type)Piston:wing_span   5.442e+03  1.543e+03   3.527 0.000461
## factor(engine_type)Propjet:wing_span   5.594e+03  1.295e+03   4.320 1.90e-05
## engine_power:max_speed              -1.437e+00  2.923e-01  -4.916 1.21e-06
## engine_power:cruise_speed            -2.733e+00  3.924e-01  -6.966 1.07e-11
## engine_power:wing_span               4.338e-01  1.040e-01   4.171 3.60e-05
## max_speed:wing_span                 7.428e+00  4.641e+00   1.600 0.110165
## cruise_speed:stall_speed             5.586e+01  1.263e+01   4.423 1.20e-05
##
## (Intercept)                    ***
## factor(engine_type)Piston      ***
## factor(engine_type)Propjet     **
## engine_power                   ***
## I(max_speed^2)
## I(max_speed^3)
## I(max_speed^4)
## max_speed                      *
## cruise_speed                   .
## stall_speed                    ***
## wing_span                      *
```

```
## factor(engine_type)Piston:engine_power **
## factor(engine_type)Propjet:engine_power ***
## factor(engine_type)Piston:max_speed .
## factor(engine_type)Propjet:max_speed ***
## factor(engine_type)Piston:cruise_speed .
## factor(engine_type)Propjet:cruise_speed
## factor(engine_type)Piston:wing_span ***
## factor(engine_type)Propjet:wing_span ***
## engine_power:max_speed ***
## engine_power:cruise_speed ***
## engine_power:wing_span ***
## max_speed:wing_span
## cruise_speed:stall_speed ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 281500 on 483 degrees of freedom
## Multiple R-squared:  0.9271, Adjusted R-squared:  0.9237
## F-statistic: 267.2 on 23 and 483 DF,  p-value: < 2.2e-16
```

The quartic term is not significant, So we proceed with the model with cubic term.

6.3. Checking higher order term for cruise_speed

```
# Checking I(cruise_speed^2)
higherOrderCruiseSpeed2 = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
                               max_speed + cruise_speed + stall_speed + wing_span +
                               factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                               factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                               engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                               max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

# Checking I(cruise_speed^3)
higherOrderCruiseSpeed3 = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) + I
(cruise_speed^3) +
                               max_speed + cruise_speed + stall_speed + wing_span +
                               factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                               factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                               engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                               max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

summary(higherOrderCruiseSpeed2)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
##     max_speed + cruise_speed + stall_speed + wing_span + factor(engine_type) *
##     engine_power + factor(engine_type) * max_speed + factor(engine_type) *
##     cruise_speed + factor(engine_type) * wing_span + engine_power *
##     max_speed + engine_power * cruise_speed + engine_power *
##     wing_span + max_speed * wing_span + cruise_speed * stall_speed,
##     data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -910848 -177986  -30919  135247  866456
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    5.823e+06  1.127e+06   5.168 3.47e-07
## factor(engine_type)Piston       -4.389e+06  8.234e+05  -5.330 1.51e-07
## factor(engine_type)Propjet      -2.755e+06  6.530e+05  -4.219 2.93e-05
## engine_power                    1.700e+03  2.115e+02   8.037 7.09e-15
## I(cruise_speed^2)               2.100e+01  4.707e+00   4.461 1.02e-05
## max_speed                      -3.170e+03  2.187e+03  -1.450 0.147823
## cruise_speed                   -7.506e+03  2.974e+03  -2.524 0.011918
## stall_speed                    -1.736e+04  3.844e+03  -4.516 7.92e-06
## wing_span                      -6.491e+03  2.158e+03  -3.008 0.002768
## factor(engine_type)Piston:engine_power -9.810e+02  2.615e+02  -3.751 0.000197
## factor(engine_type)Propjet:engine_power -2.285e+03  2.718e+02  -8.406 4.74e-16
## factor(engine_type)Piston:max_speed   1.571e+03  1.605e+03   0.979 0.328265
## factor(engine_type)Propjet:max_speed   5.697e+03  1.136e+03   5.015 7.46e-07
## factor(engine_type)Piston:cruise_speed  8.318e+03  2.384e+03   3.489 0.000529
## factor(engine_type)Propjet:cruise_speed -1.528e+02  1.579e+03  -0.097 0.922938
## factor(engine_type)Piston:wing_span    5.975e+03  1.458e+03   4.099 4.86e-05
## factor(engine_type)Propjet:wing_span    5.968e+03  1.239e+03   4.815 1.97e-06
## engine_power:max_speed              -1.270e+00  2.921e-01  -4.348 1.68e-05
## engine_power:cruise_speed            -3.213e+00  3.845e-01  -8.357 6.85e-16
## engine_power:wing_span               4.307e-01  1.028e-01   4.190 3.31e-05
## max_speed:wing_span                 1.064e+01  4.481e+00   2.374 0.017984
## cruise_speed:stall_speed             4.920e+01  1.271e+01   3.871 0.000123
##
## (Intercept) ***
## factor(engine_type)Piston ***
## factor(engine_type)Propjet ***
## engine_power ***
## I(cruise_speed^2) ***
## max_speed *
## stall_speed ***
## wing_span **
## factor(engine_type)Piston:engine_power ***
## factor(engine_type)Propjet:engine_power ***
## factor(engine_type)Piston:max_speed
## factor(engine_type)Propjet:max_speed ***
```

```
## factor(engine_type)Piston:cruise_speed ***
## factor(engine_type)Propjet:cruise_speed
## factor(engine_type)Piston:wing_span ***
## factor(engine_type)Propjet:wing_span ***
## engine_power:max_speed ***
## engine_power:cruise_speed ***
## engine_power:wing_span ***
## max_speed:wing_span *
## cruise_speed:stall_speed ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 279400 on 485 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.9248
## F-statistic: 297.1 on 21 and 485 DF, p-value: < 2.2e-16
```

```
summary(higherOrderCruiseSpeed3)
```

```
##
## Call:
## lm(formula = price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
##     I(cruise_speed^3) + max_speed + cruise_speed + stall_speed +
##     wing_span + factor(engine_type) * engine_power + factor(engine_type) *
##     max_speed + factor(engine_type) * cruise_speed + factor(engine_type) *
##     wing_span + engine_power * max_speed + engine_power * cruise_speed +
##     engine_power * wing_span + max_speed * wing_span + cruise_speed *
##     stall_speed, data = AircraftData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -908287 -178593  -32938  137168  864486
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.777e+06  1.145e+06   5.043 6.48e-07
## factor(engine_type)Piston -4.393e+06  8.244e+05  -5.329 1.52e-07
## factor(engine_type)Propjet -2.774e+06  6.589e+05  -4.211 3.04e-05
## engine_power       1.697e+03  2.121e+02   7.999 9.34e-15
## I(cruise_speed^2)    1.764e+01  1.522e+01   1.160 0.246775
## I(cruise_speed^3)    4.256e-03  1.837e-02   0.232 0.816899
## max_speed         -3.217e+03  2.198e+03  -1.463 0.143981
## cruise_speed       -6.697e+03  4.586e+03  -1.460 0.144831
## stall_speed       -1.734e+04  3.849e+03  -4.504 8.37e-06
## wing_span         -6.479e+03  2.161e+03  -2.999 0.002850
## factor(engine_type)Piston:engine_power -9.842e+02  2.621e+02  -3.755 0.000195
## factor(engine_type)Propjet:engine_power -2.279e+03  2.732e+02  -8.340 7.80e-16
## factor(engine_type)Piston:max_speed    1.672e+03  1.665e+03   1.004 0.315802
## factor(engine_type)Propjet:max_speed    5.729e+03  1.145e+03   5.002 7.96e-07
## factor(engine_type)Piston:cruise_speed  8.179e+03  2.460e+03   3.325 0.000951
## factor(engine_type)Propjet:cruise_speed -1.095e+02  1.592e+03  -0.069 0.945166
## factor(engine_type)Piston:wing_span    5.977e+03  1.459e+03   4.096 4.92e-05
## factor(engine_type)Propjet:wing_span    5.953e+03  1.242e+03   4.792 2.19e-06
## engine_power:max_speed -1.251e+00  3.028e-01  -4.132 4.23e-05
## engine_power:cruise_speed -3.228e+00  3.901e-01  -8.274 1.27e-15
## engine_power:wing_span   4.303e-01  1.029e-01   4.182 3.43e-05
## max_speed:wing_span     1.062e+01  4.485e+00   2.369 0.018240
## cruise_speed:stall_speed  4.917e+01  1.273e+01   3.864 0.000127
##
## (Intercept) ***
## factor(engine_type)Piston ***
## factor(engine_type)Propjet ***
## engine_power ***
## I(cruise_speed^2)
## I(cruise_speed^3)
## max_speed
## cruise_speed
## stall_speed ***
## wing_span **
## factor(engine_type)Piston:engine_power ***
## factor(engine_type)Propjet:engine_power ***
```

```
## factor(engine_type)Piston:max_speed
## factor(engine_type)Propjet:max_speed ***
## factor(engine_type)Piston:cruise_speed ***
## factor(engine_type)Propjet:cruise_speed
## factor(engine_type)Piston:wing_span ***
## factor(engine_type)Propjet:wing_span ***
## engine_power:max_speed ***
## engine_power:cruise_speed ***
## engine_power:wing_span ***
## max_speed:wing_span *
## cruise_speed:stall_speed ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 279700 on 484 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.9246
## F-statistic: 283.1 on 22 and 484 DF, p-value: < 2.2e-16
```

The cubic term for cruise_speed is not significant, So we proceed with the model with quadratic term.

6.4. Comparison of models with higher order terms to get the final best model

```
summary(higherOrderMaxSpeed3)$adj.r.squared
```

```
## [1] 0.9238114
```

```
sigma(higherOrderMaxSpeed3)
```

```
## [1] 281193.1
```

```
summary(higherOrderCruiseSpeed2)$adj.r.squared
```

```
## [1] 0.9247535
```

```
sigma(higherOrderCruiseSpeed2)
```

```
## [1] 279449.2
```

Based on the output above: Adjusted r square is higher for higherOrderCruiseSpeed2 model and RSE is lower, So we could say that the higherOrderMaxSpeed3 is the best model.

So from all the above models we can conclude our best model as:

```
price ~ factor(engine_type) + engine_power + I(cruise_speed^2) + max_speed + cruise_speed + stall_speed +
wing_span + factor(engine_type) * engine_power + factor(engine_type) * max_speed + factor(engine_type) *
cruise_speed + factor(engine_type) * wing_span + engine_power * max_speed + engine_power * cruise_speed +
```

engine_power * wing_span + max_speed * wing_span + cruise_speed * stall_speed

$$\widehat{\text{price}} = 5823241.56 - 4388887.92X_{\text{engine_typePiston}} - 2755059.51X_{\text{engine_typePropjet}} + 21X_{\text{cruise_speed}^2} + 1699.67X_{\text{engine_power}} - 3169.59X_{\text{max_speed}} - 7505.68X_{\text{cruise_speed}} - 17359.94X_{\text{stall_speed}} - 6490.75X_{\text{wing_span}} - 981X_{(\text{engine_type})\text{Piston}} * X_{\text{engine_power}} - 2284.54X_{(\text{engine_type})\text{Propjet}} * X_{\text{engine_power}} + 1570.64X_{(\text{engine_type})\text{Piston}} * X_{\text{max_speed}} + 5697.49X_{(\text{engine_type})\text{Propjet}} * X_{\text{max_speed}} + 8317.65X_{(\text{engine_type})\text{Piston}} * X_{\text{cruise_speed}} - 152.84X_{(\text{engine_type})\text{Propjet}} * X_{\text{cruise_speed}} + 5975.35X_{(\text{engine_type})\text{Piston}} * X_{\text{wing_span}} + 5967.59X_{(\text{engine_type})\text{Propjet}} * X_{\text{wing_span}} - 1.27X_{\text{engine_power}} * X_{\text{max_speed}} - 3.21X_{\text{engine_power}} * X_{\text{cruise_speed}} + 0.43X_{\text{engine_power}} * X_{\text{wing_span}} + 10.64X_{\text{max_speed}} * X_{\text{wing_span}} + 49.2X_{\text{cruise_speed}} * X_{\text{stall_speed}}$$

6.5. Interpretation of coefficients for best model

```
bestModel = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
               max_speed + cruise_speed + stall_speed + wing_span +
               factor(engine_type) * engine_power + factor(engine_type) * max_speed +
               factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
               engine_power * max_speed + engine_power * cruise_speed + engine_power *
               wing_span +
               max_speed * wing_span + cruise_speed * stall_speed, data = AircraftData)
a)
```

$\beta_0 = 5823241.56$

The price of an aircraft with jet engine type is \$5.8 million when all the other predictors are zero. (This is not valid in real life situations)

Main Effects of Qualitative variable:

$\beta(\text{enginetype})\text{Piston} = -4388887.92$ Aircraft with a Piston engine are, on average, \$4.4 million cheaper than the aircrafts with Jet engine type holding all other variables constant.

$\beta(\text{enginetype})\text{Propjet} = -2755059.51$ The price of an aircraft with a Propjet engine is \$2.8 million lesser than the jet engine type aircrafts. Still cheaper, but not as much as a piston plane.

Main effects of Quantitative variable:

$\beta \text{cruise_speed}^2 = 21$ The squared term suggests a nonlinear relationship between cruise speed and price. For a 1 unit increase in cruise_speed, the price increases by $21 * \text{cruisespeed}^2$, indicating that price rises more steeply at higher cruise speeds.

$\beta \text{enginepower} = 1699.67$

A 1-horsepower increase in engine_power price jump by \$1,699.67, reflecting the value of greater power. we know—stronger engines aren't cheap.

$\beta \text{maxspeed} = -3169.59$

For every 1 knot increase of max speed, the price drops by \$3,170. usually Faster planes should cost more but here this effect is tangled up with other factors we will get to—like interactions.

$\beta \text{cruisespeed} = -7505.68$

A 1-knot increase in cruise speed decreases price by \$7,505.68. #####checkkkkk

$\beta \text{stallspeed} = -17359.94$

If the stall speed goes up by a one knot, the price drops by \$17,360.

$\beta \text{ wingspan} = -6490.75$

A 1-inch increase in wingspan decreases price by \$6,491.

Main effect on Interaction Terms with Engine Type:

$\beta (\text{enginetype})\text{Piston} * \text{enginepower} = -981$

For Piston engines, each additional horsepower reduces the price increase from engine_power by \$981 compared to the reference type. Net effect for Piston: $1,699.67 - 981 = 718.67$ dollars per horsepower.

$\beta (\text{enginetype})\text{Propjet} * \text{enginepower} = -2,284.54$ For Propjet engines, each horsepower reduces the price increase by \$2,284.54.

$\beta (\text{enginetype})\text{Piston} * \text{maxspeed} = 1,570.64$

For Piston engines, a 1-knot increase in max speed increases price by an additional \$1,570.64 over the reference. Net effect: $-3,169.59 + 1,570.64 = -1,598.95$ dollars per unit. With piston engines, a knot increase of max speed cuts the price drop to \$1,599.

$\beta (\text{enginetype})\text{Propjet} * \text{maxspeed} = 5,697.49$

For Propjets, a 1-knot increase in max speed adds \$5,697.49 to the price of the aircraft. Net effect: $-3,169.59 + 5,697.49 = 2,527.90$, indicating higher max speed boosts price for Propjets.

$\beta (\text{enginetype})\text{Piston} * \text{cruisespeed} = 8,317.65$

For Piston engines, a 1-knot increase in cruise speed adds \$8,317.65. Net effect: $-7,505.68 + 8,317.65 = 811.97$ dollars per unit. Here, a knot increase of cruise speed adds \$812 to the price for aircrafts with piston engine.

$\beta (\text{enginetype})\text{Propjet} * \text{cruisespeed} = -152.84$

For Propjets, a 1-knot increase reduces the cruise speed effect by \$152.84. Net effect: $-7,505.68 - 152.84 = -7,658.52$. For propjets, cruise speed still drags the price down, by \$7,659 per knot.

$\beta (\text{enginetype})\text{Piston} * \text{wingspan} = 5,975.35$

For Piston engines, a 1-inch increase in wingspan adds \$5,975.35 to the aircraft price. Net effect: $-6,490.75 + 5,975.35 = -515.40$. Piston planes soften the wingspan price drop to about \$515 per inch.

$\beta (\text{enginetype})\text{Propjet} * \text{wingspan} = 5,967.59$

For Propjets, a 1-inch increase in wingspan increases the aircraft price by \$5,967.59. Net effect: $-6,490.75 + 5,967.59 = -523.16$. Propjet planes soften the wingspan price drop to about \$515 per inch.

Main effect of Quantitative Interaction Terms:

$\beta \text{ enginepower} * \text{maxspeed} = -1.27$

For a 1-knot increase in max speed, the effect of engine power on price decreases by \$1.27 per horsepower.

$\beta \text{ enginepower} * \text{cruisespeed} = -3.21$

For a 1-knot increase in cruise speed, the effect of engine power on aircraft price decreases by \$3.21 per horsepower.

$\beta \text{ enginepower} * \text{wingspan} = 0.43$

For a 1-inch increase in wingspan, the effect of engine power on aircraft price increases by \$0.43 per horsepower.

$\beta \text{ maxspeed} * \text{wingspan} = 10.64$

For a 1-inch increase in wingspan, the effect of max speed on aircraft price increases by \$10.64 per knot.

$\beta \text{ cruisespeed} * \text{stallspeed} = 49.2$

For a 1-knot increase in stall speed, the effect of cruise speed on price of the aircraft increases by \$49.2 per knot.

6.6. Confidence Interval Estimation for best mdoel

```
conf_intervals <- confint(bestModel, level = 0.95)
print(conf_intervals)
```

##	2.5 %	97.5 %
## (Intercept)	3.609132e+06	8.037351e+06
## factor(engine_type)Piston	-6.006723e+06	-2.771053e+06
## factor(engine_type)Propjet	-4.038185e+06	-1.471934e+06
## engine_power	1.284119e+03	2.115226e+03
## I(cruise_speed^2)	1.174739e+01	3.024338e+01
## max_speed	-7.465888e+03	1.126718e+03
## cruise_speed	-1.334846e+04	-1.662894e+03
## stall_speed	-2.491285e+04	-9.807029e+03
## wing_span	-1.073089e+04	-2.250609e+03
## factor(engine_type)Piston:engine_power	-1.494853e+03	-4.671547e+02
## factor(engine_type)Propjet:engine_power	-2.818517e+03	-1.750558e+03
## factor(engine_type)Piston:max_speed	-1.582947e+03	4.724235e+03
## factor(engine_type)Propjet:max_speed	3.465151e+03	7.929835e+03
## factor(engine_type)Piston:cruise_speed	3.633557e+03	1.300175e+04
## factor(engine_type)Propjet:cruise_speed	-3.255795e+03	2.950109e+03
## factor(engine_type)Piston:wing_span	3.111048e+03	8.839655e+03
## factor(engine_type)Propjet:wing_span	3.532246e+03	8.402934e+03
## engine_power:max_speed	-1.843571e+00	-6.958844e-01
## engine_power:cruise_speed	-3.968461e+00	-2.457532e+00
## engine_power:wing_span	2.287495e-01	6.326622e-01
## max_speed:wing_span	1.833294e+00	1.944134e+01
## cruise_speed:stall_speed	2.422499e+01	7.418080e+01

Our model identifies key drivers of aircraft price, with engine type, engine power, cruise speed, stall speed, and wing span playing significant roles. Engine power increases price, while stall speed and wing span tend to decrease it. Notably, Propjet aircraft benefit more from higher speeds and larger wingspans. The most significant predictors are cruise speed, engine power, and cruise speed squared. We are 95% confident that for every one-unit increase in cruise speed, price decreases by 1,660 to 13,300, while engine power increases price by 1,280 to 2,120 per unit. Additionally, the quadratic cruise speed term shows a nonlinear price effect, meaning price increases at higher cruise speeds but at a diminishing rate.

6.7. Prediction with Best model

```
bestModel = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
               max_speed + cruise_speed + stall_speed + wing_span +
               factor(engine_type) * engine_power + factor(engine_type) * max_speed +
               factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
               engine_power * max_speed + engine_power * cruise_speed + engine_power *
               wing_span +
               max_speed * wing_span + cruise_speed * stall_speed, data = AircraftData)

newdata = data.frame(engine_type = "Piston", engine_power=300, max_speed =240, cruise_speed =200,
                     stall_speed=70, wing_span=400)
predict(bestModel, newdata, interval="predict", level = .95)
```

```
##          fit      lwr      upr
## 1 2324571 1765506 2883635
```

The model predicts a fitted price of 2,324,571 USD for a piston-engine aircraft with the given specifications: engine_power = 300, max_speed = 240, cruise_speed = 200, stall_speed = 70, and wing_span = 400. The 95% prediction interval ranges from 1,765,506 USD to 2,883,635 USD.

7. Checking the remaining Regression Assumptions

7.1. Linearity Assumption

The linear regression model assumes that there is a straight-line (linear) relationship between the predictors and the aircraft price. For identifying the non linearity we are using the residual plot.

```
#Residual Plot
bestModel = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
               max_speed + cruise_speed + stall_speed + wing_span +
               factor(engine_type) * engine_power + factor(engine_type) * max_speed +
               factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
               engine_power * max_speed + engine_power * cruise_speed + engine_power *
               wing_span +
               max_speed * wing_span + cruise_speed * stall_speed, data = AircraftData)

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

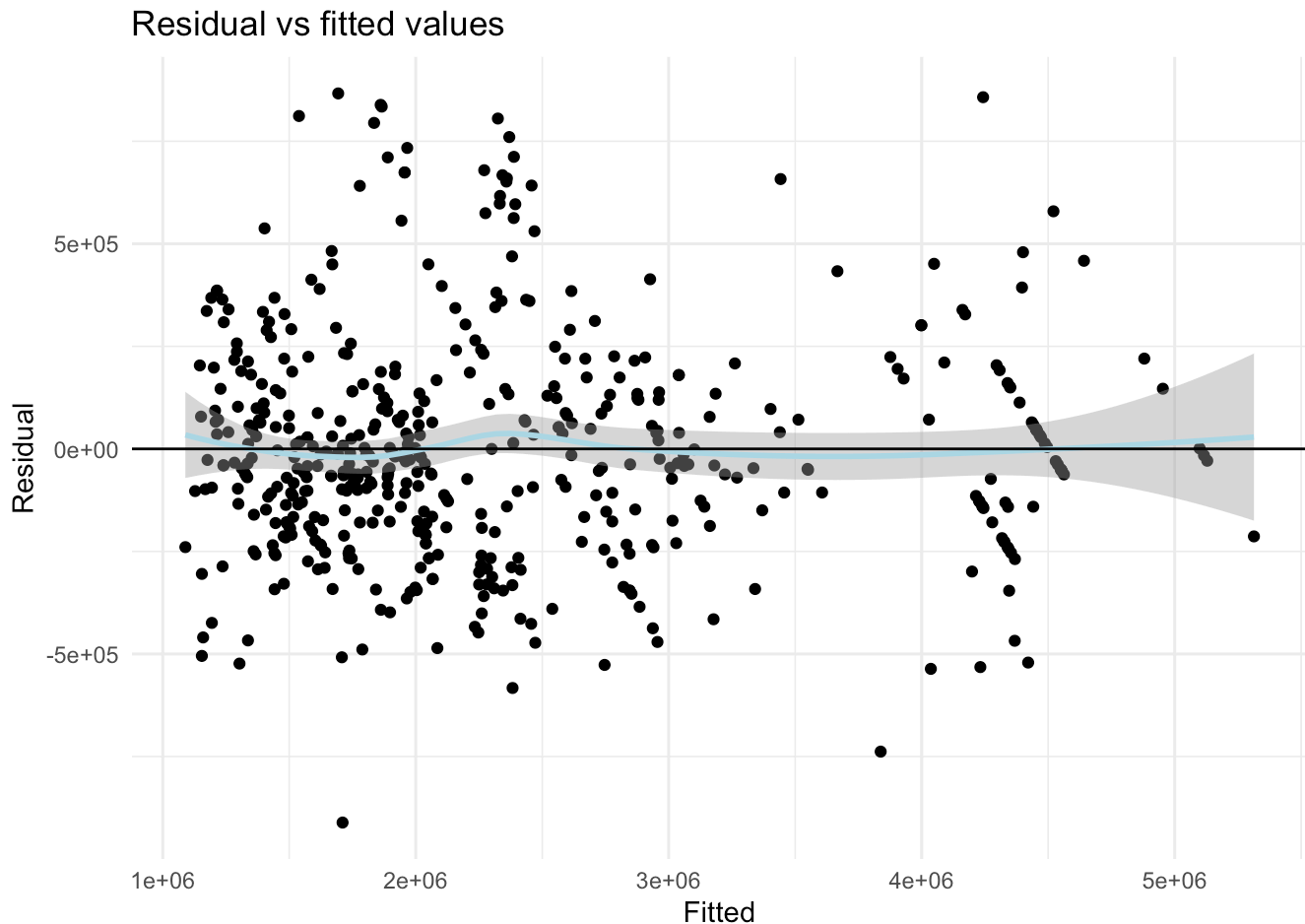
```
## [1] 0.9247535
```

```
sigma(bestModel)
```

```
## [1] 279449.2
```

```
ggplot(bestModel, aes(x=.fitted, y=.resid)) +
  geom_point() + geom_smooth(color = "lightblue") +
  geom_hline(yintercept = 0) +
  labs(title = "Residual vs fitted values",
       x = "Fitted",
       y = "Residual",
       color = "Method") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



There appears to be no pattern of the residuals at all. The R^2_{adj} of the bestModel is 0.9247535 indicates the variation in Price that can be explained by this model is 92.48% with RMSE = 279449.2. We can conclude that the above model named bestModel, is the best fit model to predict Price among the models we considered.

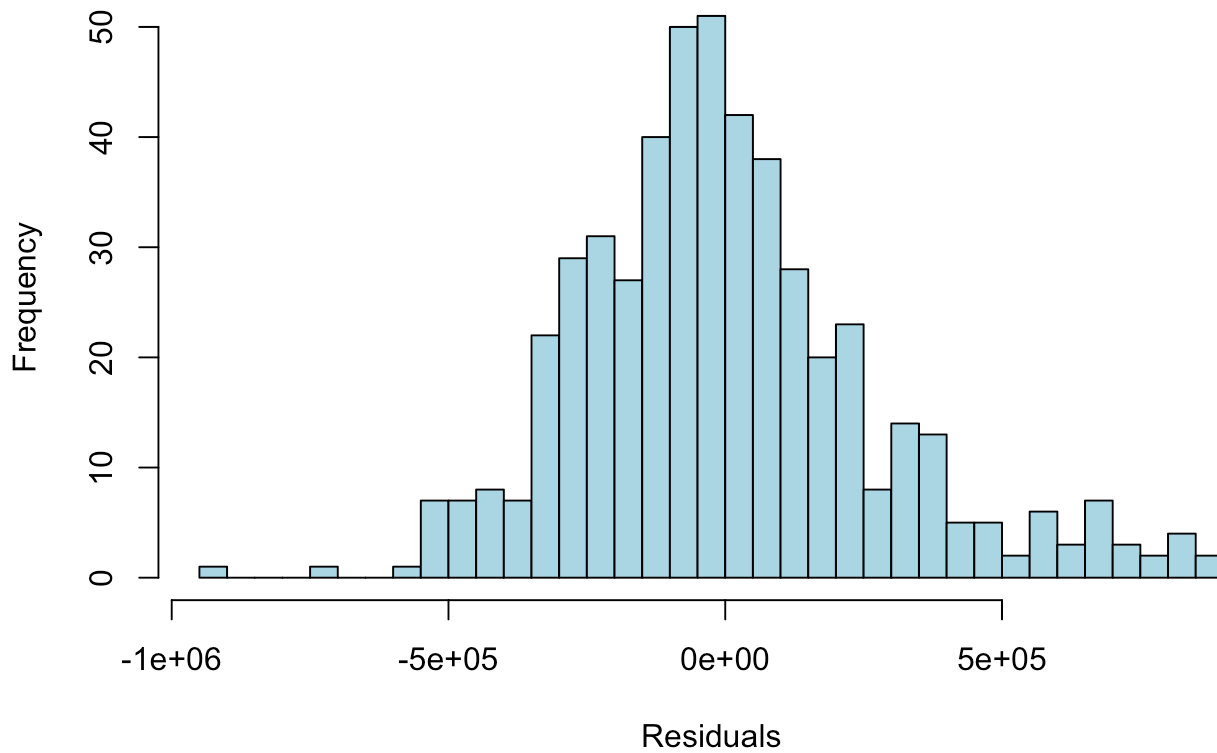
We will now extract the residuals and plot this data to analyze the histogram, qq plot, and residual vs fitted plot.

7.2. Normality Assumption

7.2.1 Histogram of Residuals

```
residuals = resid(bestModel)
hist(residuals, main="Histogram of Residuals - Best Model", xlab="Residuals", col="lightblue", b
order="black", breaks = 30)
```

Histogram of Residuals - Best Model

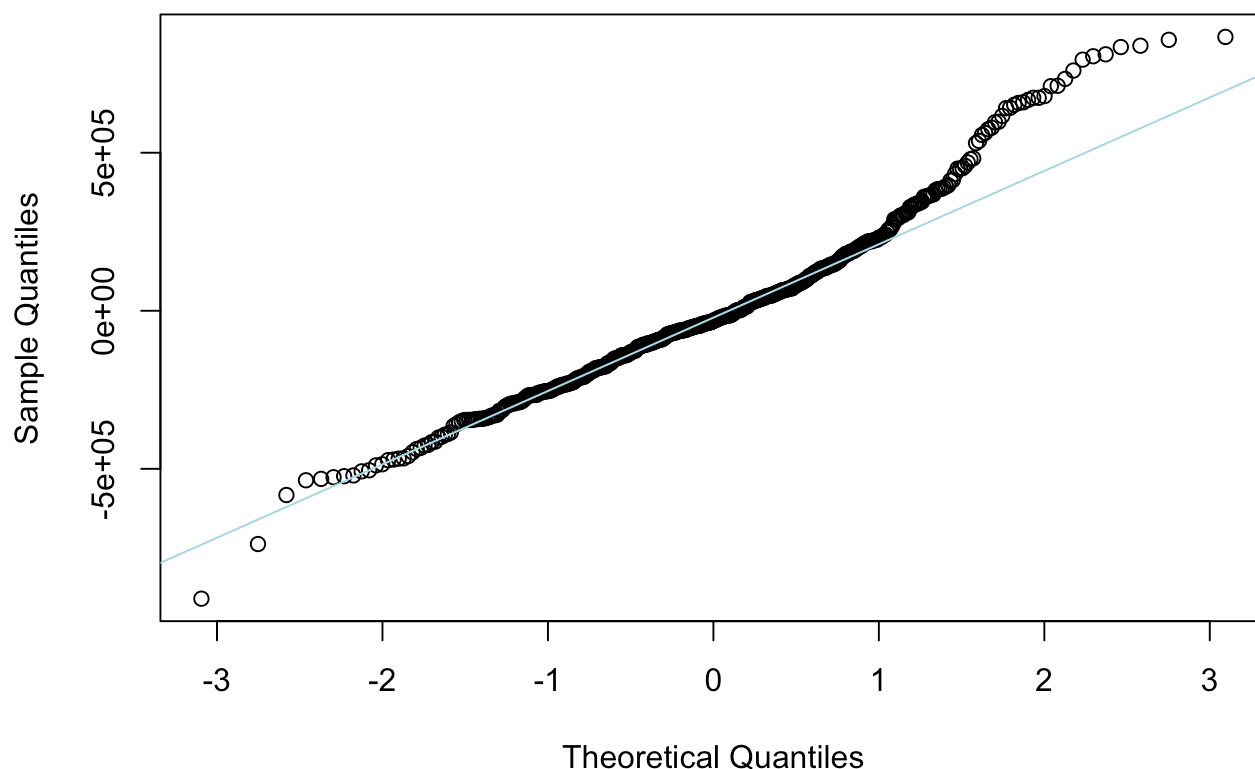


The histogram appears to be somewhat centered around zero, which is good. However, the shape does not perfectly follow a bell curve (normal distribution). The bars are somewhat skewed—there seems to be a longer right tail, suggesting potential positive skewness. Next, we will confirm whether the residuals deviate from normality, using the Q-Q Plot.

7.2.2 Q-Q Plot of Residuals

```
# Q-Q plot of residuals
qqnorm(residuals, main="Q-Q Plot of Residuals")
qqline(residuals, col="lightblue")
```

Q-Q Plot of Residuals

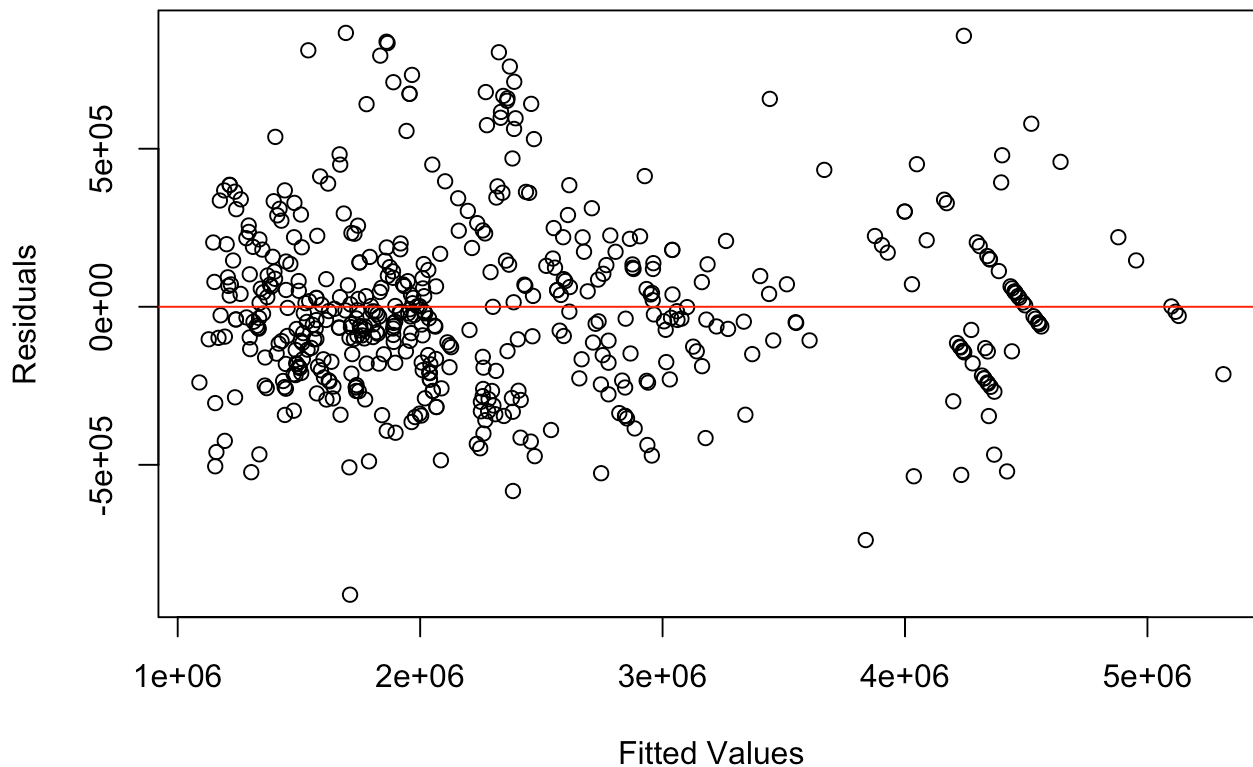


The points deviate significantly at both ends (left and right tails), indicating non-normality in the residuals. The right tail (higher values) has several points far above the line, suggesting the presence of high-value outliers (potentially high-priced aircraft affecting the model). The left tail (lower values) slightly deviates as well. Most points near the center follow the red line well, which means most residuals are approximately normal. Since the p-value is very small, your residuals do not follow a normal distribution. This violates the normality assumption in linear regression, which can impact hypothesis testing and confidence intervals.

7.2.3 Residuals vs Fitted Plot

```
# Plot Residuals vs Fitted Values
plot(fitted(bestModel), residuals, main="Residuals vs Fitted", xlab="Fitted Values", ylab="Residuals")
abline(h=0, col="red")
```

Residuals vs Fitted



The Residuals vs. Fitted plot shows signs of heteroscedasticity, as residuals exhibit increasing variance for larger fitted values, indicating that the model struggles to predict high aircraft prices accurately. Additionally, a slight curvature suggests potential non-linearity, meaning the model might be missing key interaction or polynomial effects. There are also outliers that could be influential observations, impacting the model's stability. These issues violate regression assumptions, potentially leading to unreliable standard errors and biased predictions.

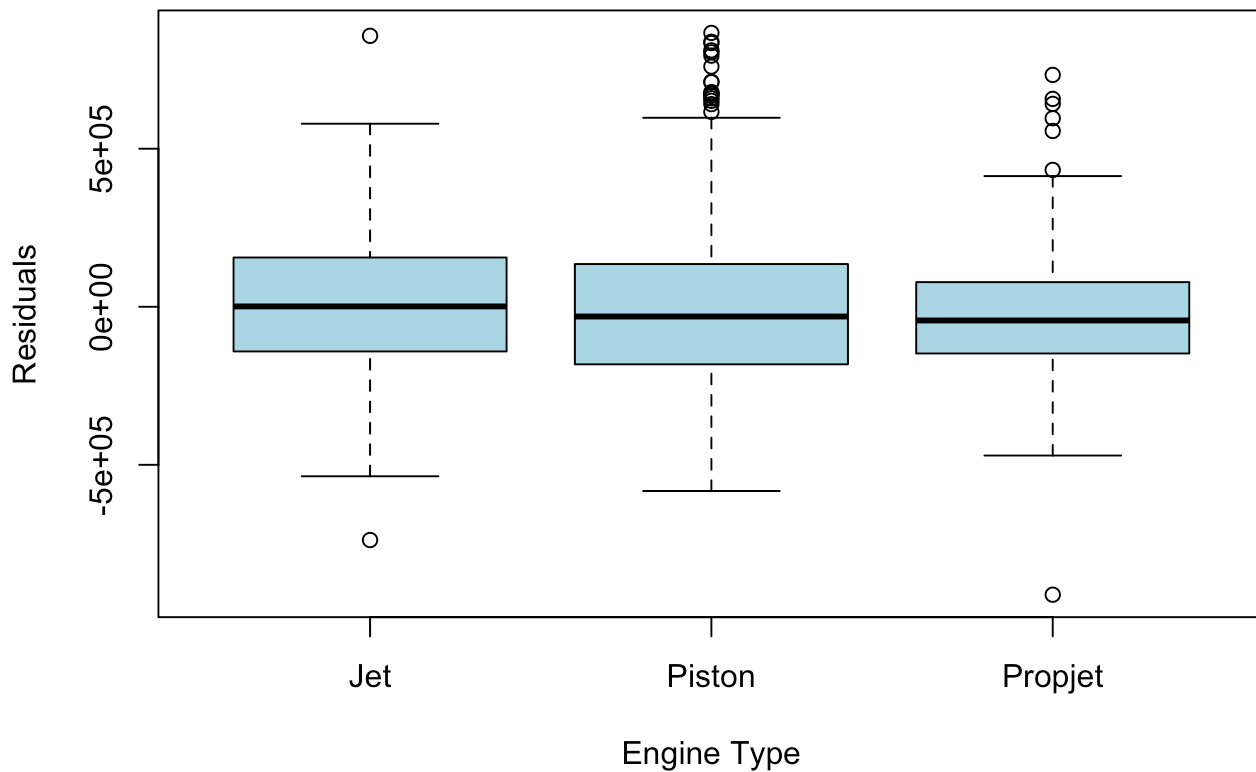
Overall, this Residuals vs. Fitted plot confirms that your model suffers from heteroscedasticity and possible outliers. Applying log transformation, WLS regression, or robust regression techniques should improve the model's reliability

7.3. Independence Assumption

In linear regression, the error terms ($\epsilon_1, \epsilon_2, \epsilon_3$, etc.) should not be related to each other. This means the value of one error doesn't give any clue about the next one. When this assumption is violated (e.g., errors are correlated), it often happens in time-series data, where observations are taken over time. As we do not have any time series data or spatial variable, we considered to plot the residuals vs engine_type category.

```
residuals <- residuals(bestModel)
residuals_data <- data.frame(Residuals = residuals, EngineType = AircraftData$engine_type)
boxplot(Residuals ~ EngineType, data = residuals_data,
        main = "Residuals vs Engine Type",
        xlab = "Engine Type",
        ylab = "Residuals",
        col = "lightblue")
```

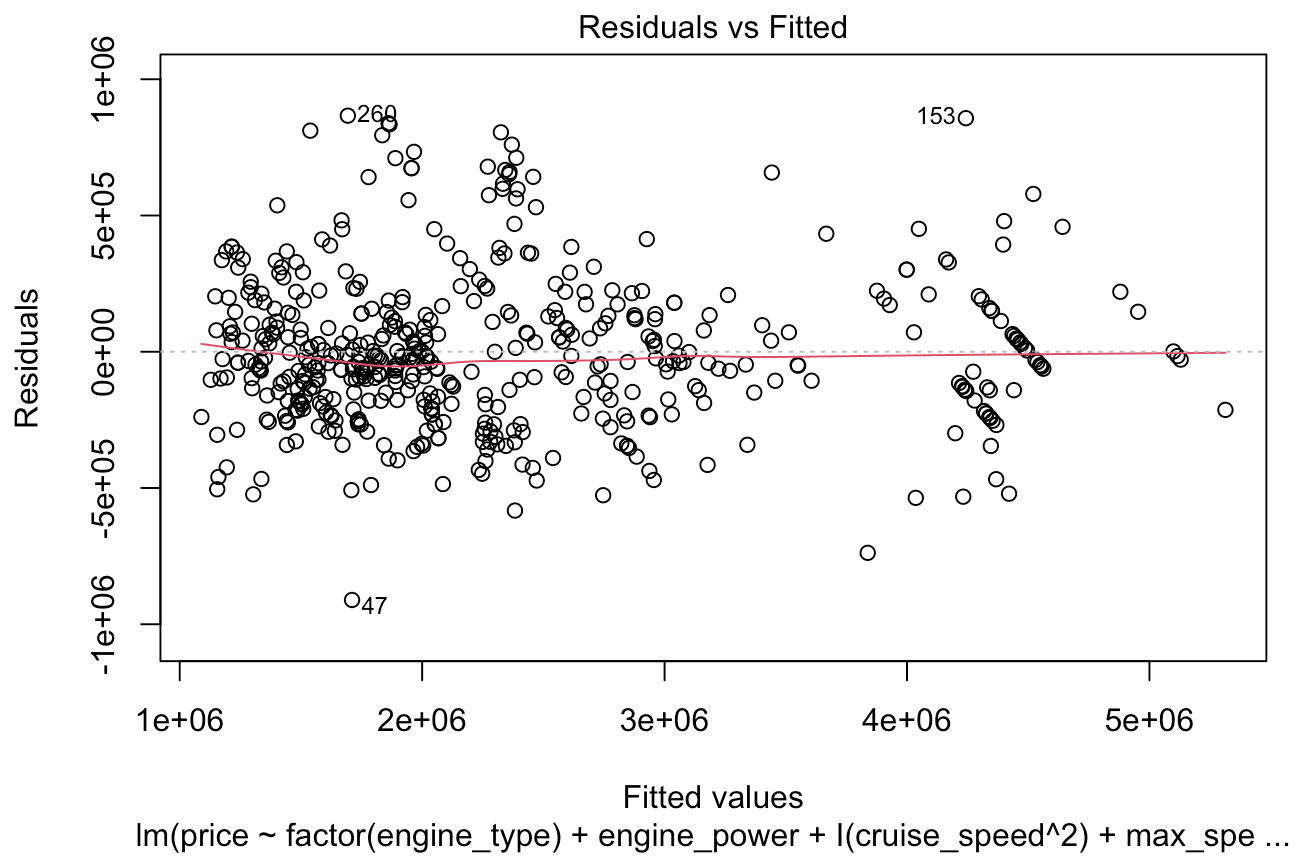
Residuals vs Engine Type



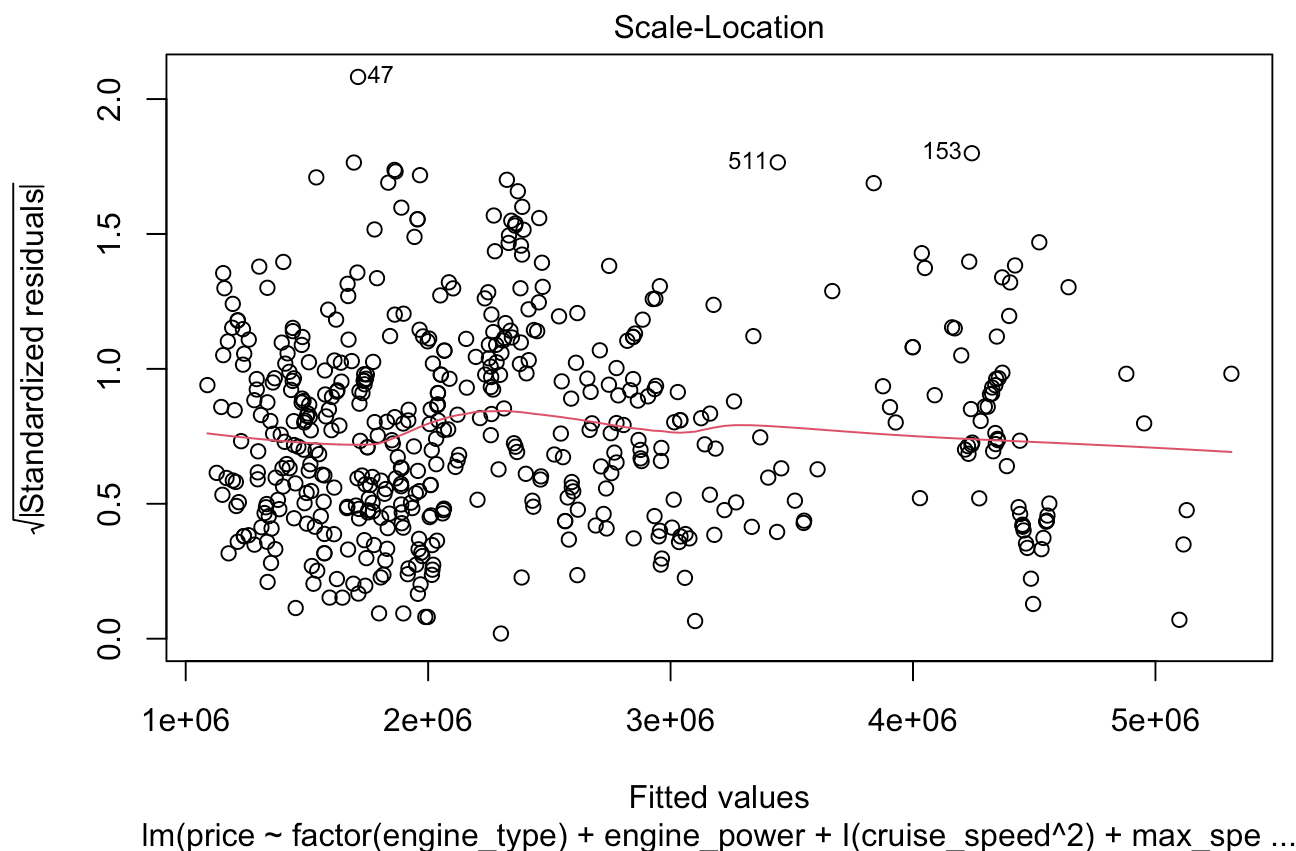
The box plot above indicates that the residuals for different engine types are centered around zero and do not display any noticeable patterns. Therefore, the independence assumption is satisfied.

7.4. Equal Variance Assumption

```
bestModel = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +  
                max_speed + cruise_speed + stall_speed + wing_span +  
                factor(engine_type) * engine_power + factor(engine_type) * max_speed +  
                factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +  
                engine_power * max_speed + engine_power * cruise_speed + engine_power *  
wing_span +  
                max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat  
a)  
  
#residuals plot  
plot(bestModel, which=1)
```

```
#scale-location plot  
plot(bestModel, which=3)
```



```
#The Breusch-Pagan Test
bptest(bestModel)
```

```
##
## studentized Breusch-Pagan test
##
## data: bestModel
## BP = 67.112, df = 21, p-value = 1.013e-06
```

H_0 : heteroscedasticity is not present (homoscedasticity)

H_1 : heteroscedasticity is present

Here we reject the null hypothesis ($p\text{-value} < 0.05$), so we conclude we do have heteroscedasticity. Maybe we can try a higher order model to get rid of this issue.

To resolve we can apply log transformation

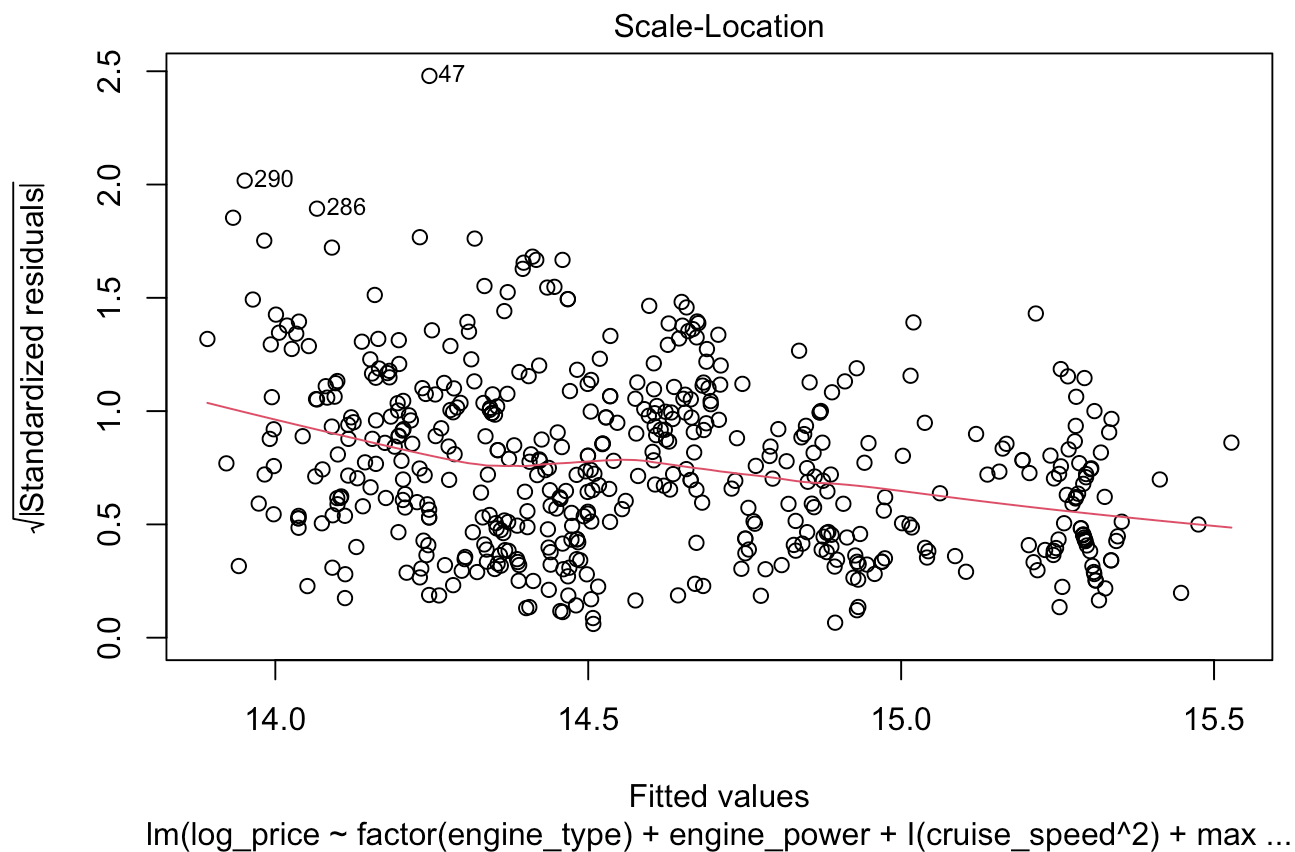
7.4.1. Log transformation

```
AircraftData$log_price = log(AircraftData$price)
bestModel_log <- lm(log_price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
                    max_speed + cruise_speed + stall_speed + wing_span +
                    factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                    factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                    engine_power * max_speed + engine_power * cruise_speed + engine_power *
                    wing_span +
                    max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

bptest(bestModel_log)
```

```
##
## studentized Breusch-Pagan test
##
## data: bestModel_log
## BP = 124.51, df = 21, p-value < 2.2e-16
```

```
plot(bestModel_log, which=3)
```

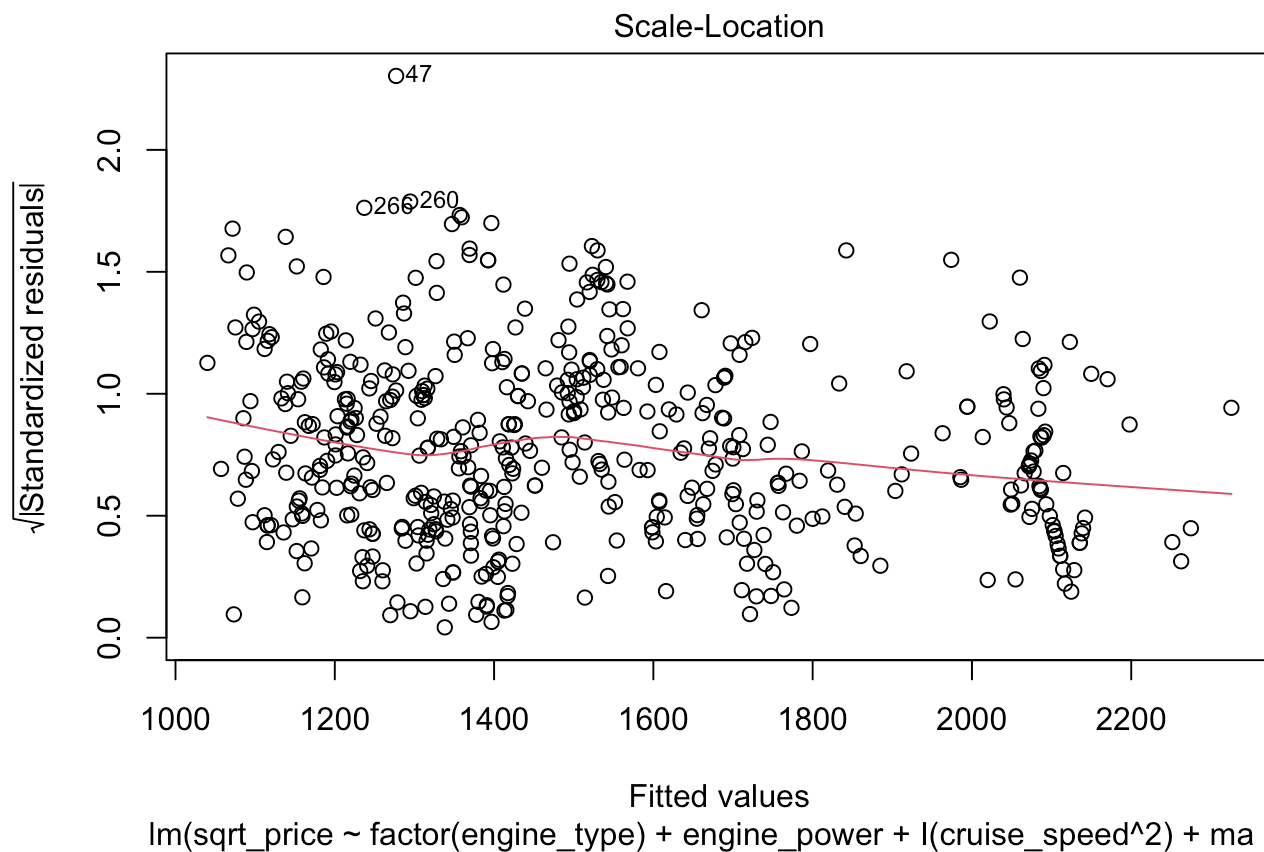


```
#sqrt transformation
AircraftData$sqrt_price = sqrt(AircraftData$price)
bestModel_sqrt <- lm(sqrt_price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
                      max_speed + cruise_speed + stall_speed + wing_span +
                      factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                      factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                      engine_power * max_speed + engine_power * cruise_speed + engine_power *
                      wing_span +
                      max_speed * wing_span + cruise_speed * stall_speed, data = AircraftData)

bptest(bestModel_sqrt)
```

```
##
## studentized Breusch-Pagan test
##
## data: bestModel_sqrt
## BP = 93.847, df = 21, p-value = 3.473e-11
```

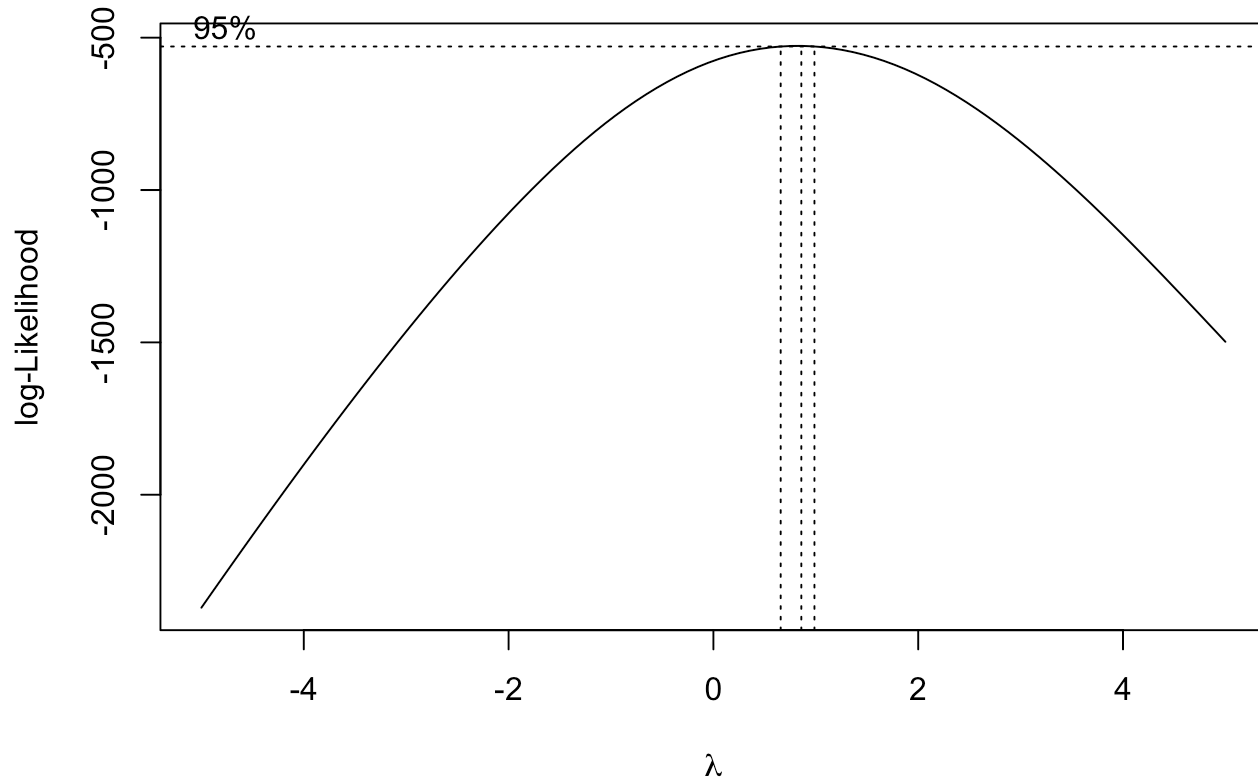
```
plot(bestModel_sqrt, which=3)
```



p-value < 2.2e-16 is less than 0.05, so we reject H_0 . Therefore there exists heteroscedasticity. Now we apply Box-Cox transformations.

7.4.2. Box-Cox transformations

```
bc=boxcox(bestModel,lambda=seq(-5,5))
```



```
bestlambda=bc$x[which(bc$y==max(bc$y))]  
(bestlambda)
```

```
## [1] 0.8585859
```

```
#The Breusch-Pagan Test  
bptest(bestModel)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: bestModel  
## BP = 67.112, df = 21, p-value = 1.013e-06
```

```

bcmodel=lm((((price^0.8585859)-1)/0.8585859) ~ factor(engine_type) + engine_power + I(cruise_spe
ed^2) +
                max_speed + cruise_speed + stall_speed + wing_span +
                factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)
#summary(bcmodel)
bptest(bcmodel)

```

```

##
## studentized Breusch-Pagan test
##
## data: bcmodel
## BP = 71.935, df = 21, p-value = 1.714e-07

```

The Box-Cox transformation (using $\lambda = 0.8586$) helped to adjust the dependent variable, but it did not completely solve the heteroscedasticity problem in the model.

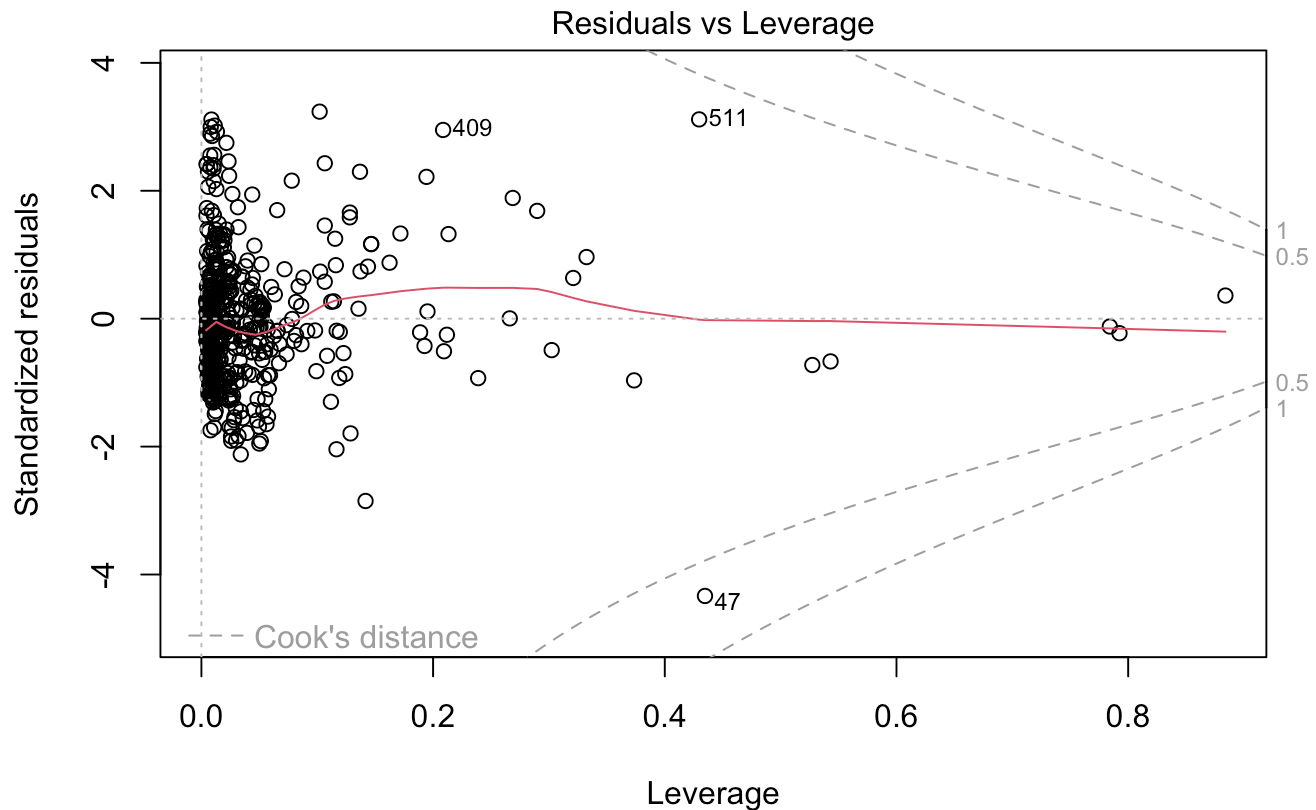
7.5. Outliers

```

bestModel = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
                max_speed + cruise_speed + stall_speed + wing_span +
                factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                engine_power * max_speed + engine_power * cruise_speed + engine_power *
wing_span +
                max_speed * wing_span + cruise_speed * stall_speed, data = AircraftDat
a)

plot(bestModel,which=5)

```

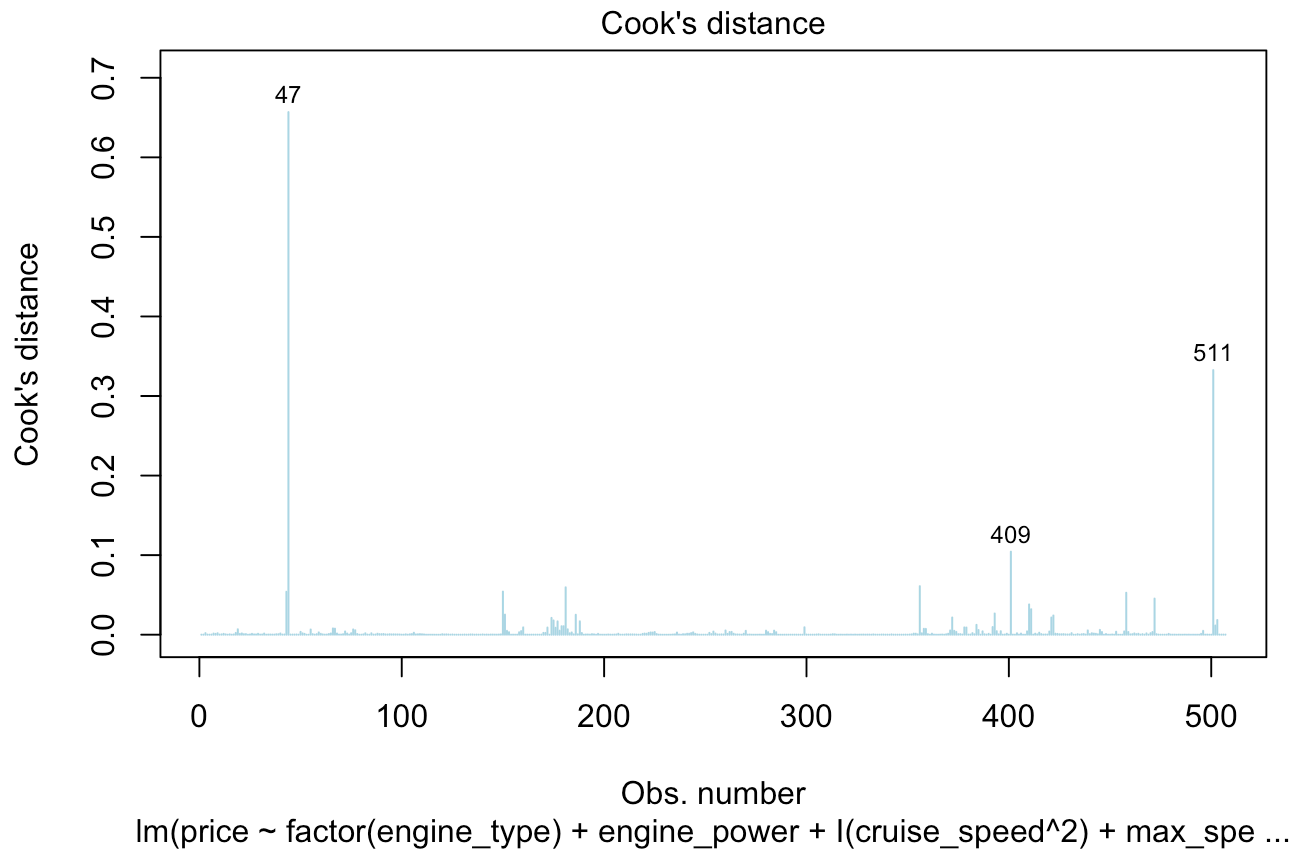


lm(price ~ factor(engine_type) + engine_power + l(cruise_speed^2) + max_spe ...

```
AircraftData[cooks.distance(bestModel)>0.5,]
```

```
##           model_name engine_type engine_power max_speed cruise_speed
## 47 100 Darter (S.L. Industries)   Propjet      1050      158      126
##   stall_speed fuel_tank all_eng_roc out_eng_roc takeoff_distance
## 47         52      216      650      1706      1525
##   landing_distance empty_weight length wing_span range price log_price
## 47         12500      5829      402      438      600 8e+05 13.59237
##   sqrt_price
## 47   894.4272
```

```
#which =4 only prints the cook distance plot.
plot(bestModel,pch=18,col="lightblue",which=c(4))
```



```
#cook distance
AircraftData[cooks.distance(bestModel)>0.5,]
```

```
##           model_name engine_type engine_power max_speed cruise_speed
## 47 100 Darter (S.L. Industries)   Propjet      1050      158      126
##   stall_speed fuel_tank all_eng_roc out_eng_roc takeoff_distance
## 47         52      216      650      1706      1525
##   landing_distance empty_weight length wing_span range price log_price
## 47         12500      5829    402      438    600 8e+05 13.59237
##   sqrt_price
## 47   894.4272
```

```
#leverages points plots
plot(bestModel,pch=18,col="lightblue",which=c(4)) #which =4 only prints the cook distance plot.
```

From here you can see that data point 47 is an influential case and we need to work for that point so find cook distance. Observation 47 (0.65) well above this threshold, indicating they are highly influential points.


```
lev=hatvalues(bestModel)
p = length(coef(bestModel))
n = nrow(AircraftData)
outlier2p = lev[lev>(2*p/n)]
outlier3p = lev[lev>(3*p/n)]
print("h_I>2p/n, outliers are")
```

```
## [1] "h_I>2p/n, outliers are"
```

```
print(outlier2p)
```

```
##          14          45          47          48          49          50          153
## 0.11211970 0.19426975 0.43469053 0.13561182 0.09810647 0.09139571 0.10206065
##          154          155          156          175          176          177          179
## 0.37351354 0.30238020 0.20935904 0.79251544 0.26615958 0.33245314 0.32092689
##          181          184          185          186          187          188          189
## 0.17178242 0.14374943 0.14654533 0.14654533 0.26874702 0.16242552 0.08810582
##          190          191          192          194          305          362          377
## 0.10251860 0.11437497 0.11659846 0.12811915 0.11543268 0.14168106 0.11898507
##          378          379          388          389          390          400          401
## 0.12871465 0.12417699 0.19259490 0.18877506 0.23889980 0.11180028 0.52735558
##          409          412          417          418          419          423          424
## 0.20863546 0.10646193 0.11603622 0.13688508 0.10649390 0.78416687 0.21189792
##          425          426          427          428          429          430          450
## 0.19516285 0.11945897 0.11672524 0.13734294 0.21329228 0.54310889 0.12264173
##          468          469          480          482          511          512          513
## 0.28983059 0.09922146 0.10850892 0.88392524 0.42966304 0.10650121 0.12815809
```

```
print(outlier3p)
```

```
##          45          47          48          154          155          156          175          176
## 0.1942698 0.4346905 0.1356118 0.3735135 0.3023802 0.2093590 0.7925154 0.2661596
##          177          179          181          184          185          186          187          188
## 0.3324531 0.3209269 0.1717824 0.1437494 0.1465453 0.1465453 0.2687470 0.1624255
##          362          388          389          390          401          409          418          423
## 0.1416811 0.1925949 0.1887751 0.2388998 0.5273556 0.2086355 0.1368851 0.7841669
##          424          425          428          429          430          468          482          511
## 0.2118979 0.1951629 0.1373429 0.2132923 0.5431089 0.2898306 0.8839252 0.4296630
```

Here 47 leverage point is 0.4346 and cook distance 0.65 so this is high influential points and remove and check the result.

```
#ouliers removed
AircraftData_NoOutliers = read.csv("/Users/anithajoseph/Documents/aircraft_price_no_outliers.csv")
sum(is.na(AircraftData_NoOutliers))
```

```
## [1] 10
```

```
Aircraft_NoOutliers = na.omit(AircraftData_NoOutliers)

bestModel_NoOutliers = lm(price ~ factor(engine_type) + engine_power + I(cruise_speed^2) +
                           max_speed + cruise_speed + stall_speed + wing_span +
                           factor(engine_type) * engine_power + factor(engine_type) * max_speed +
                           factor(engine_type) * cruise_speed + factor(engine_type) * wing_span +
                           engine_power * max_speed + engine_power * cruise_speed + engine_power *
                           wing_span +
                           max_speed * wing_span + cruise_speed * stall_speed, data = Aircraft_No
                           Outliers)

summary(bestModel_NoOutliers)$adj.r.square
```

```
## [1] 0.9273244
```

```
sigma(bestModel_NoOutliers)
```

```
## [1] 274264.5
```

```
summary(bestModel)$adj.r.square
```

```
## [1] 0.9247535
```

```
sigma(bestModel)
```

```
## [1] 279449.2
```

8. Summary

8.1. Key Findings

Our best model predicts aircraft price based on engine type, performance metrics such as cruise speed, engine power, stall speed, max speed, and design factors like wing span. On analysis of the best model, we can see the below findings:

1. Engine Type:

Jet type is the reference category

Piston price is less than Jet by approximately 4.39 million USD

Propjet price is less than Jet by approximately 2.76 million USD

2. Cruise Speed impact:

Linear term: negative effect on price

Quadratic term: positive effect with exponential price increase at higher speeds

3. Negative Effects: for stall speed and wing span

4. Interaction terms:

For Propjets, Higher max speed benefits, but higher cruise speed reduces price.

Engine Power + Max Speed / Cruise Speed: Slight negative price impact

Max Speed + Wing Span: Contributes positively to price

Jet Engines: Predicted price is 5,823,241.56 USD when all other variables are zero. Jets do not experience price reductions like piston engines or propjets. Negative effects come from stall speed and wing span, with mixed interactions across other factors.

Piston Engines: Predicted price is about 4.39 million USD lower than jets. Engine power decreases price by 981 units per unit increase, while max speed and cruise speed positively impact price. Greater wing span also increases price, but stall speed and wing span have overall negative effects.

PropJets Engines: Predicted price is approximately 2.76 million USD lower than jets. Engine power reduces price by 2,284.54 units per unit increase, but max speed and wing span increase price. Higher cruise speed slightly decreases price, while stall speed and wing span have negative effects.

8.2. Improvements

8.2.1 To address heteroscedasticity for Equal Variance

Since the best model has not met with homoscedasticity or equal variance assumption. It could be resolved by doing some of the following methods

1. applying appropriate transformation of dependent variable (other than log, sqrt. Since we tried it)
2. Use WLS instead of Ordinary Least Squares (OLS) to assign weights to observations inversely proportional to their variance.
3. Check for omitted variables by introducing variables may help capture the relationships causing heteroscedasticity
4. Identify and consider removing influential observations or outliers that might cause variance issues.

8.2.2 Reduction of terms with VIF > 5

Given that the predictors engine_power and cruise_speed are statistically and theoretically significant for our dataset, we chose to retain them despite their VIF being greater than 5. While these terms hold theoretical importance, removing them might yield alternative models capable of predicting prices effectively. Further analysis is necessary to evaluate their relevance, and we suggest examining whether the newly developed model could address the heteroscedasticity problem.