STAT 139 Final Homework Exploratory Data Analysis

Harry Hu, Changming Liu, Wendy Wang, Yixuan Li, Rongzhi Chen

2025-11-22

Background

New cars vary widely in price, and buyers often care about how observable features translate into a higher or lower sticker price. In this project we study how technical characteristics of a car (such as engine power, number of cylinders, city and highway fuel economy, and popularity) and broader design choices (brand, size class, body style, fuel type, transmission, driven wheels, and number of doors) are associated with the manufacturer suggested retail price (MSRP). We focus on a linear regression model with MSRP as the outcome and these features as main effects only, so that each coefficient can be interpreted as the average difference in price associated with a one unit change in a continuous predictor or with belonging to a particular category, holding all other variables fixed. This setup allows us to quantify which car features are most strongly related to price and to compare the relative importance of performance, comfort, and branding related variables.

Data Sources

We use the "Car Features and MSRP" dataset from Kaggle, originally scraped from Edmunds and Twitter. The raw data file data.csv contains 11,914 rows and 16 columns, where each row corresponds to a specific car model and each column records one attribute of that model. Our continuous variables include engine horsepower, number of cylinders, city and highway miles per gallon, popularity, and MSRP. The remaining variables are treated as categorical: make, model, model year, engine fuel type, transmission type, driven wheels, number of doors, market category, vehicle size, and vehicle style. We perform basic cleaning by removing records with "N/A" in Market.Category or "UNKNOWN" in Transmission.Type and coding all categorical variables as factors. The main effects of these continuous and categorical predictors are then used as inputs in our linear model for MSRP.

data source: https://www.kaggle.com/datasets/CooperUnion/cardataset?resource=download

```r
dat <- read.csv("data.csv")
dim(dat)
```

```
## [1] 11914    16
```

```r
names(dat)
```

```
##  [1] "Make"              "Model"             "Year"
##  [4] "Engine.Fuel.Type"  "Engine.HP"         "Engine.Cylinders"
##  [7] "Transmission.Type" "Driven_Wheels"     "Number.of.Doors"
## [10] "Market.Category"   "Vehicle.Size"      "Vehicle.Style"
## [13] "highway.MPG"       "city.mpg"          "Popularity"
## [16] "MSRP"
```

```r
str(dat)
```

```
## 'data.frame':    11914 obs. of  16 variables:
##  $ Make             : chr  "BMW" "BMW" "BMW" "BMW" ...
##  $ Model            : chr  "1 Series M" "1 Series" "1 Series" "1 Series" ...
```

```
##  $ Year             : int  2011 2011 2011 2011 2011 2012 2012 2012 2012 2013 ...
##  $ Engine.Fuel.Type : chr  "premium unleaded (required)" "premium unleaded (required)" "premium unlea
##  $ Engine.HP        : int  335 300 300 230 230 230 300 300 230 230 ...
##  $ Engine.Cylinders : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ Transmission.Type: chr  "MANUAL" "MANUAL" "MANUAL" "MANUAL" ...
##  $ Driven_Wheels    : chr  "rear wheel drive" "rear wheel drive" "rear wheel drive" "rear wheel drive
##  $ Number.of.Doors  : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Market.Category  : chr  "Factory Tuner,Luxury,High-Performance" "Luxury,Performance" "Luxury,High-
##  $ Vehicle.Size     : chr  "Compact" "Compact" "Compact" "Compact" ...
##  $ Vehicle.Style    : chr  "Coupe" "Convertible" "Coupe" "Coupe" ...
##  $ highway.MPG      : int  26 28 28 28 28 28 26 28 28 27 ...
##  $ city.mpg         : int  19 19 20 18 18 18 17 20 18 18 ...
##  $ Popularity       : int  3916 3916 3916 3916 3916 3916 3916 3916 3916 3916 ...
##  $ MSRP             : int  46135 40650 36350 29450 34500 31200 44100 39300 36900 37200 ...
```

Our data contains 11914 rows and 16 columns, and the variable names are shown above.

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(knitr)

# List continuous variables
cont_vars <- c("Engine.HP", "Engine.Cylinders",
               "highway.MPG", "city.mpg",
               "Popularity", "MSRP")

# Build summary table directly (no summarise -> no warning)
cont_summary <- data.frame(
  Variable    = cont_vars,
  non_missing = sapply(dat[cont_vars], function(x) sum(!is.na(x))),
  missing     = sapply(dat[cont_vars], function(x) sum(is.na(x))),
  mean        = sapply(dat[cont_vars], function(x) mean(x, na.rm = TRUE)),
  median      = sapply(dat[cont_vars], function(x) median(x, na.rm = TRUE)),
  sd          = sapply(dat[cont_vars], function(x) sd(x, na.rm = TRUE)),
  IQR         = sapply(dat[cont_vars], function(x) IQR(x, na.rm = TRUE))
)

kable(cont_summary, caption = "Summary of Continuous Variables")
```

Table 1: Summary of Continuous Variables

	Variable	non_missing	missing	mean	median	sd	IQR
Engine.HP	Engine.HP	11845	69	249.386070	227	109.191870	130.00
Engine.Cylinders	Engine.Cylinders	11884	30	5.628829	6	1.780559	2.00
highway.MPG	highway.MPG	11914	0	26.637485	26	8.863001	8.00

	Variable	non_missing	missing	mean	median	sd	IQR
city.mpg	city.mpg	11914	0	19.733255	18	8.987798	6.00
Popularity	Popularity	11914	0	1554.911197	1385	1441.855347	1460.00
MSRP	MSRP	11914	0	40594.737032	29995	60109.103604	21231.25

Categorical variables:

```
library(dplyr)
library(knitr)
library(tidyr)

# Choose the variables you want to treat as categorical
cat_vars <- c(
  "Make",
  "Model",
  "Year",              # Treat year as categorical
  "Engine.Fuel.Type",
  "Transmission.Type",
  "Driven_Wheels",
  "Number.of.Doors",    # numeric but categorical-ish
  "Market.Category",
  "Vehicle.Size",
  "Vehicle.Style"
)


topN <- 5

cat_top <- dat %>%
  # make sure all categorical vars are character (including Year, Number.of.Doors)
  mutate(across(all_of(cat_vars), as.character)) %>%
  select(all_of(cat_vars)) %>%
  pivot_longer(
    cols = everything(),
    names_to = "Variable",
    values_to = "Level"
  ) %>%
  filter(!is.na(Level)) %>%
  group_by(Variable, Level) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(Variable) %>%
  slice_max(n, n = topN)

kable(cat_top,
      caption = "Top 5 Levels per Categorical Variable")
```

Table 2: Top 5 Levels per Categorical Variable

Variable	Level	n
Driven_Wheels	front wheel drive	4787
Driven_Wheels	rear wheel drive	3371
Driven_Wheels	all wheel drive	2353
Driven_Wheels	four wheel drive	1403

Variable	Level	n
Engine.Fuel.Type	regular unleaded	7172
Engine.Fuel.Type	premium unleaded (required)	2009
Engine.Fuel.Type	premium unleaded (recommended)	1523
Engine.Fuel.Type	flex-fuel (unleaded/E85)	899
Engine.Fuel.Type	diesel	154
Make	Chevrolet	1123
Make	Ford	881
Make	Volkswagen	809
Make	Toyota	746
Make	Dodge	626
Market.Category	N/A	3742
Market.Category	Crossover	1110
Market.Category	Flex Fuel	872
Market.Category	Luxury	855
Market.Category	Luxury,Performance	673
Model	Silverado 1500	156
Model	Tundra	140
Model	F-150	126
Model	Sierra 1500	90
Model	Beetle Convertible	89
Number.of.Doors	4	8353
Number.of.Doors	2	3160
Number.of.Doors	3	395
Transmission.Type	AUTOMATIC	8266
Transmission.Type	MANUAL	2935
Transmission.Type	AUTOMATED_MANUAL	626
Transmission.Type	DIRECT_DRIVE	68
Transmission.Type	UNKNOWN	19
Vehicle.Size	Compact	4764
Vehicle.Size	Midsize	4373
Vehicle.Size	Large	2777
Vehicle.Style	Sedan	3048
Vehicle.Style	4dr SUV	2488
Vehicle.Style	Coupe	1211
Vehicle.Style	Convertible	793
Vehicle.Style	4dr Hatchback	702
Year	2015	2170
Year	2016	2157
Year	2017	1668
Year	2014	589
Year	2012	387

The table above summarizes the five most frequent categories for each categorical variable. It highlights strong imbalances in several variables. For example, most vehicles have 4 doors, use regular unleaded fuel, or fall into the Compact/Midsize size classes. The presence of entries such as "N/A" in Market.Category and "UNKNOWN" in Transmission.Type also indicates possible data-quality issues that may require cleaning or standardization.

```
# We drop rows whose Market.Category is N/A or Transmission.Type is UNKNOWN
dat_clean <- dat %>%
  filter(
    Market.Category != "N/A",
```

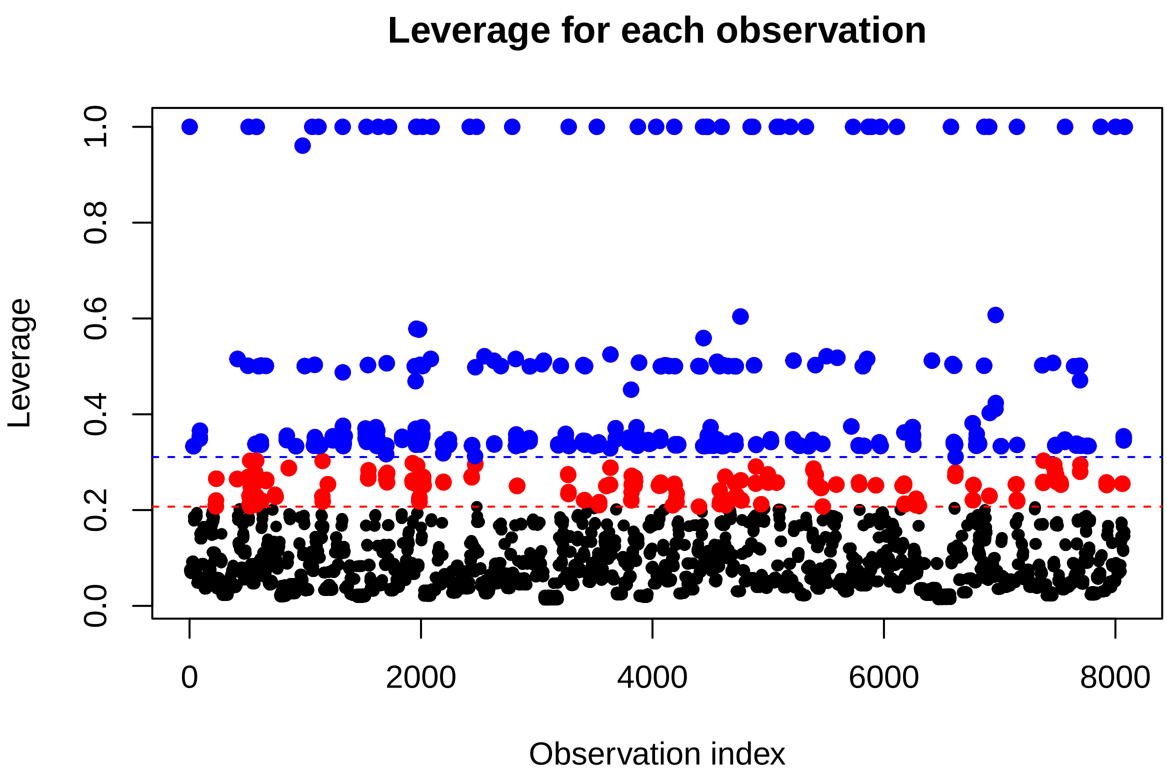
```r
    Transmission.Type != "UNKNOWN"
  )

dat_clean <- dat_clean %>%
  mutate(across(all_of(cat_vars), as.factor))
```

```r
all_predictors <- c(setdiff(cont_vars, "MSRP"), cat_vars)

formula_full <- as.formula(
  paste("MSRP ~", paste(all_predictors, collapse = " + "))
)

mod_full <- lm(formula_full, data = dat_clean)
# summary(mod_full)
```

```r
# leverage points
lev <- hatvalues(mod_full)
mean_lev <- mean(lev)

high_lev_idx <- which(lev > 2 * mean_lev)
very_high_lev_idx <- which(lev > 3 * mean_lev)

high_leverage_points <- dat[high_lev_idx, ]
head(high_leverage_points)
```

```
##      Make      Model Year               Engine.Fuel.Type Engine.HP
## 1     BMW 1 Series M 2011    premium unleaded (required)       335
## 33   FIAT 124 Spider 2017 premium unleaded (recommended)       160
## 34   FIAT 124 Spider 2017 premium unleaded (recommended)       160
## 35   FIAT 124 Spider 2017 premium unleaded (recommended)       160
## 88 Nissan      200SX 1996               regular unleaded       115
## 89 Nissan      200SX 1996               regular unleaded       115
##    Engine.Cylinders Transmission.Type     Driven_Wheels Number.of.Doors
## 1                 6            MANUAL  rear wheel drive               2
## 33                4            MANUAL  rear wheel drive               2
## 34                4            MANUAL  rear wheel drive               2
## 35                4            MANUAL  rear wheel drive               2
## 88                4            MANUAL front wheel drive               2
## 89                4            MANUAL front wheel drive               2
##                       Market.Category Vehicle.Size Vehicle.Style highway.MPG
## 1  Factory Tuner,Luxury,High-Performance      Compact         Coupe          26
## 33                        Performance      Compact   Convertible          35
## 34                        Performance      Compact   Convertible          35
## 35                        Performance      Compact   Convertible          35
## 88                                N/A      Compact         Coupe          36
## 89                                N/A      Compact         Coupe          36
##    city.mpg Popularity  MSRP
## 1        19       3916 46135
## 33       26        819 27495
## 34       26        819 24995
## 35       26        819 28195
## 88       26       2009  2000
## 89       26       2009  2000
```

```r
# ---- Visualization: Leverage ----
plot(
  lev,
  ylab = "Leverage",
  xlab = "Observation index",
  main = "Leverage for each observation",
  pch = 20
)
abline(h = 2 * mean_lev, lty = 2, col = "red")   # common rule-of-thumb cutoff
abline(h = 3 * mean_lev, lty = 2, col = "blue")  # more extreme cutoff

# highlight high leverage points
points(high_lev_idx, lev[high_lev_idx], pch = 19, col = "red")
if (length(very_high_lev_idx) > 0) {
  points(very_high_lev_idx, lev[very_high_lev_idx], pch = 19, col = "blue")
}
```

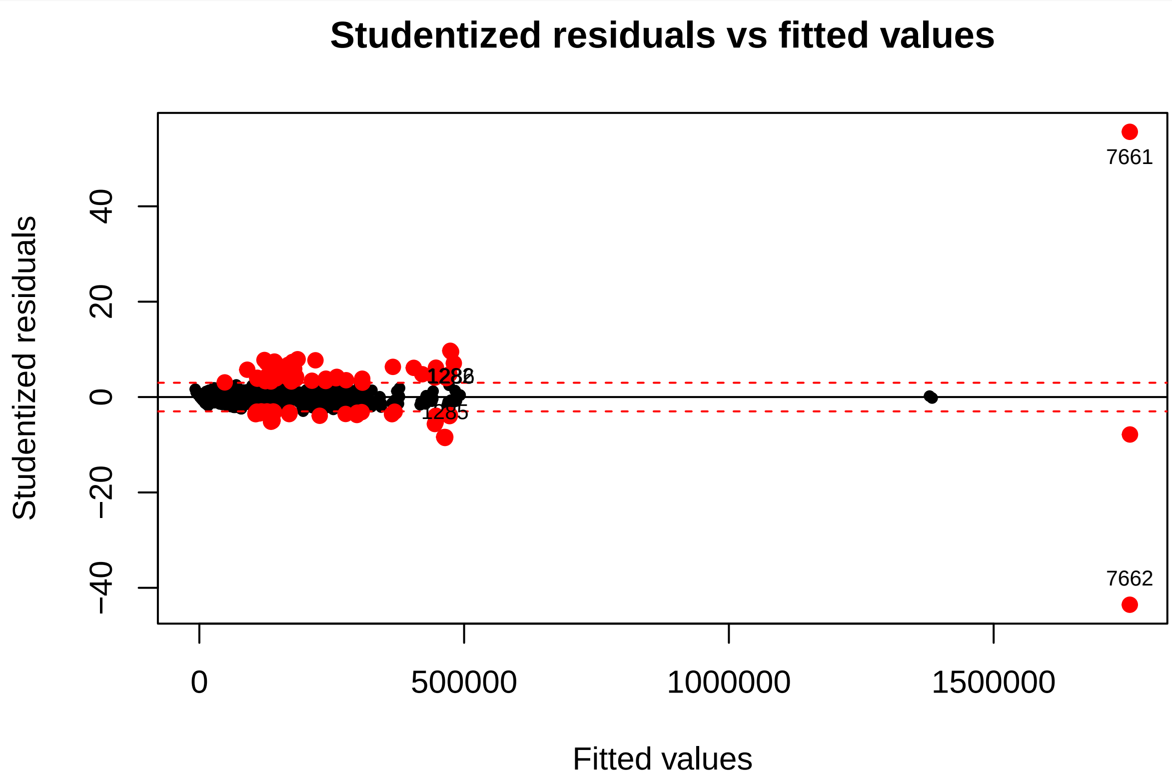


Most cars have relatively low leverage, but a non-trivial subset lies above the 2×mean and 3×mean reference lines, indicating observations whose combinations of predictors are unusual in this dataset. These high-leverage points correspond to relatively rare models such as sporty BMW and FIAT coupes and older Nissan 200SX entries. While high leverage alone does not imply a bad data point, these cars have the potential to strongly affect coefficient estimates and should be kept in mind when interpreting the fitted model.

```r
# outliers
stud_res <- rstudent(mod_full)
reg_outlier_idx <- which(abs(stud_res) > 3)
regression_outliers <- dat[reg_outlier_idx, ]
```

```r
# ---- Visualization: Outliers (studentized residuals) ----
plot(
```

```
  fitted(mod_full),
  stud_res,
  xlab = "Fitted values",
  ylab = "Studentized residuals",
  main = "Studentized residuals vs fitted values",
  pch = 20
)
abline(h = 0, lty = 1)                   # reference line
abline(h = c(-3, 3), lty = 2, col = "red")  # outlier cutoff

# highlight outliers
points(
  fitted(mod_full)[reg_outlier_idx],
  stud_res[reg_outlier_idx],
  pch = 19,
  col = "red"
)

# indices of the most extreme studentized residuals (top 5 by magnitude)
extreme_res_idx <- order(abs(stud_res), decreasing = TRUE)[1:5]

# label them on the plot: positive -> label below, negative -> label above
text(
  x = fitted(mod_full)[extreme_res_idx],
  y = stud_res[extreme_res_idx],
  labels = extreme_res_idx,
  pos = ifelse(stud_res[extreme_res_idx] > 0, 1, 3),  # 1 = below, 3 = above
  cex = 0.7
)
```



```r
# print the rows these points correspond to
cat("\nRows with most extreme studentized residuals:\n")
```

```
##
## Rows with most extreme studentized residuals:
```

```r
print(dat[extreme_res_idx, ])
```

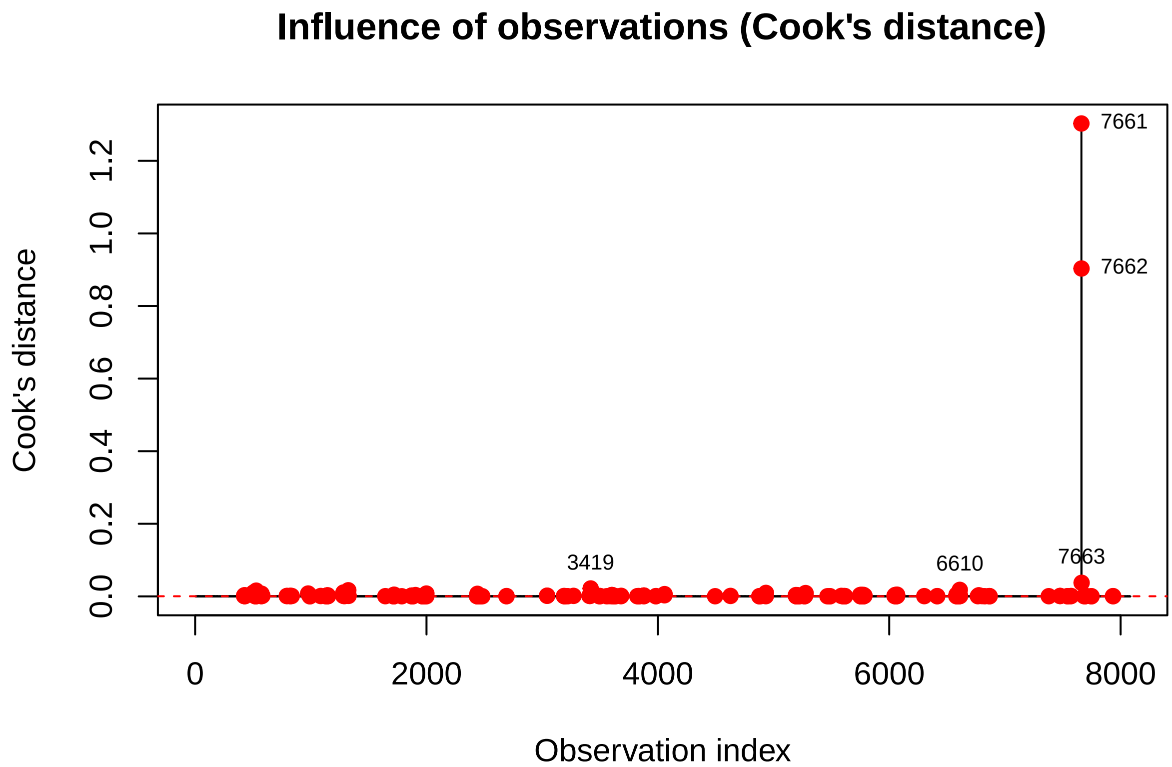
```
##        Make  Model Year Engine.Fuel.Type Engine.HP Engine.Cylinders
## 7661 Toyota Previa 1997 regular unleaded       161                4
## 7662 Toyota Previa 1997 regular unleaded       161                4
## 1282  Honda Accord 2017 regular unleaded       185                4
## 1286  Honda Accord 2017 regular unleaded       189                4
## 1285  Honda Accord 2017 regular unleaded       185                4
##      Transmission.Type     Driven_Wheels Number.of.Doors Market.Category
## 7661         AUTOMATIC  rear wheel drive               3             N/A
## 7662         AUTOMATIC   all wheel drive               3             N/A
## 1282         AUTOMATIC front wheel drive               4             N/A
## 1286            MANUAL front wheel drive               4             N/A
## 1285            MANUAL front wheel drive               4             N/A
##      Vehicle.Size     Vehicle.Style highway.MPG city.mpg Popularity  MSRP
## 7661      Compact Passenger Minivan          20       16       2031  2242
## 7662      Compact Passenger Minivan          19       15       2031  2728
## 1282      Midsize             Sedan          36       27       2202 26530
## 1286      Midsize             Sedan          32       23       2202 25415
## 1285      Midsize             Sedan          32       23       2202 25730
```

```r
# influential points
cooks <- cooks.distance(mod_full)
n <- nrow(dat)
p <- length(coef(mod_full)) - 1   # number of predictors

cook_cut <- 4 / (n - p - 1)
influential_idx <- which(cooks > cook_cut)
influential_points <- dat[influential_idx, ]
```

The residual plot shows that most fitted values have studentized residuals between -3 and 3, suggesting the linear model fits the bulk of the data reasonably well. However, a small number of points—most notably several Toyota Previa minivans and Honda Accord trims (labels 7661-7662 and 1282/1285/1286)—have extremely large negative residuals, with observed MSRPs far below what the model predicts. These records may reflect data-entry issues (e.g., heavily discounted or used vehicles recorded as new) or cars that are systematically underpriced relative to their features, and they warrant closer inspection or sensitivity analysis.

```r
# ---- Visualization: Influence (Cook's distance) ----
plot(
  cooks,
  type = "h",
  xlab = "Observation index",
  ylab = "Cook's distance",
  main = "Influence of observations (Cook's distance)"
)
abline(h = cook_cut, lty = 2, col = "red")  # rule-of-thumb cutoff

# highlight influential points
points(influential_idx, cooks[influential_idx], pch = 19, col = "red")
```

```
# indices of the largest Cook's distances (top 5)
top_cook_idx <- order(cooks, decreasing = TRUE)[1:5]

# label them on the plot
text(
  x = top_cook_idx,
  y = cooks[top_cook_idx],
  labels = top_cook_idx,
  pos = ifelse(cooks[top_cook_idx] > 0.6, 4, 3),
  cex = 0.7
)
```




```
# print the rows these points correspond to
cat("\nRows with largest Cook's distance:\n")
```

```
##
## Rows with largest Cook's distance:
```

```
print(dat[top_cook_idx, ])
```

```
##        Make  Model Year             Engine.Fuel.Type Engine.HP Engine.Cylinders
## 7661 Toyota Previa 1997             regular unleaded       161                4
## 7662 Toyota Previa 1997             regular unleaded       161                4
## 7663 Toyota Previa 1997             regular unleaded       161                4
## 3419  Mazda   CX-9 2016             regular unleaded       227                4
## 6610    BMW     M4 2016 premium unleaded (required)       425                6
##      Transmission.Type    Driven_Wheels Number.of.Doors
## 7661         AUTOMATIC rear wheel drive               3
## 7662         AUTOMATIC  all wheel drive               3
## 7663         AUTOMATIC rear wheel drive               3
```

```
## 3419         AUTOMATIC  all wheel drive               4
## 6610            MANUAL rear wheel drive               2
##                       Market.Category Vehicle.Size     Vehicle.Style
## 7661                              N/A      Compact Passenger Minivan
## 7662                              N/A      Compact Passenger Minivan
## 7663                              N/A      Compact Passenger Minivan
## 3419                        Crossover        Large           4dr SUV
## 6610 Factory Tuner,Luxury,High-Performance      Midsize             Coupe
##      highway.MPG city.mpg Popularity  MSRP
## 7661          20       16       2031  2242
## 7662          19       15       2031  2728
## 7663          20       16       2031  2580
## 3419          27       21        586 37770
## 6610          26       17       3916 65700
```

Cook's distance is near zero for most cars, indicating that deleting any single one of these observations would not materially change the fitted model. In contrast, three Toyota Previa records (7661-7663), along with a Mazda CX-9 (3419) and a BMW M4 (6610), have much larger Cook's distances and are highly influential. Because these influential points are also associated with atypical MSRPs given their features, our substantive conclusions about how car characteristics relate to price should be checked for robustness to removing or down-weighting these specific observations.

```
# Simple baseline linear model

mod_base <- lm(
  MSRP ~ factor(Make) + factor(Year) + Engine.HP,
  data = dat_clean
)

summary(mod_base)
```

```
##
## Call:
## lm(formula = MSRP ~ factor(Make) + factor(Year) + Engine.HP,
##     data = dat_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -257999   -7461    -889    5869 1155786
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -3.322e+04  5.092e+03  -6.525 7.20e-11 ***
## factor(Make)Alfa Romeo     2.455e+04  1.302e+04   1.886 0.059326 .
## factor(Make)Aston Martin   1.123e+05  3.630e+03  30.924  < 2e-16 ***
## factor(Make)Audi           1.194e+04  2.445e+03   4.882 1.07e-06 ***
## factor(Make)Bentley        1.519e+05  3.987e+03  38.098  < 2e-16 ***
## factor(Make)BMW            7.993e+03  2.445e+03   3.269 0.001083 **
## factor(Make)Bugatti        1.567e+06  1.706e+04  91.856  < 2e-16 ***
## factor(Make)Buick         -1.525e+03  3.183e+03  -0.479 0.631854
## factor(Make)Cadillac       1.593e+03  2.358e+03   0.676 0.499313
## factor(Make)Chevrolet     -9.839e+03  2.193e+03  -4.488 7.31e-06 ***
## factor(Make)Chrysler      -6.902e+03  3.349e+03  -2.061 0.039327 *
## factor(Make)Dodge         -1.463e+04  2.482e+03  -5.894 3.92e-09 ***
## factor(Make)Ferrari        1.471e+05  4.132e+03  35.597  < 2e-16 ***
```

```
## factor(Make)FIAT           3.006e+03  5.059e+03   0.594 0.552380    
## factor(Make)Ford          -9.440e+03  2.259e+03  -4.178 2.97e-05 ***
## factor(Make)Genesis       -1.134e+04  1.674e+04  -0.677 0.498287    
## factor(Make)GMC           -7.056e+03  2.565e+03  -2.751 0.005961 ** 
## factor(Make)Honda         -6.228e+02  2.535e+03  -0.246 0.805917    
## factor(Make)HUMMER        -8.016e+03  7.331e+03  -1.093 0.274214    
## factor(Make)Hyundai       -5.977e+03  2.710e+03  -2.205 0.027463 *  
## factor(Make)Infiniti      -8.310e+03  2.450e+03  -3.392 0.000697 ***
## factor(Make)Kia           -5.863e+03  3.338e+03  -1.757 0.079030 .  
## factor(Make)Lamborghini    2.213e+05  4.635e+03  47.751  < 2e-16 ***
## factor(Make)Land Rover     1.518e+04  3.051e+03   4.975 6.65e-07 ***
## factor(Make)Lexus          4.387e+03  2.739e+03   1.602 0.109221    
## factor(Make)Lincoln       -2.367e+03  2.965e+03  -0.798 0.424688    
## factor(Make)Lotus          2.336e+04  5.696e+03   4.102 4.14e-05 ***
## factor(Make)Maserati       4.103e+04  4.270e+03   9.609  < 2e-16 ***
## factor(Make)Maybach        4.383e+05  7.625e+03  57.485  < 2e-16 ***
## factor(Make)Mazda         -4.512e+02  2.586e+03  -0.174 0.861524    
## factor(Make)McLaren        1.308e+05  1.309e+04   9.995  < 2e-16 ***
## factor(Make)Mercedes-Benz  1.762e+04  2.454e+03   7.180 7.60e-13 ***
## factor(Make)Mitsubishi     3.387e+02  3.168e+03   0.107 0.914846    
## factor(Make)Nissan        -7.046e+03  2.482e+03  -2.839 0.004537 ** 
## factor(Make)Oldsmobile    -5.695e+03  1.683e+04  -0.338 0.735081    
## factor(Make)Plymouth       9.650e+03  5.191e+03   1.859 0.063083 .  
## factor(Make)Pontiac       -1.315e+04  3.913e+03  -3.361 0.000779 ***
## factor(Make)Porsche        3.698e+04  3.139e+03  11.781  < 2e-16 ***
## factor(Make)Rolls-Royce    2.660e+05  5.577e+03  47.700  < 2e-16 ***
## factor(Make)Saab          -1.039e+03  3.428e+03  -0.303 0.761829    
## factor(Make)Scion         -5.584e+02  4.580e+03  -0.122 0.902960    
## factor(Make)Spyker         1.411e+05  1.683e+04   8.386  < 2e-16 ***
## factor(Make)Subaru        -2.478e+03  2.716e+03  -0.912 0.361605    
## factor(Make)Suzuki        -6.086e+03  3.508e+03  -1.735 0.082772 .  
## factor(Make)Toyota        -2.603e+03  2.488e+03  -1.046 0.295560    
## factor(Make)Volkswagen     2.188e+02  2.220e+03   0.099 0.921482    
## factor(Make)Volvo          8.684e+02  2.542e+03   0.342 0.732600    
## factor(Year)1991          -1.187e+03  6.004e+03  -0.198 0.843292    
## factor(Year)1992           3.769e+03  5.675e+03   0.664 0.506630    
## factor(Year)1993          -2.417e+03  5.589e+03  -0.432 0.665421    
## factor(Year)1994           1.030e+03  6.009e+03   0.171 0.863880    
## factor(Year)1995          -3.205e+03  6.036e+03  -0.531 0.595475    
## factor(Year)1996          -3.315e+02  6.111e+03  -0.054 0.956733    
## factor(Year)1997           3.998e+01  6.050e+03   0.007 0.994728    
## factor(Year)1998          -4.843e+03  6.960e+03  -0.696 0.486517    
## factor(Year)1999          -2.186e+03  6.410e+03  -0.341 0.733111    
## factor(Year)2000          -1.077e+03  6.072e+03  -0.177 0.859190    
## factor(Year)2001           2.736e+04  5.950e+03   4.599 4.32e-06 ***
## factor(Year)2002           2.048e+04  5.861e+03   3.495 0.000476 ***
## factor(Year)2003           2.832e+04  5.661e+03   5.002 5.79e-07 ***
## factor(Year)2004           2.497e+04  5.631e+03   4.434 9.36e-06 ***
## factor(Year)2005           3.160e+04  5.624e+03   5.619 1.98e-08 ***
## factor(Year)2006           2.453e+04  5.502e+03   4.458 8.37e-06 ***
## factor(Year)2007           2.107e+04  5.138e+03   4.101 4.15e-05 ***
## factor(Year)2008           3.029e+04  5.182e+03   5.844 5.28e-09 ***
## factor(Year)2009           2.796e+04  5.061e+03   5.524 3.41e-08 ***
## factor(Year)2010           2.562e+04  5.123e+03   5.001 5.83e-07 ***
```

```
## factor(Year)2011          2.877e+04  5.125e+03   5.613 2.05e-08 ***
## factor(Year)2012          2.625e+04  5.044e+03   5.203 2.01e-07 ***
## factor(Year)2013          2.078e+04  5.035e+03   4.128 3.70e-05 ***
## factor(Year)2014          2.287e+04  4.935e+03   4.634 3.64e-06 ***
## factor(Year)2015          2.458e+04  4.799e+03   5.123 3.08e-07 ***
## factor(Year)2016          2.411e+04  4.797e+03   5.026 5.13e-07 ***
## factor(Year)2017          2.394e+04  4.818e+03   4.968 6.89e-07 ***
## Engine.HP                 1.936e+02  4.007e+00  48.308  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28790 on 8036 degrees of freedom
##   (58 observations deleted due to missingness)
## Multiple R-squared:  0.8335, Adjusted R-squared:  0.832
## F-statistic: 543.7 on 74 and 8036 DF,  p-value: < 2.2e-16
```

We use Make, Year, and Engine.HP to form a simple baseline model because they capture the main structural drivers of MSRP: brand positioning, model-year effects, and basic performance. This small set keeps the model interpretable and avoids the long, crowded output that comes from including many correlated or highly categorical variables.

The baseline model explains a large share of MSRP variation (adjusted $R^2 = 0.83$). Horsepower has a strong positive effect, and price differences across brands and years follow expected patterns—luxury makes and newer model years are consistently more expensive. Overall, the model gives a clear, interpretable first look at how brand, year, and performance relate to price.

Future Steps

Going forward, we can build on both the baseline and full models in several ways. First, the diagnostic results (e.g., high-leverage FIAT/BMW coupes, large-residual Toyota Previa entries, and highly influential points such as the Mazda CX-9 and BMW M4) suggest that robustness checks—including refitting models with influential observations removed—would help assess the stability of our conclusions. Second, because several predictors are highly imbalanced (e.g., Market.Category, Transmission.Type), it may be beneficial to collapse rare categories or explore regularization methods (ridge/LASSO) to prevent overfitting in the full model. Third, interactions among performance and body-style variables or nonlinear relationships (e.g., log-transformed MSRP) could reveal richer structure not captured by main effects alone. Finally, comparing predictive performance across alternative models through cross-validation would help determine whether the simpler baseline model is adequate or whether the expanded feature set substantially improves accuracy.