Midterm - exploratory

```r
# step 0. Lets clean quickly
df <- read.csv('../midterm/data/vehicle_price_prediction.csv')

str(df)
```

```
## 'data.frame':    1000000 obs. of  20 variables:
##  $ make            : chr  "Volkswagen" "Lexus" "Subaru" "Cadillac" ...
##  $ model           : chr  "Jetta" "RX" "Crosstrek" "Lyriq" ...
##  $ year            : int  2016 2010 2016 2016 2018 2021 2018 2022 2015 2025 ...
##  $ mileage         : int  183903 236643 103199 118889 204170 44907 129770 29146 169165 5081 ...
##  $ engine_hp       : int  173 352 188 338 196 479 160 172 197 198 ...
##  $ transmission    : chr  "Manual" "Manual" "Automatic" "Manual" ...
##  $ fuel_type       : chr  "Electric" "Gasoline" "Diesel" "Gasoline" ...
##  $ drivetrain      : chr  "RWD" "FWD" "AWD" "AWD" ...
##  $ body_type       : chr  "Sedan" "Sedan" "Sedan" "SUV" ...
##  $ exterior_color  : chr  "Blue" "Silver" "Silver" "Black" ...
##  $ interior_color  : chr  "Brown" "Beige" "Beige" "Gray" ...
##  $ owner_count     : int  5 5 5 3 5 5 1 5 3 1 ...
##  $ accident_history: chr  "None" "Minor" "None" "None" ...
##  $ seller_type     : chr  "Dealer" "Dealer" "Dealer" "Private" ...
##  $ condition       : chr  "Excellent" "Good" "Excellent" "Good" ...
##  $ trim            : chr  "EX" "LX" "Touring" "Base" ...
##  $ vehicle_age     : int  9 15 9 9 7 4 7 3 10 1 ...
##  $ mileage_per_year: num  20434 15776 11467 13210 29167 ...
##  $ brand_popularity: num  0.0401 0.0399 0.0402 0.0398 0.0396 ...
##  $ price           : num  7209 6912 11916 25985 8151 ...
```

```r
summary(df)
```

```
##      make               model                year         mileage
##  Length:1000000     Length:1000000     Min.   :2000   Min.   :   500
##  Class :character   Class :character   1st Qu.:2015   1st Qu.: 57654
##  Mode  :character   Mode  :character   Median :2018   Median :103331
##                                        Mean   :2017   Mean   :112660
##                                        3rd Qu.:2020   3rd Qu.:157865
##                                        Max.   :2025   Max.   :300000
##    engine_hp     transmission        fuel_type          drivetrain
##  Min.   : 90.0   Length:1000000     Length:1000000     Length:1000000
##  1st Qu.:162.0   Class :character   Class :character   Class :character
##  Median :215.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :235.7
##  3rd Qu.:300.0
##  Max.   :581.0
##   body_type         exterior_color     interior_color      owner_count
##  Length:1000000     Length:1000000     Length:1000000     Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:2.000
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
```

```
##                                                          Mean   :3.478
##                                                          3rd Qu.:5.000
##                                                          Max.   :5.000
##  accident_history   seller_type         condition            trim
##  Length:1000000     Length:1000000     Length:1000000     Length:1000000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   vehicle_age     mileage_per_year   brand_popularity      price
##  Min.   : 1.000   Min.   :   33.33   Min.   :0.03932   Min.   : 1500
##  1st Qu.: 5.000   1st Qu.:10487.63   1st Qu.:0.03984   1st Qu.:10325
##  Median : 7.000   Median :14688.12   Median :0.04002   Median :17865
##  Mean   : 7.586   Mean   :14540.67   Mean   :0.04000   Mean   :20329
##  3rd Qu.:10.000   3rd Qu.:18685.94   3rd Qu.:0.04015   3rd Qu.:27601
##  Max.   :25.000   Max.   :55391.00   Max.   :0.04048   Max.   :93422
```

```
colSums(is.na(df)) # no nulls, good to go
```

```
##             make            model             year          mileage
##                0                0                0                0
##        engine_hp     transmission        fuel_type       drivetrain
##                0                0                0                0
##        body_type   exterior_color   interior_color      owner_count
##                0                0                0                0
## accident_history      seller_type        condition             trim
##                0                0                0                0
##      vehicle_age mileage_per_year brand_popularity            price
##                0                0                0                0
```

```
# step 1. Lets sample from this large

# Set a seed for reproducibility
set.seed(42)

# Take a random sample of up to 10,000 rows
df_sample <- df[sample(nrow(df), size = min(10000, nrow(df))), ]
head(df_sample)
```

```
##           make      model year mileage engine_hp transmission fuel_type
## 61413    Lexus         IS 2024    1233       276    Automatic  Electric
## 54425    Acura        TLX 2019  128520       219       Manual  Gasoline
## 623844     GMC    Terrain 2016   67053       228    Automatic  Electric
## 74362  Porsche      Macan 2016   76241       463    Automatic  Electric
## 46208   Toyota Highlander 2019   12399       170       Manual  Electric
## 964632 Porsche   Panamera 2022   46563       450       Manual  Gasoline
##        drivetrain    body_type exterior_color interior_color owner_count
## 61413         RWD        Sedan           Blue          Black           1
## 54425         FWD        Sedan           Gray          Beige           2
## 623844        RWD Pickup Truck          White          Brown           3
## 74362         AWD          SUV            Red          Brown           5
## 46208         FWD        Sedan          Black          Black           3
## 964632        FWD        Coupe            Red          Brown           1
##        accident_history seller_type condition    trim vehicle_age
```

```
## 61413              None      Dealer Excellent      LX           1
## 54425              None      Dealer      Good   Sport           6
## 623844            Minor      Dealer      Good      LX           9
## 74362              None      Dealer      Good      EX           9
## 46208              None     Private Excellent      LX           6
## 964632             None     Private      Fair Limited           3
##        mileage_per_year brand_popularity    price
## 61413          1233.000         0.039921 49945.88
## 54425         21420.000         0.040147 24459.16
## 623844         7450.333         0.039754 17670.10
## 74362          8471.222         0.040210 40183.55
## 46208          2066.500         0.039627 20426.55
## 964632        15521.000         0.040210 54966.06
```

```
# step 2. building up the linear model
# data transformation -- brand popularity scaled as the values range from 0.39 to 0.40
df_sample$brand_popularity_std <- scale(df_sample$brand_popularity)
df_sample <- transform(
  df_sample,
  mileage_k = (mileage - mean(mileage))/1000,          # center & put in thousands
  age_c     = scale(vehicle_age, center=TRUE, scale=FALSE)
)
head(df_sample)
```

```
##           make      model year mileage engine_hp transmission fuel_type
## 61413    Lexus         IS 2024    1233       276    Automatic  Electric
## 54425    Acura        TLX 2019  128520       219       Manual  Gasoline
## 623844     GMC    Terrain 2016   67053       228    Automatic  Electric
## 74362  Porsche      Macan 2016   76241       463    Automatic  Electric
## 46208   Toyota Highlander 2019   12399       170       Manual  Electric
## 964632 Porsche   Panamera 2022   46563       450       Manual  Gasoline
##        drivetrain    body_type exterior_color interior_color owner_count
## 61413         RWD        Sedan           Blue          Black           1
## 54425         FWD        Sedan           Gray          Beige           2
## 623844        RWD Pickup Truck          White          Brown           3
## 74362         AWD          SUV            Red          Brown           5
## 46208         FWD        Sedan          Black          Black           3
## 964632        FWD        Coupe            Red          Brown           1
##        accident_history seller_type condition    trim vehicle_age
## 61413              None      Dealer Excellent      LX           1
## 54425              None      Dealer      Good   Sport           6
## 623844            Minor      Dealer      Good      LX           9
## 74362              None      Dealer      Good      EX           9
## 46208              None     Private Excellent      LX           6
## 964632             None     Private      Fair Limited           3
##        mileage_per_year brand_popularity    price brand_popularity_std
## 61413          1233.000         0.039921 49945.88          -0.3433388
## 54425         21420.000         0.040147 24459.16           0.6280467
## 623844         7450.333         0.039754 17670.10          -1.0611325
## 74362          8471.222         0.040210 40183.55           0.8988312
## 46208          2066.500         0.039627 20426.55          -1.6069996
## 964632        15521.000         0.040210 54966.06           0.8988312
##         mileage_k   age_c
## 61413  -111.74267 -6.6356
## 54425    15.54433 -1.6356
```

```
## 623844  -45.92267  1.3644
## 74362   -36.73467  1.3644
## 46208  -100.57667 -1.6356
## 964632  -66.41267 -4.6356
```

```r
# Step 1. Variable selection using forward steps

# Install if needed
# install.packages(c("leaps", "broom"))

library(leaps)

cont_vars <- c(
  "year", "mileage", "engine_hp", "owner_count",
  "vehicle_age", "mileage_per_year", "brand_popularity"
)
fmla <- as.formula(paste("log(price) ~", paste(cont_vars, collapse = " + ")))

# Run exhaustive subset search
regfit <- regsubsets(
  fmla,
  data = df_sample,
  nbest = 1,
  nvmax = length(cont_vars),
  method = "exhaustive"
)

reg_summary <- summary(regfit)
reg_summary
```

```
## Subset selection object
## Call: regsubsets.formula(fmla, data = df_sample, nbest = 1, nvmax = length(cont_vars),
##     method = "exhaustive")
## 7 Variables  (and intercept)
##                  Forced in Forced out
## year                 FALSE      FALSE
## mileage              FALSE      FALSE
## engine_hp            FALSE      FALSE
## owner_count          FALSE      FALSE
## vehicle_age          FALSE      FALSE
## mileage_per_year     FALSE      FALSE
## brand_popularity     FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##          year mileage engine_hp owner_count vehicle_age mileage_per_year
## 1  ( 1 ) " "  "*"     " "       " "         " "         " "
## 2  ( 1 ) " "  "*"     "*"       " "         " "         " "
## 3  ( 1 ) " "  "*"     "*"       " "         "*"         " "
## 4  ( 1 ) " "  "*"     "*"       " "         "*"         "*"
## 5  ( 1 ) " "  "*"     "*"       "*"         "*"         "*"
## 6  ( 1 ) " "  "*"     "*"       "*"         "*"         "*"
## 7  ( 1 ) "*"  "*"     "*"       "*"         "*"         "*"
##          brand_popularity
## 1  ( 1 ) " "
## 2  ( 1 ) " "
```

```
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) " "
## 6  ( 1 ) "*"
## 7  ( 1 ) "*"
```

```r
### Identified the models below via variable importance

# Model 1: mileage
m1 <- lm(log(price) ~ mileage, data = df_sample)

# Model 2: mileage + engine_hp
m2 <- lm(log(price) ~ mileage + engine_hp, data = df_sample)

# Model 3: mileage + engine_hp + vehicle_age
m3 <- lm(log(price) ~ mileage + engine_hp + vehicle_age, data = df_sample)

# Model 4: mileage + engine_hp + vehicle_age + mileage_per_year
m4 <- lm(log(price) ~ mileage + engine_hp + vehicle_age + mileage_per_year, data = df_sample)

# Model 5: mileage + engine_hp + owner_count + vehicle_age + mileage_per_year
m5 <- lm(log(price) ~ mileage + engine_hp + owner_count + vehicle_age + mileage_per_year, data = df_samp

# Model 6: mileage + engine_hp + owner_count + vehicle_age + mileage_per_year + brand_popularity
m6 <- lm(log(price) ~ mileage + engine_hp + owner_count + vehicle_age + mileage_per_year + brand_popula

# Model 7: year + mileage + engine_hp + owner_count + vehicle_age + mileage_per_year + brand_popularity
m7 <- lm(log(price) ~ year + mileage + engine_hp + owner_count + vehicle_age + mileage_per_year + brand_

# Names of the *new* variable added at each step
added_vars <- c(
  "mileage",                    # first variable
  "+ engine_hp",                # added second
  "+ vehicle_age",              # added third
  "+ mileage_per_year",         # added fourth
  "+ owner_count",              # added fifth
  "+ brand_popularity",         # added sixth
  "+ year"                      # added last
)

# Compute metrics
r2_values <- c(summary(m1)$r.squared,
               summary(m2)$r.squared,
               summary(m3)$r.squared,
               summary(m4)$r.squared,
               summary(m5)$r.squared,
               summary(m6)$r.squared,
               summary(m7)$r.squared)

adjr2_values <- c(summary(m1)$adj.r.squared,
                  summary(m2)$adj.r.squared,
                  summary(m3)$adj.r.squared,
                  summary(m4)$adj.r.squared,
                  summary(m5)$adj.r.squared,
```

```
                  summary(m6)$adj.r.squared,
                  summary(m7)$adj.r.squared)

aic_values <- c(AIC(m1), AIC(m2), AIC(m3), AIC(m4), AIC(m5), AIC(m6), AIC(m7))
bic_values <- c(BIC(m1), BIC(m2), BIC(m3), BIC(m4), BIC(m5), BIC(m6), BIC(m7))

# Combine into a data frame
results <- data.frame(
  Step = 1:7,
  Added_Variable = added_vars,
  R2 = round(r2_values, 4),
  Adj_R2 = round(adjr2_values, 4),
  AIC = round(aic_values, 2),
  BIC = round(bic_values, 2)
)

# Print neatly
print(results, row.names = FALSE)
```

```
##  Step     Added_Variable     R2 Adj_R2      AIC      BIC
##     1            mileage 0.5075 0.5075 18330.39 18352.02
##     2        + engine_hp 0.8223 0.8223  8136.83  8165.67
##     3      + vehicle_age 0.8807 0.8807  4155.73  4191.78
##     4 + mileage_per_year 0.8882 0.8881  3511.84  3555.10
##     5      + owner_count 0.8886 0.8885  3478.16  3528.63
##     6 + brand_popularity 0.8889 0.8888  3451.35  3509.03
##     7             + year 0.8891 0.8890  3437.76  3502.65
```

```
# Base continuous model
m_base <- lm(log(price) ~ mileage + engine_hp + vehicle_age, data = df_sample)
cat_vars <- c("transmission", "fuel_type", "drivetrain", "body_type",
              "accident_history", "seller_type", "condition", "trim")
df_sample[cat_vars] <- lapply(df_sample[cat_vars], factor)
# Initialize storage
results <- data.frame(Variable = character(),
                      F_value = numeric(),
                      df = integer(),
                      p_value = numeric(),
                      stringsAsFactors = FALSE)

# Loop through each categorical variable
for (v in cat_vars) {
  fmla <- as.formula(paste("log(price) ~ mileage + engine_hp + vehicle_age +", v))
  m_test <- lm(fmla, data = df_sample)
  a <- anova(m_base, m_test)

  # Extract the F-stat and p-value from the second row (the comparison)
  results <- rbind(results, data.frame(
    Variable = v,
    F_value = round(a$F[2], 3),
    df = a$Df[2],
    p_value = signif(a$`Pr(>F)`[2], 4)
  ))
}
```

```r
# Sort by p-value
# Round F-values and p-values neatly
results$F_value <- round(results$F_value, 3)
results$p_value <- formatC(results$p_value, digits = 7, format = "f")

# Sort again by p-value if needed
results <- results[order(as.numeric(results$p_value)), ]

print(results, row.names = FALSE)
```

```
##          Variable F_value df   p_value
##         body_type  12.327  6 0.0000000
##  accident_history 264.762  2 0.0000000
##       seller_type  63.280  1 0.0000000
##         condition  97.025  2 0.0000000
##      transmission   1.626  1 0.2022000
##         fuel_type   1.300  2 0.2726000
##              trim   0.768  5 0.5724000
##        drivetrain   0.025  2 0.9750000
```

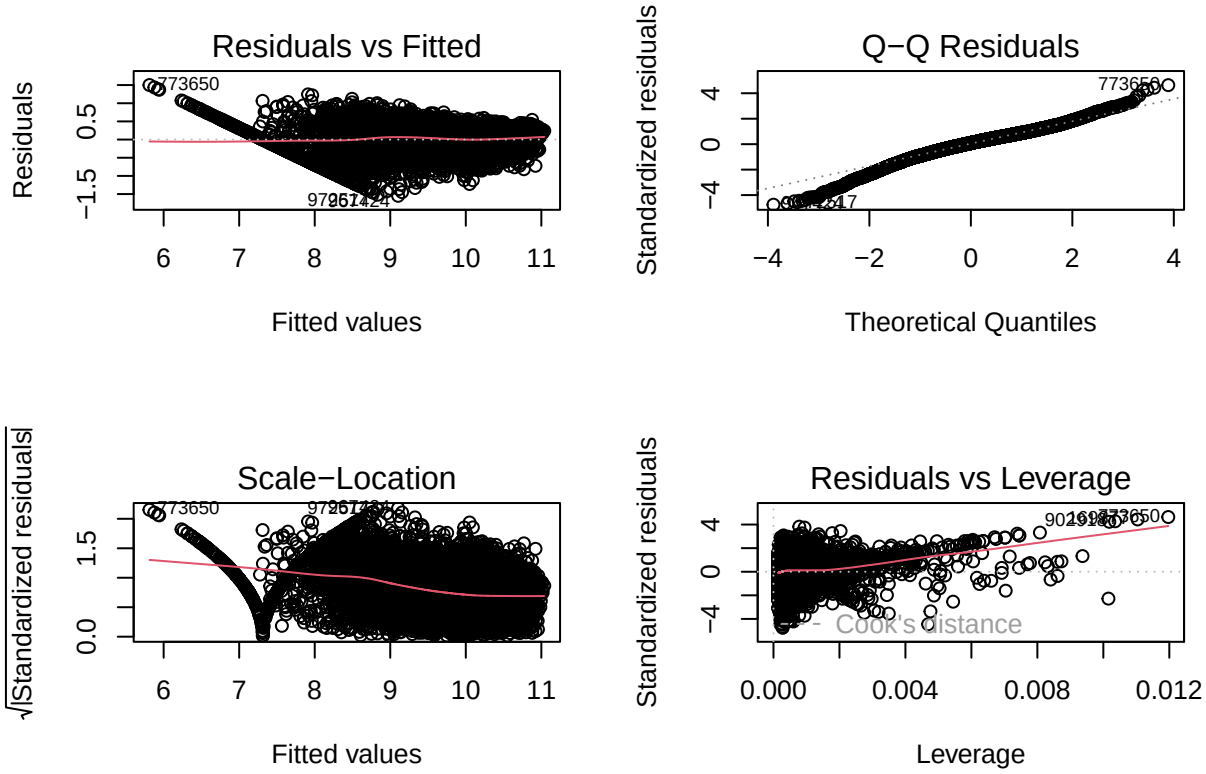
```r
model_3 <- lm(
  log(price) ~ mileage + I(mileage^2) + log(engine_hp) +
    mileage:log(engine_hp) + I(mileage^2):log(engine_hp),
  data = df_sample
)

summary(model_3)
```

```
##
## Call:
## lm(formula = log(price) ~ mileage + I(mileage^2) + log(engine_hp) +
##     mileage:log(engine_hp) + I(mileage^2):log(engine_hp), data = df_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.54765 -0.17964  0.02967  0.20600  1.50073
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.587e+00  1.194e-01  55.150  < 2e-16 ***
## mileage                     -2.585e-05  1.989e-06 -12.996  < 2e-16 ***
## I(mileage^2)                -3.327e-11  7.084e-12  -4.696 2.69e-06 ***
## log(engine_hp)               7.152e-01  2.209e-02  32.380  < 2e-16 ***
## mileage:log(engine_hp)       3.830e-06  3.680e-07  10.408  < 2e-16 ***
## I(mileage^2):log(engine_hp)  3.915e-12  1.312e-12   2.984  0.00285 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3256 on 9994 degrees of freedom
## Multiple R-squared:  0.8574, Adjusted R-squared:  0.8573
## F-statistic: 1.201e+04 on 5 and 9994 DF,  p-value: < 2.2e-16
```

```r
# assuming model name m_full or m_full_int
par(mfrow = c(2, 2))
```

```r
plot(model_3)   # residuals, QQ, scale-location, Cook's D
```

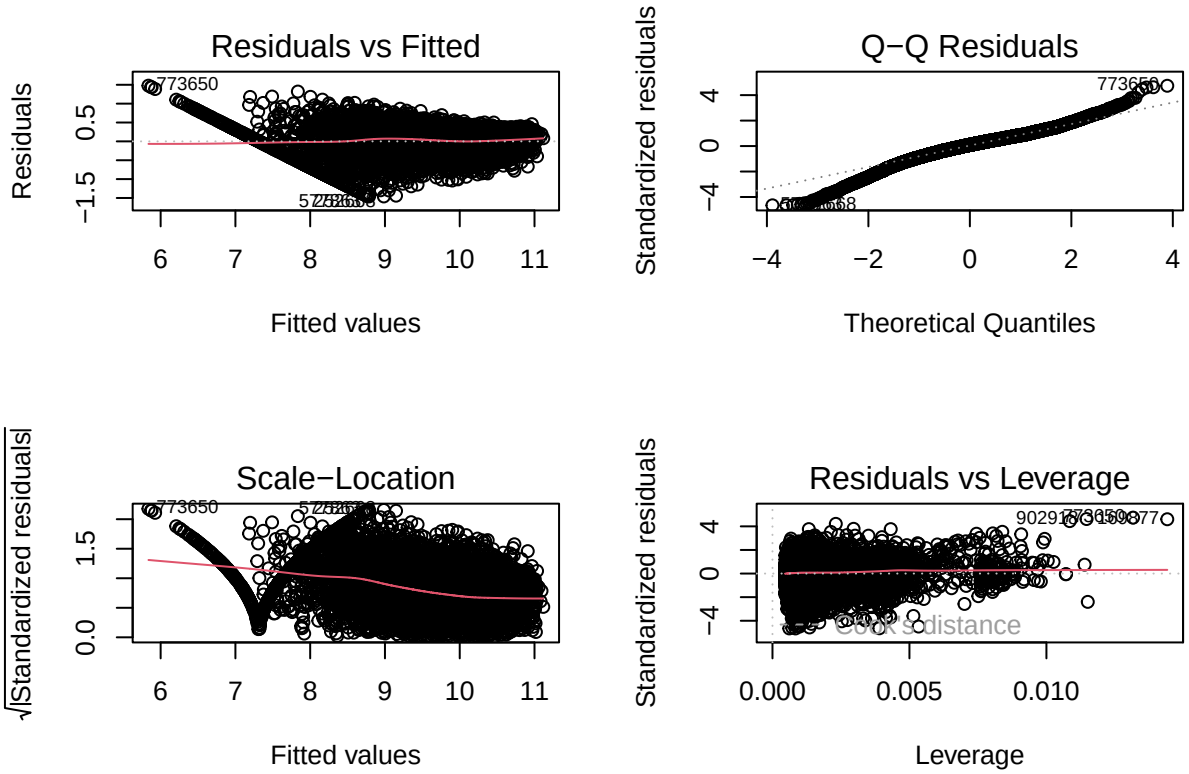



```r
par(mfrow = c(1, 1))
```

```r
model_4 <- lm(
  log(price) ~
    # Continuous + interaction structure
    mileage + I(mileage^2) + log(engine_hp) +
    mileage:log(engine_hp) + I(mileage^2):log(engine_hp) +

    # Significant categorical predictors
    factor(accident_history) + factor(condition) + factor(seller_type) + factor(body_type),
  data = df_sample
)

# Diagnostic plots
par(mfrow = c(2, 2))
plot(model_4)
```




```r
par(mfrow = c(1, 1))

# Model summary
summary(model_4)
```

```
##
## Call:
## lm(formula = log(price) ~ mileage + I(mileage^2) + log(engine_hp) +
##     mileage:log(engine_hp) + I(mileage^2):log(engine_hp) + factor(accident_history) +
##     factor(condition) + factor(seller_type) + factor(body_type),
##     data = df_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45936 -0.16558  0.02948  0.19163  1.47614
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      6.522e+00  1.182e-01  55.201  < 2e-16 ***
## mileage                         -2.649e-05  1.918e-06 -13.813  < 2e-16 ***
## I(mileage^2)                    -3.162e-11  6.830e-12  -4.630  3.7e-06 ***
## log(engine_hp)                   6.965e-01  2.143e-02  32.508  < 2e-16 ***
## factor(accident_history)Minor    1.929e-01  1.579e-02  12.216  < 2e-16 ***
## factor(accident_history)None     2.856e-01  1.463e-02  19.523  < 2e-16 ***
## factor(condition)Fair            -1.481e-01  1.111e-02 -13.334  < 2e-16 ***
## factor(condition)Good            -4.645e-02  6.661e-03  -6.973  3.3e-12 ***
## factor(seller_type)Private       -6.094e-02  6.800e-03  -8.962  < 2e-16 ***
## factor(body_type)Hatchback       -8.505e-02  2.203e-02  -3.860 0.000114 ***
## factor(body_type)Minivan         -5.685e-02  2.020e-02  -2.815 0.004894 **
```

```
## factor(body_type)Pickup Truck -4.831e-02  1.483e-02  -3.259 0.001124 **
## factor(body_type)Sedan        -4.095e-02  1.228e-02  -3.334 0.000860 ***
## factor(body_type)SUV          -2.171e-02  1.196e-02  -1.815 0.069572 .
## factor(body_type)Wagon         6.918e-02  2.903e-02   2.383 0.017195 *
## mileage:log(engine_hp)         3.946e-06  3.548e-07  11.124  < 2e-16 ***
## I(mileage^2):log(engine_hp)    3.630e-12  1.265e-12   2.870 0.004115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3138 on 9983 degrees of freedom
## Multiple R-squared:  0.8677, Adjusted R-squared:  0.8675
## F-statistic:  4092 on 16 and 9983 DF,  p-value: < 2.2e-16
```