Car Engine Displacement & Horsepower Analysis

Emerson Fleming

9/10/2022

Step 1: The Cleaning of the Data!

```
Cars_Data <- read_csv("./EFleming-train-data-used-cars - train-data-used-cars (3).csv")
```

```
## Rows: 6019 Columns: 14

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): Name, Location, Fuel_Type, Transmission, Owner_Type, Mileage, Engin...
## dbl (4): Year, Kilometers_Driven, Seats, Price in $
## lgl (2): Make, Model

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Cars_Data$Make <- word(Cars_Data$Name, 1)
Cars_Data$Model <- word(Cars_Data$Name, 2)
Cars_Data <- rename(Cars_Data, "Price" = "Price in $")
#Here, we are simply creating a new Make and Model column in the dataset to compare vehicles further. O
```

```
Cars_Data %>% summarise_all(funs(sum(is.na(.)))) #NAs in Mileage, Engine, Power, and Seats. We want to
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))

## # A tibble: 1 x 14
##    Name  Make Model Location  Year Kilometers_Driven Fuel_Type Transmission
##   <int> <int> <int>    <int> <int>             <int>     <int>        <int>
```

```
## 1     0     0     0        0     0                 0         0            0
## # ... with 6 more variables: Owner_Type <int>, Mileage <int>, Engine <int>,
## #   Power <int>, Seats <int>, Price <int>
```

```
Imputed_Data <- mice(Cars_Data, m=5, method = "rf") #Used MICE for imputed data
```

```
##
##  iter imp variable
##   1   1  Seats
##   1   2  Seats
##   1   3  Seats
##   1   4  Seats
##   1   5  Seats
##   2   1  Seats
##   2   2  Seats
##   2   3  Seats
##   2   4  Seats
##   2   5  Seats
##   3   1  Seats
##   3   2  Seats
##   3   3  Seats
##   3   4  Seats
##   3   5  Seats
##   4   1  Seats
##   4   2  Seats
##   4   3  Seats
##   4   4  Seats
##   4   5  Seats
##   5   1  Seats
##   5   2  Seats
##   5   3  Seats
##   5   4  Seats
##   5   5  Seats

## Warning: Number of logged events: 10
```

```
Cars_Data_Imputed <- complete(Imputed_Data)
Cars_Data_Imputed <- na.omit(Cars_Data_Imputed) #Omitted the few variables MICE did not create a variab
Cars_Data_Imputed$Engine = as.numeric(sub("\\ .*", "", Cars_Data_Imputed$Engine))
Cars_Data_Imputed$Mileage = as.numeric(sub("\\ .*", "", Cars_Data_Imputed$Mileage))
Cars_Data_Imputed$Power = as.numeric(sub("\\ .*", "", Cars_Data_Imputed$Power))
```

```
## Warning: NAs introduced by coercion
```

```
##Here we are taking off the the original "bhp (for "Power), kmpl (for "Mileage), and cc (for "Engine")
```

```
Cars_Data_Imputed$Make[Cars_Data_Imputed$Make == "ISUZU"] = "Isuzu"
Cars_Data_Imputed$Make[Cars_Data_Imputed$Make == "MiniCooper"] = "Mini"
unique(Cars_Data_Imputed$Make)
```

```
##  [1] "Maruti"        "Hyundai"       "Honda"         "Audi"
```

```
##  [5] "Nissan"        "Toyota"        "Volkswagen"    "Tata"
##  [9] "LandRover"     "Mitsubishi"    "Renault"       "Mercedes-Benz"
## [13] "BMW"           "Mahindra"      "Ford"          "Porsche"
## [17] "Datsun"        "Jaguar"        "Volvo"         "Chevrolet"
## [21] "Skoda"         "Mini"          "Fiat"          "Jeep"
## [25] "Smart"         "Ambassador"    "Isuzu"         "Force"
## [29] "Bentley"       "Lamborghini"
```

```
##Here we fix problems in the dataset. For instance, there are two occurrences of two makes which needed
```

```
sapply(Cars_Data_Imputed, function(x) sum(is.na(x))) #function to try and find NA values. We want to ma
```

```
##              Name              Make             Model          Location
##                 0                 0                 0                 0
##              Year Kilometers_Driven         Fuel_Type      Transmission
##                 0                 0                 0                 0
##        Owner_Type           Mileage            Engine             Power
##                 0                 0                 0               107
##             Seats             Price
##                 0                 0
```

```
summary(Cars_Data_Imputed) #Here, we take a peak at the data before building a model for multivariate a
```

```
##      Name               Make              Model             Location
##  Length:5981        Length:5981        Length:5981        Length:5981
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       Year      Kilometers_Driven  Fuel_Type         Transmission
##  Min.   :1998   Min.   :    171   Length:5981        Length:5981
##  1st Qu.:2011   1st Qu.:  33931   Class :character   Class :character
##  Median :2014   Median :  53000   Mode  :character   Mode  :character
##  Mean   :2013   Mean   :  58688
##  3rd Qu.:2016   3rd Qu.:  73000
##  Max.   :2019   Max.   :6500000
##
##   Owner_Type           Mileage          Engine         Power
##  Length:5981        Min.   : 0.00   Min.   : 624   Min.   : 34.2
##  Class :character   1st Qu.:15.20   1st Qu.:1198   1st Qu.: 75.0
##  Mode  :character   Median :18.16   Median :1493   Median : 97.7
##                     Mean   :18.18   Mean   :1622   Mean   :113.3
##                     3rd Qu.:21.10   3rd Qu.:1984   3rd Qu.:138.1
##                     Max.   :33.54   Max.   :5998   Max.   :560.0
##                                                    NA's   :107
##      Seats            Price
##  Min.   : 0.000   Min.   :  0.440
##  1st Qu.: 5.000   1st Qu.:  3.500
##  Median : 5.000   Median :  5.650
##  Mean   : 5.279   Mean   :  9.495
```

```
##  3rd Qu.: 5.000   3rd Qu.:  9.950
##  Max.   :10.000   Max.   :160.000
##
```

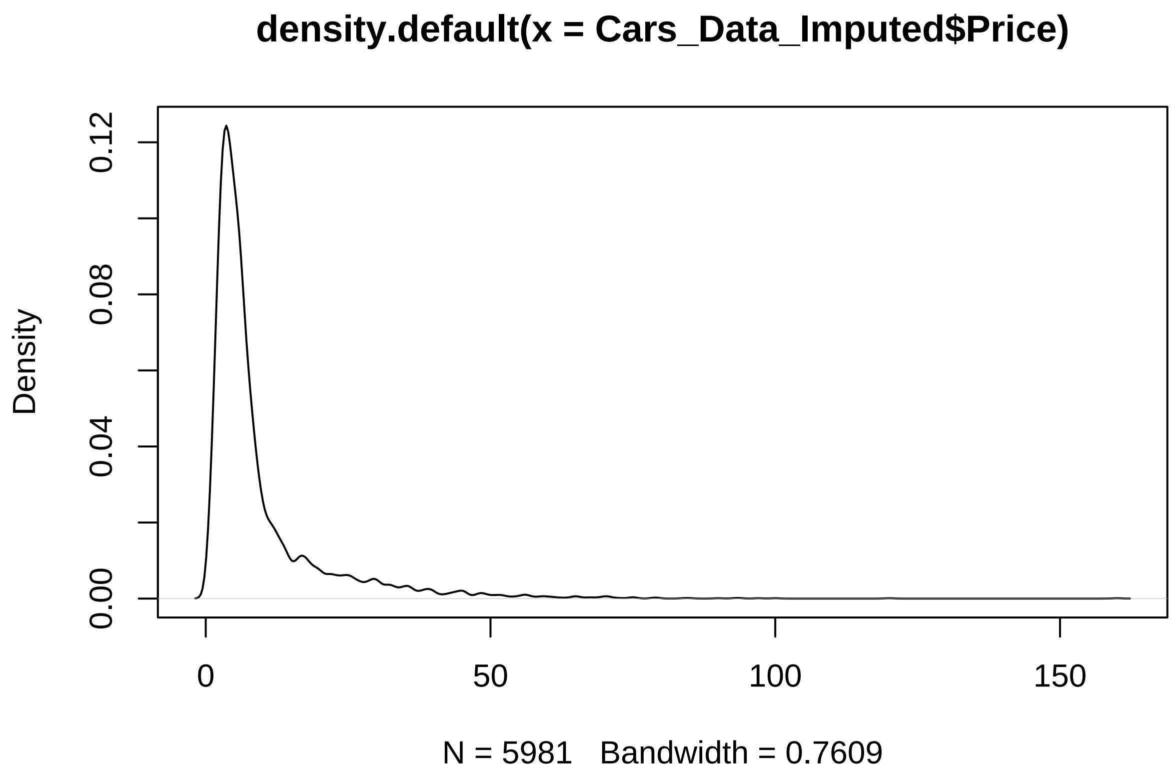
```r
Reg <- lm(Price~Engine + Mileage + Owner_Type + Seats + as.factor(Transmission) + Make + Year, data = Ca
summary(Reg)
```

```
##
## Call:
## lm(formula = Price ~ Engine + Mileage + Owner_Type + Seats +
##     as.factor(Transmission) + Make + Year, data = Cars_Data_Imputed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.701  -1.988  -0.282   1.561 113.075
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -2.245e+03  5.390e+01 -41.648  < 2e-16 ***
## Engine                        7.872e-03  2.258e-04  34.867  < 2e-16 ***
## Mileage                      -1.104e-01  2.289e-02  -4.821 1.46e-06 ***
## Owner_TypeFourth & Above      2.827e-01  1.984e+00   0.142 0.886705
## Owner_TypeSecond             -5.159e-01  2.089e-01  -2.469 0.013566 *
## Owner_TypeThird               9.279e-01  5.611e-01   1.654 0.098258 .
## Seats                        -4.069e-01  1.326e-01  -3.068 0.002164 **
## as.factor(Transmission)Manual -6.123e-01  2.360e-01  -2.595 0.009488 **
## MakeAudi                      6.864e+00  5.616e+00   1.222 0.221644
## MakeBentley                   1.825e+01  7.965e+00   2.291 0.021989 *
## MakeBMW                       7.097e+00  5.615e+00   1.264 0.206295
## MakeChevrolet                -5.230e+00  5.618e+00  -0.931 0.351910
## MakeDatsun                   -7.357e+00  5.811e+00  -1.266 0.205534
## MakeFiat                     -4.458e+00  5.706e+00  -0.781 0.434688
## MakeForce                    -7.563e+00  6.462e+00  -1.170 0.241905
## MakeFord                     -5.089e+00  5.606e+00  -0.908 0.364038
## MakeHonda                    -5.793e+00  5.602e+00  -1.034 0.301178
## MakeHyundai                  -4.724e+00  5.601e+00  -0.843 0.398990
## MakeIsuzu                    -1.093e+01  6.464e+00  -1.690 0.091043 .
## MakeJaguar                    1.516e+01  5.677e+00   2.671 0.007580 **
## MakeJeep                     -9.851e-01  5.785e+00  -0.170 0.864779
## MakeLamborghini               7.793e+01  7.947e+00   9.806  < 2e-16 ***
## MakeLandRover                 1.847e+01  5.654e+00   3.266 0.001095 **
## MakeMahindra                 -8.376e+00  5.614e+00  -1.492 0.135721
## MakeMaruti                   -3.904e+00  5.601e+00  -0.697 0.485824
## MakeMercedes-Benz             7.297e+00  5.613e+00   1.300 0.193639
## MakeMini                      1.073e+01  5.710e+00   1.880 0.060201 .
## MakeMitsubishi               -6.458e+00  5.699e+00  -1.133 0.257141
## MakeNissan                   -6.104e+00  5.629e+00  -1.084 0.278221
## MakePorsche                   2.002e+01  5.771e+00   3.468 0.000527 ***
## MakeRenault                  -5.583e+00  5.619e+00  -0.994 0.320415
## MakeSkoda                    -5.460e+00  5.615e+00  -0.972 0.330880
## MakeSmart                    -3.058e-01  7.921e+00  -0.039 0.969201
## MakeTata                     -6.477e+00  5.614e+00  -1.154 0.248651
## MakeToyota                   -5.790e+00  5.609e+00  -1.032 0.301945
## MakeVolkswagen               -5.949e+00  5.607e+00  -1.061 0.288719
```

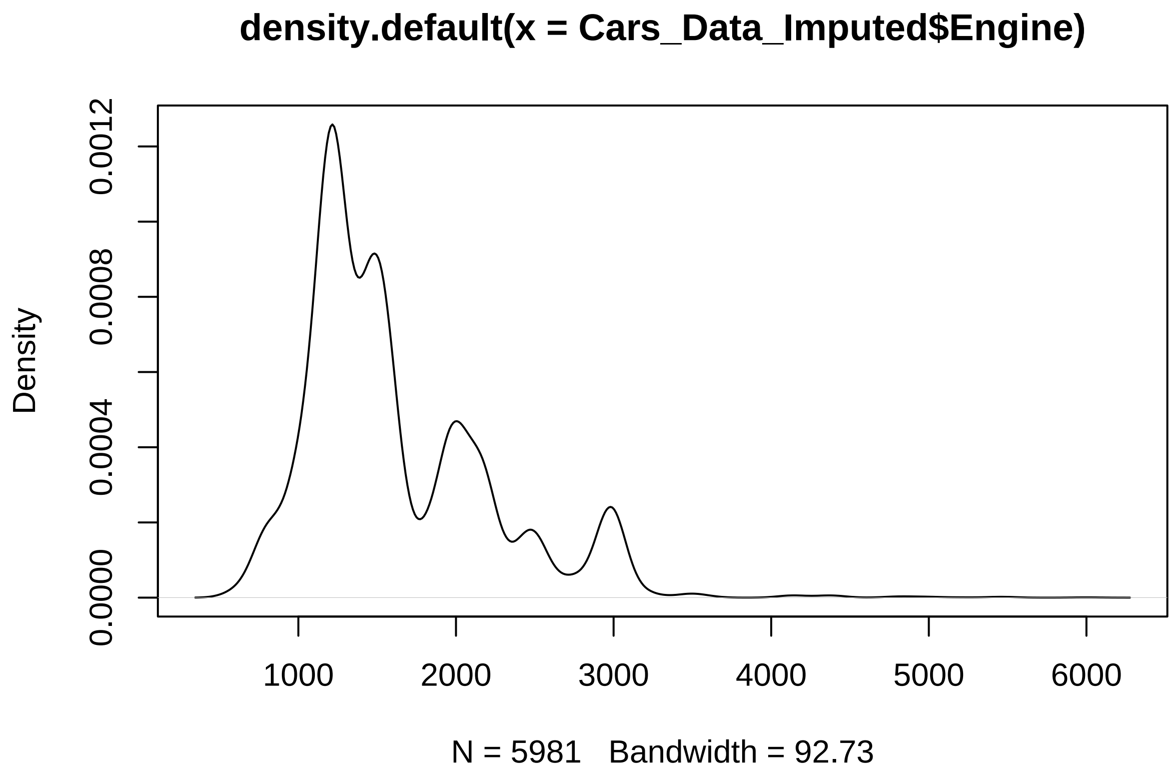
```
## MakeVolvo                         1.521e+00  5.734e+00   0.265 0.790863
## Year                              1.117e+00  2.676e-02  41.740  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.571 on 5943 degrees of freedom
## Multiple R-squared:  0.7542, Adjusted R-squared:  0.7527
## F-statistic: 492.8 on 37 and 5943 DF,  p-value: < 2.2e-16
```

```
#Here, we start to build a model to get an idea of what we would like the final model to look like and
```

```
plot(density(Cars_Data_Imputed$Price)) #We use a density plot in order to see if our data is misleading
```

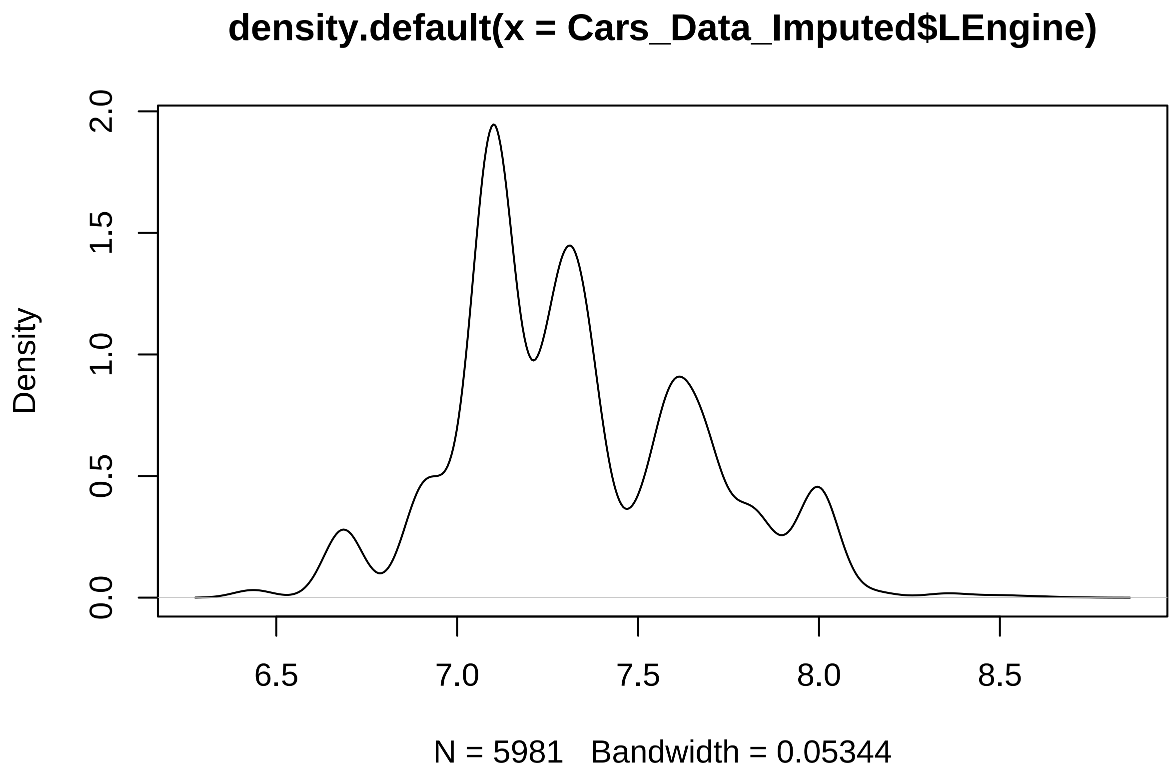



```
plot(density(Cars_Data_Imputed$Engine)) #This particular graph demonstrates the mass majority of the ca
```

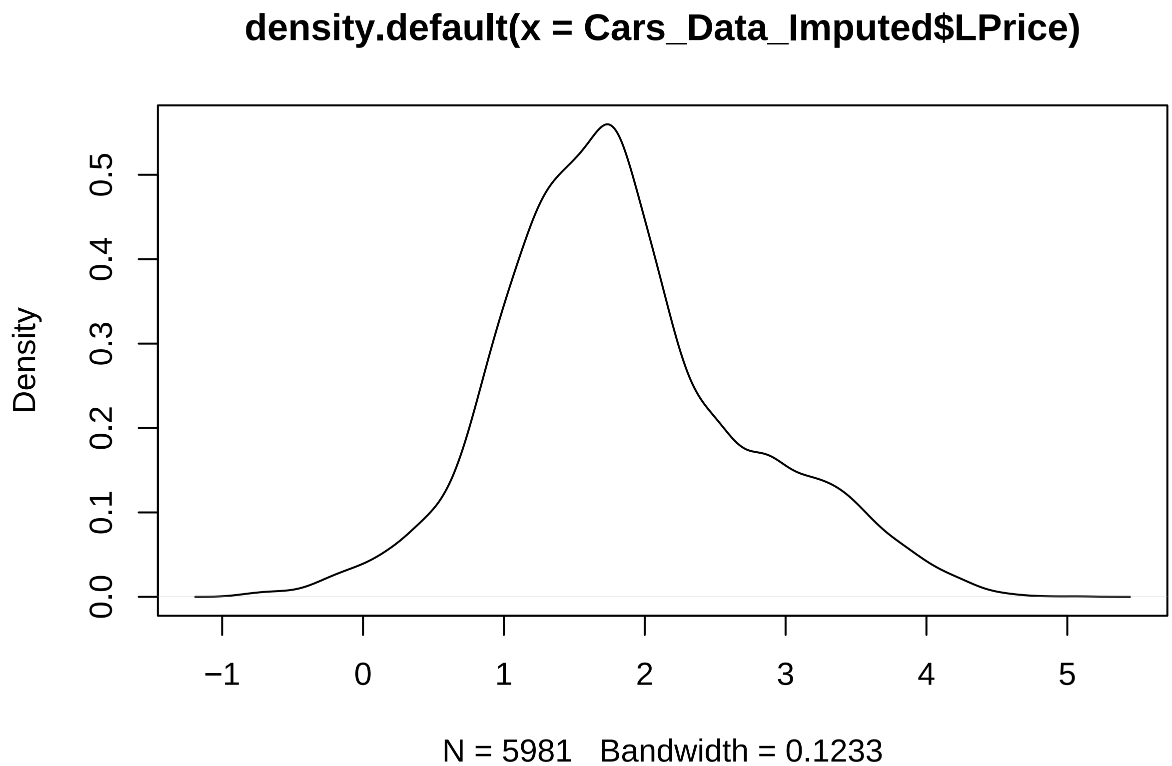



```
Cars_Data_Imputed$LPrice = log(Cars_Data_Imputed$Price) #Here, we create the logs themselves.
Cars_Data_Imputed$LEngine = log(Cars_Data_Imputed$Engine)
```

```
plot(density(Cars_Data_Imputed$LEngine)) #Here, we can observe a much better and more balanced result f
```

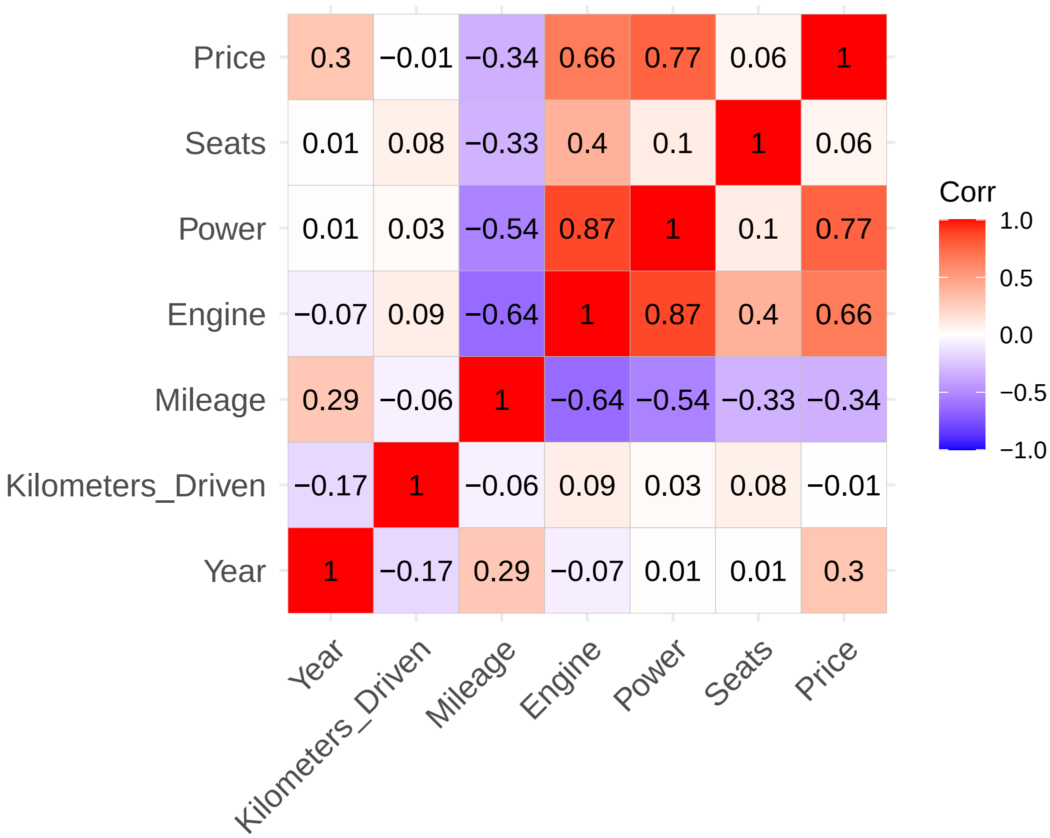



```
plot(density(Cars_Data_Imputed$LPrice)) #Again, we can observe a much better and more balanced result f
```



Step 2: Exploratory Analysis

```
Cars_Data_Imputed <- Cars_Data_Imputed %>% drop_na() #NAs were dropped in order to create this correlat
Cars_Data_Imputed_Cor <- Cars_Data_Imputed[, c("Year", "Kilometers_Driven", "Mileage", "Engine", "Power
Cor_Data_Test <- cor(Cars_Data_Imputed_Cor)
ggcorrplot(Cor_Data_Test, lab = TRUE) #The following plot visualizes upper correlation coefficients in
```



Step 3: Visualizations before model

*The seats visualization was not used as Marginal Error was used to compare models and assess results. Therefore, there was no accessible way to also fit in seats.

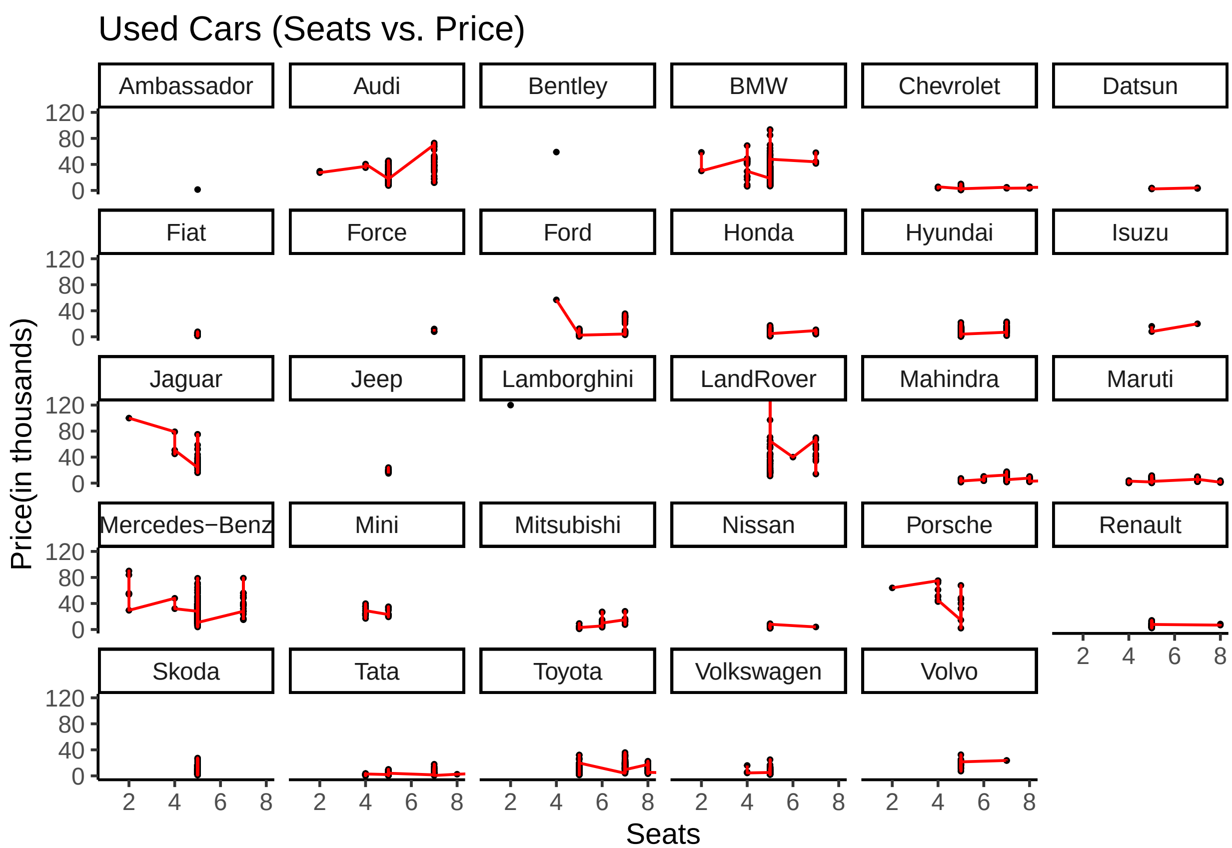
```
ggplot(Cars_Data_Imputed,
  aes(x = Seats, y = Price)) +
  coord_cartesian(xlim = c(1, 8), ylim = c(0, 120)) +
  geom_point(size = 0.5) +
  geom_line(colour = "red") +
  scale_y_continuous(breaks=seq(0, 120, by=40)) +
  geom_smooth() +
  facet_wrap(~Make) +
  labs(title = "Used Cars (Seats vs. Price)",
       x = "Seats",
       y = "Price(in thousands)")+
  theme_classic()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```

```
## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in ‘stat_smooth()‘:
## x has insufficient unique values to support 10 knots: reduce k.
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?

## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

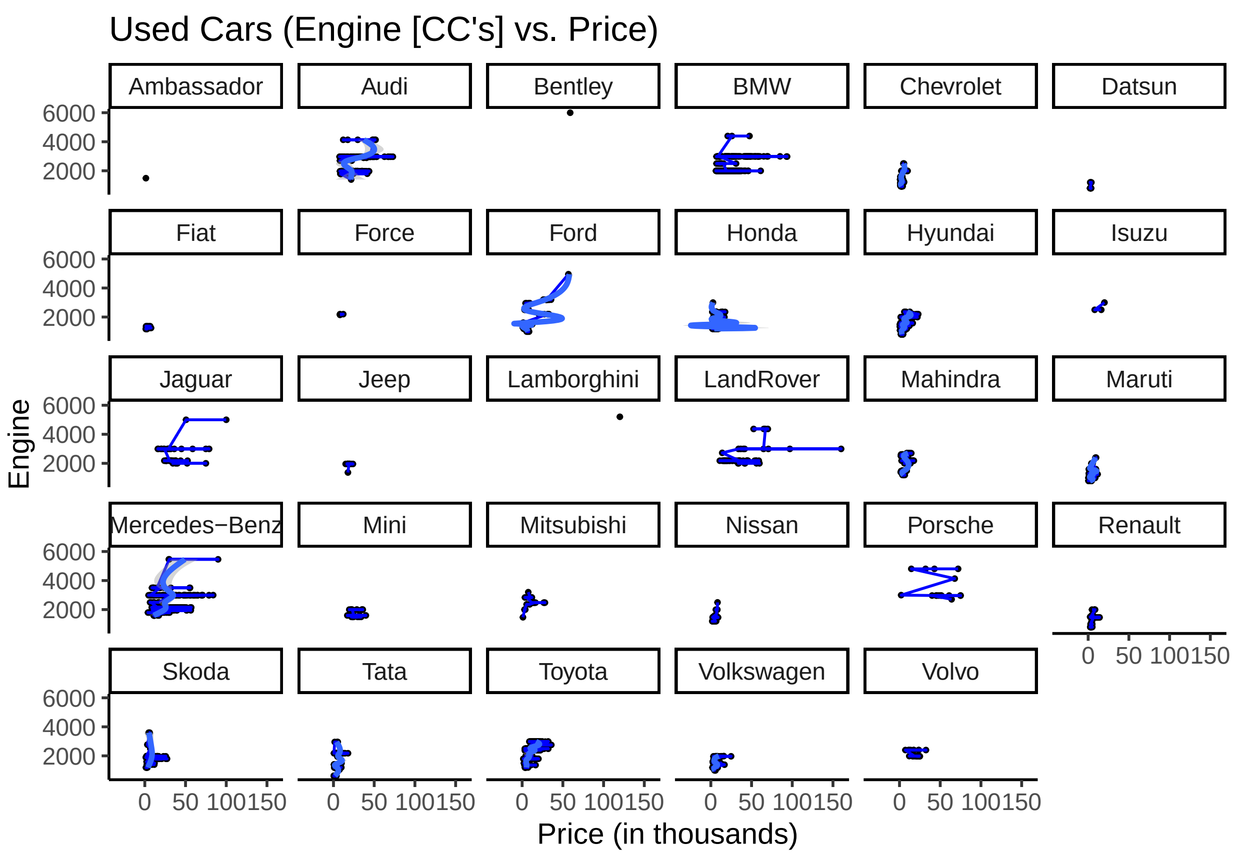



```r
ggplot(Cars_Data_Imputed,
  aes(x = Engine, y = Price)) +
  coord_cartesian(xlim = c(0, 5000), ylim = c(0, 120)) +
  geom_point(size = 0.5) +
  geom_line(colour = "blue") +
  geom_smooth() +
  coord_flip() +
  facet_wrap(~Make) +
```

```
  labs(title = "Used Cars (Engine [CC's] vs. Price)",
       x = "Engine",
       y = "Price (in thousands)")+
  theme_classic()
```

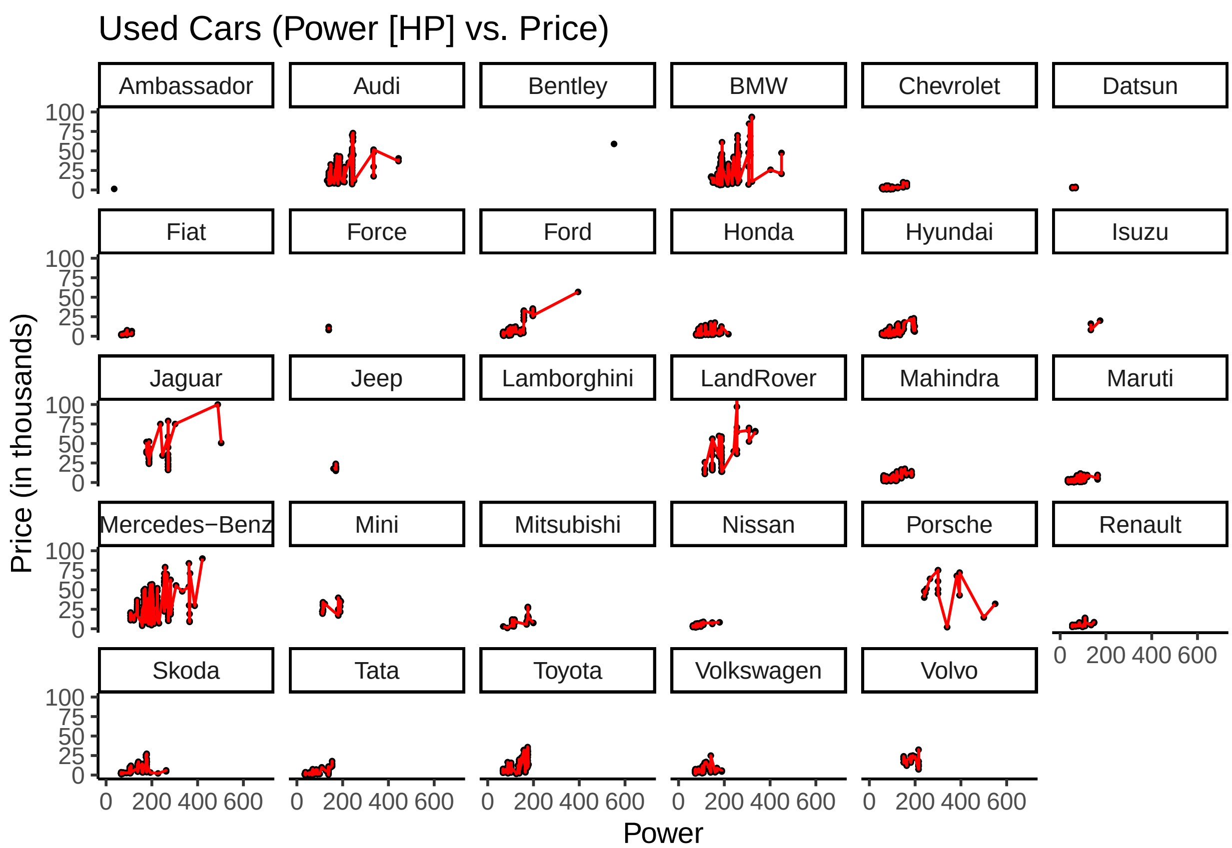
```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.

## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```




```r
ggplot(Cars_Data_Imputed,
  aes(x = Power, y = Price)) +
  coord_cartesian(xlim = c(0, 700), ylim = c(0, 100)) +
  geom_point(size = 0.5) +
  geom_line(colour = "red") +
  facet_wrap(~Make) +
  labs(title = "Used Cars (Power [HP] vs. Price)",
       x = "Power",
       y = "Price (in thousands)")+
  theme_classic()
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



Step 4: Modeling

Inclusive Cars Model

$lm(LPrice\ LEngine*Power+Owner_Type+Mileage+Year+Seats+Make+Kilometers_Driven, data = Cars_Data_Imputed)$

```
Reg_w_L <- lm(LPrice~LEngine * Power + Owner_Type + Mileage + Year + Seats + Make + Kilometers_Driven, 
summary(Reg_w_L)
```

```
##
## Call:
## lm(formula = LPrice ~ LEngine * Power + Owner_Type + Mileage +
##     Year + Seats + Make + Kilometers_Driven, data = Cars_Data_Imputed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8555 -0.1484  0.0091  0.1630  1.5010
##
## Coefficients:
##                              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                -2.618e+02  2.546e+00 -102.843  < 2e-16 ***
## LEngine                     9.846e-01  2.819e-02   34.930  < 2e-16 ***
## Power                       3.167e-02  1.504e-03   21.051  < 2e-16 ***
## Owner_TypeFourth & Above    3.465e-02  9.532e-02    0.363 0.716253
```

```
## Owner_TypeSecond         -5.760e-02  9.538e-03   -6.038 1.65e-09 ***
## Owner_TypeThird          -1.017e-01  2.639e-02   -3.854 0.000117 ***
## Mileage                   4.908e-03  1.096e-03    4.477 7.72e-06 ***
## Year                      1.270e-01  1.257e-03  101.039  < 2e-16 ***
## Seats                     6.605e-02  6.240e-03   10.585  < 2e-16 ***
## MakeAudi                  1.954e-01  2.535e-01    0.771 0.440889
## MakeBentley               1.130e+00  3.648e-01    3.098 0.001960 **
## MakeBMW                   1.185e-01  2.537e-01    0.467 0.640304
## MakeChevrolet            -8.184e-01  2.533e-01   -3.230 0.001243 **
## MakeDatsun               -9.108e-01  2.620e-01   -3.476 0.000513 ***
## MakeFiat                 -7.120e-01  2.578e-01   -2.762 0.005767 **
## MakeForce                -6.341e-01  2.913e-01   -2.177 0.029538 *
## MakeFord                 -5.492e-01  2.527e-01   -2.173 0.029795 *
## MakeHonda                -6.007e-01  2.527e-01   -2.377 0.017463 *
## MakeHyundai              -5.222e-01  2.526e-01   -2.067 0.038743 *
## MakeIsuzu                -6.813e-01  2.911e-01   -2.341 0.019291 *
## MakeJaguar                2.740e-01  2.564e-01    1.068 0.285437
## MakeJeep                 -3.598e-01  2.612e-01   -1.377 0.168475
## MakeLamborghini           1.302e+00  3.632e-01    3.585 0.000339 ***
## MakeLandRover             5.057e-01  2.550e-01    1.983 0.047378 *
## MakeMahindra             -7.006e-01  2.530e-01   -2.769 0.005641 **
## MakeMaruti               -4.728e-01  2.526e-01   -1.872 0.061273 .
## MakeMercedes-Benz         1.898e-01  2.533e-01    0.749 0.453672
## MakeMini                  5.249e-01  2.576e-01    2.038 0.041635 *
## MakeMitsubishi           -2.733e-01  2.567e-01   -1.065 0.287076
## MakeNissan               -5.654e-01  2.537e-01   -2.228 0.025913 *
## MakePorsche               1.800e-01  2.621e-01    0.687 0.492177
## MakeRenault              -4.868e-01  2.533e-01   -1.921 0.054739 .
## MakeSkoda                -4.732e-01  2.532e-01   -1.869 0.061713 .
## MakeTata                 -8.850e-01  2.531e-01   -3.497 0.000474 ***
## MakeToyota               -3.404e-01  2.527e-01   -1.347 0.178114
## MakeVolkswagen           -5.259e-01  2.528e-01   -2.080 0.037528 *
## MakeVolvo                 1.215e-02  2.589e-01    0.047 0.962580
## Kilometers_Driven        -1.194e-09  3.643e-08   -0.033 0.973862
## LEngine:Power            -3.407e-03  1.833e-04  -18.585  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2507 on 5835 degrees of freedom
## Multiple R-squared:  0.9163, Adjusted R-squared:  0.9158
## F-statistic:  1682 on 38 and 5835 DF,  p-value: < 2.2e-16
```

```
#Here, we can plug in our new logged "LPrice" and "LEngine" variables into our original model for a bet

#All variables are included at first and we implement backward elimination in order to create the best 

#The interesting thing to note is the interaction between Power and Engine. Power and Engine have a cor

#We also will take out Transmission as this only seems to be less inclusive when added. In other words,

#Adding Fuel_Type also seems to make the data less inclusive so no point in adding this variable either

#There does appear to be some correlation between location but not a lot and adding this variable fails
```

```
#Strangely, there is no correlation between LPrice and Kilometers_Driven so we will use Mileage (Gas Mi

#Ultimately, we end up with Statistical Significance across 16 makes.

#We see the highest correlation between LPrice and Year, Lprice and Power and LPrice and Seats. Therefo

#Kilometers Driven will serve as the control variable due to its very low (if any) statistical signific
```

Expensive Cars Model

*This uses the same model as the inclusive model but is modified to only include a dataset with expensive models.

```
expensive = c("Audi", "BMW", "Bentley", "Jaguar", "Lamborghini", "Porsche", "Mercedes-Benz", "LandRover
Cars_expensive = subset(Cars_Data_Imputed, Make %in% expensive)
Reg_w_L_expensive <- lm(LPrice~LEngine * Power + Owner_Type + Mileage + Year + Seats + Make + Kilometers
summary(Reg_w_L_expensive)
```

```
##
## Call:
## lm(formula = LPrice ~ LEngine * Power + Owner_Type + Mileage +
##     Year + Seats + Make + Kilometers_Driven, data = Cars_expensive)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8757 -0.1596  0.0055  0.1652  1.3643
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.051e+02  7.491e+00 -40.729  < 2e-16 ***
## LEngine            1.231e+00  1.560e-01   7.891 8.60e-15 ***
## Power              2.909e-02  5.015e-03   5.800 9.12e-09 ***
## Owner_TypeSecond  -4.710e-02  2.535e-02  -1.858 0.063529 .
## Owner_TypeThird   -1.406e-01  9.348e-02  -1.504 0.132958
## Mileage           -4.303e-03  3.067e-03  -1.403 0.160896
## Year               1.481e-01  3.684e-03  40.192  < 2e-16 ***
## Seats             -4.448e-03  1.562e-02  -0.285 0.775878
## MakeBentley        1.171e+00  3.258e-01   3.595 0.000342 ***
## MakeBMW           -5.181e-02  2.796e-02  -1.853 0.064219 .
## MakeJaguar         6.191e-02  5.083e-02   1.218 0.223527
## MakeLamborghini    1.173e+00  3.293e-01   3.562 0.000387 ***
## MakeLandRover      2.410e-01  4.462e-02   5.401 8.44e-08 ***
## MakeMercedes-Benz -4.329e-02  2.564e-02  -1.688 0.091728 .
## MakePorsche       -5.885e-03  8.124e-02  -0.072 0.942275
## Kilometers_Driven  5.739e-09  4.454e-08   0.129 0.897491
## LEngine:Power     -3.269e-03  6.225e-04  -5.251 1.89e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2882 on 911 degrees of freedom
## Multiple R-squared:  0.7553, Adjusted R-squared:  0.751
## F-statistic: 175.7 on 16 and 911 DF,  p-value: < 2.2e-16
```

```r
#The following model includes only 12 makes. However, what is particularly notable is that with just ta

#Kilometers Driven will serve as the control variable due to its very low (if any) statistical signific
```

Affordable Cars Model

*This uses the same model as the inclusive model but is modified to only include a dataset with affordable models.

```r
Cars_affordable= subset(Cars_Data_Imputed, !(Make %in% expensive))
Reg_w_L_affordable <- lm(LPrice~LEngine * Power + Owner_Type + Mileage + Year + Seats + Make + Kilometer
summary(Reg_w_L_affordable)
```

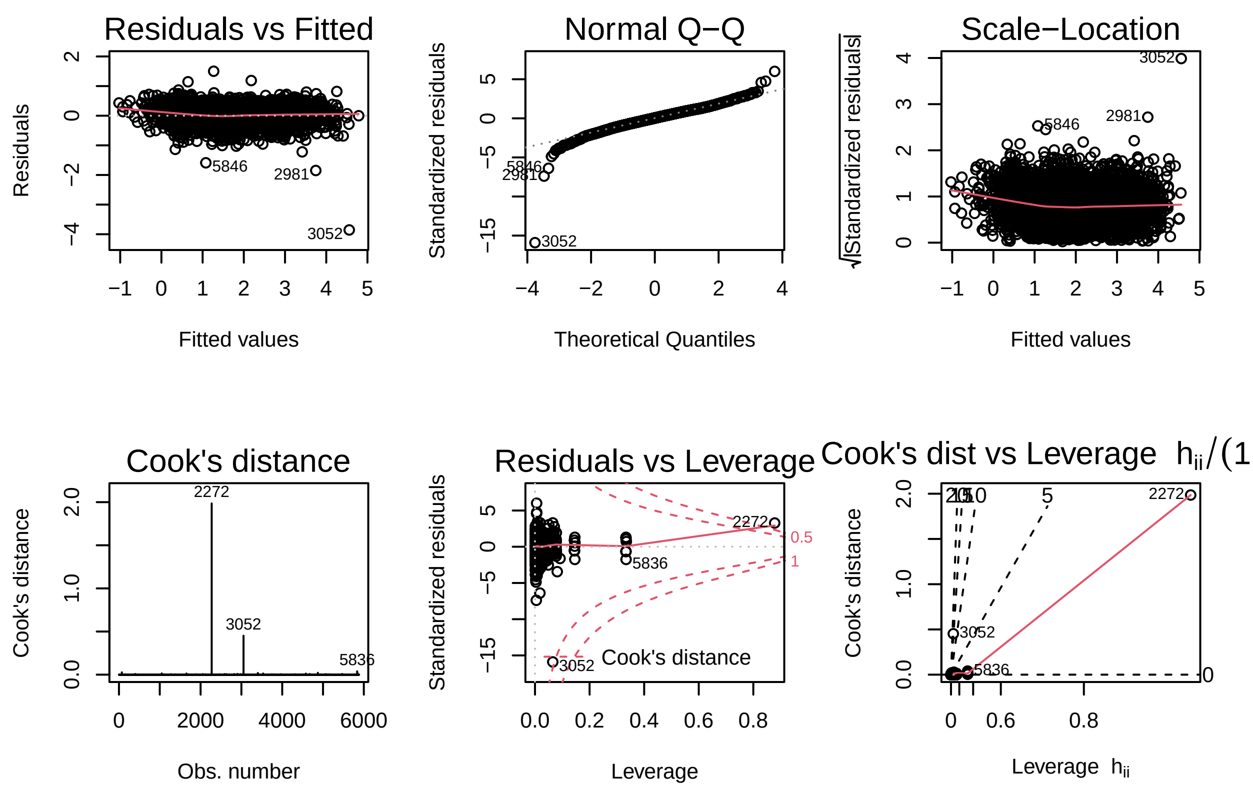
```
##
## Call:
## lm(formula = LPrice ~ LEngine * Power + Owner_Type + Mileage +
##     Year + Seats + Make + Kilometers_Driven, data = Cars_affordable)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64637 -0.14812  0.00935  0.16268  1.53152
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.486e+02  2.881e+00 -86.266  < 2e-16 ***
## LEngine                   8.283e-01  3.467e-02  23.891  < 2e-16 ***
## Power                     2.649e-02  2.461e-03  10.765  < 2e-16 ***
## Owner_TypeFourth & Above  3.171e-02  9.123e-02   0.348 0.728141
## Owner_TypeSecond         -5.794e-02  1.013e-02  -5.720 1.13e-08 ***
## Owner_TypeThird          -1.007e-01  2.673e-02  -3.768 0.000166 ***
## Mileage                   9.627e-03  1.190e-03   8.087 7.63e-16 ***
## Year                      1.209e-01  1.426e-03  84.763  < 2e-16 ***
## Seats                     8.845e-02  6.881e-03  12.854  < 2e-16 ***
## MakeChevrolet            -8.652e-01  2.425e-01  -3.568 0.000364 ***
## MakeDatsun               -9.578e-01  2.508e-01  -3.818 0.000136 ***
## MakeFiat                 -7.386e-01  2.468e-01  -2.992 0.002784 **
## MakeForce                -7.103e-01  2.788e-01  -2.548 0.010874 *
## MakeFord                 -5.782e-01  2.419e-01  -2.391 0.016844 *
## MakeHonda                -6.376e-01  2.419e-01  -2.636 0.008426 **
## MakeHyundai              -5.553e-01  2.418e-01  -2.297 0.021684 *
## MakeIsuzu                -6.954e-01  2.786e-01  -2.497 0.012572 *
## MakeJeep                 -4.090e-01  2.502e-01  -1.635 0.102185
## MakeMahindra             -7.497e-01  2.422e-01  -3.095 0.001977 **
## MakeMaruti               -5.184e-01  2.418e-01  -2.144 0.032105 *
## MakeMini                  4.829e-01  2.467e-01   1.958 0.050306 .
## MakeMitsubishi           -3.133e-01  2.457e-01  -1.275 0.202275
## MakeNissan               -5.859e-01  2.429e-01  -2.412 0.015887 *
## MakeRenault              -5.184e-01  2.425e-01  -2.137 0.032635 *
## MakeSkoda                -5.086e-01  2.424e-01  -2.098 0.035944 *
## MakeTata                 -9.187e-01  2.422e-01  -3.793 0.000151 ***
## MakeToyota               -3.829e-01  2.419e-01  -1.583 0.113515
## MakeVolkswagen           -5.475e-01  2.419e-01  -2.263 0.023684 *
```

```
## MakeVolvo                  -7.489e-02  2.481e-01  -0.302 0.762796
## Kilometers_Driven          -1.781e-07  1.043e-07  -1.708 0.087775 .
## LEngine:Power              -2.580e-03  3.168e-04  -8.145 4.75e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2397 on 4915 degrees of freedom
## Multiple R-squared:  0.8686, Adjusted R-squared:  0.8678
## F-statistic:  1083 on 30 and 4915 DF,  p-value: < 2.2e-16
```

```
#This model includes a large percentage of the affordable car brands. It does include Bentley, Lamborgh

#Kilometers Driven will serve as the control variable due to its very low (if any) statistical signific
```

```
par(mfrow = c(2,3))
plot(Reg_w_L, which = 1:6)
```

```
## Warning: not plotting observations with leverage one:
##   1190, 5388, 5643

## Warning: not plotting observations with leverage one:
##   1190, 5388, 5643
```



Step 5: Diagnostic plots

Residuals vs. Fitted

As we can see, for the residuals vs. fitted portion, the models is doing well and things look great for the most part. Non-linearity is not violated. The residuals are for the most part, bouncing randomly around the 0 line and are primarily horizontal. However, there is an outlier (entry 3133).

QQ Plot

The model demonstrates homoskedacity. The QQ plot also looks solid, the points are on an upward trajectory but do no fall perfectly along this line. This is quite good. However, again, entry 3133 is at least slightly alarming.

Scale-Location

Heteroskedacitiy does no appearThe Scale-Location plot looks quite good. The spread across the red like does not appear to vary with regards to values. Entry 3060 is a bit troubling and entry 3133 makes an appearance yet again.

Cook's Distance

The Cook's Distance plot looks fine. Entry 1222 is worth investigating but it looks promising for the most part.

Residuals vs. Leverage

The Residuals vs. Leverage plot looks acceptable. Points are well outisde of the dashed lines.

Cook's Distance vs. Leverage

Overall, Cook's Distance vs. Leverage is complex and can be confusing to read. Therefore, this particular plot will not be used for comparison.

Investigation of Outliers (3060)

```
Cars_Data_Imputed[Cars_Data_Imputed$Name=="BMW 3 Series 320d Luxury Line",]
```

```
##                               Name Make Model   Location Year Kilometers_Driven
## 111  BMW 3 Series 320d Luxury Line  BMW     3     Mumbai 2015             56087
## 547  BMW 3 Series 320d Luxury Line  BMW     3  Bangalore 2014             47000
## 710  BMW 3 Series 320d Luxury Line  BMW     3      Kochi 2015             58390
## 742  BMW 3 Series 320d Luxury Line  BMW     3 Coimbatore 2016             14351
## 1108 BMW 3 Series 320d Luxury Line  BMW     3     Jaipur 2013             62655
## 1164 BMW 3 Series 320d Luxury Line  BMW     3    Chennai 2012             65000
## 1286 BMW 3 Series 320d Luxury Line  BMW     3      Kochi 2013             37613
## 1352 BMW 3 Series 320d Luxury Line  BMW     3  Hyderabad 2013             30000
## 2446 BMW 3 Series 320d Luxury Line  BMW     3     Mumbai 2014             18600
## 2886 BMW 3 Series 320d Luxury Line  BMW     3      Kochi 2017             55389
## 2981 BMW 3 Series 320d Luxury Line  BMW     3      Delhi 2019             87000
```

```
## 3788 BMW 3 Series 320d Luxury Line  BMW     3     Mumbai 2013             38000
## 5198 BMW 3 Series 320d Luxury Line  BMW     3  Hyderabad 2015             47000
## 5429 BMW 3 Series 320d Luxury Line  BMW     3      Kochi 2016             62404
##      Fuel_Type Transmission Owner_Type Mileage Engine Power Seats Price
## 111     Diesel    Automatic      First   22.69   1995   190     5 20.75
## 547     Diesel    Automatic      First   18.88   1995   184     5 25.50
## 710     Diesel    Automatic      First   18.88   1995   184     5 19.86
## 742     Diesel    Automatic      First   18.88   1995   184     5 35.55
## 1108    Diesel    Automatic      First   18.88   1995   184     5 14.50
## 1164    Diesel    Automatic      First   22.69   1995   190     5 14.00
## 1286    Diesel    Automatic      First   22.69   1995   190     5 13.95
## 1352    Diesel    Automatic      First   18.88   1995   184     5 22.00
## 2446    Diesel    Automatic     Second   22.69   1995   190     5 21.00
## 2886    Diesel    Automatic      First   18.88   1995   184     5 28.45
## 2981    Diesel    Automatic      First   22.69   1995   190     5  6.67
## 3788    Diesel    Automatic      First   18.88   1995   184     5 19.50
## 5198    Diesel    Automatic      First   22.69   1995   190     5 29.50
## 5429    Diesel    Automatic      First   22.69   1995   190     5 21.33
##        LPrice  LEngine
## 111  3.032546 7.598399
## 547  3.238678 7.598399
## 710  2.988708 7.598399
## 742  3.570940 7.598399
## 1108 2.674149 7.598399
## 1164 2.639057 7.598399
## 1286 2.635480 7.598399
## 1352 3.091042 7.598399
## 2446 3.044522 7.598399
## 2886 3.348148 7.598399
## 2981 1.897620 7.598399
## 3788 2.970414 7.598399
## 5198 3.384390 7.598399
## 5429 3.060115 7.598399
```

```
#For entry 3060, it happens to be a BMW 3 Series 320d Luxury Line. Fortunately, there are other entries
```

Investigation of Outliers (3133)

```
Cars_Data_Imputed[Cars_Data_Imputed$Name=="Porsche Cayenne Base",]
```

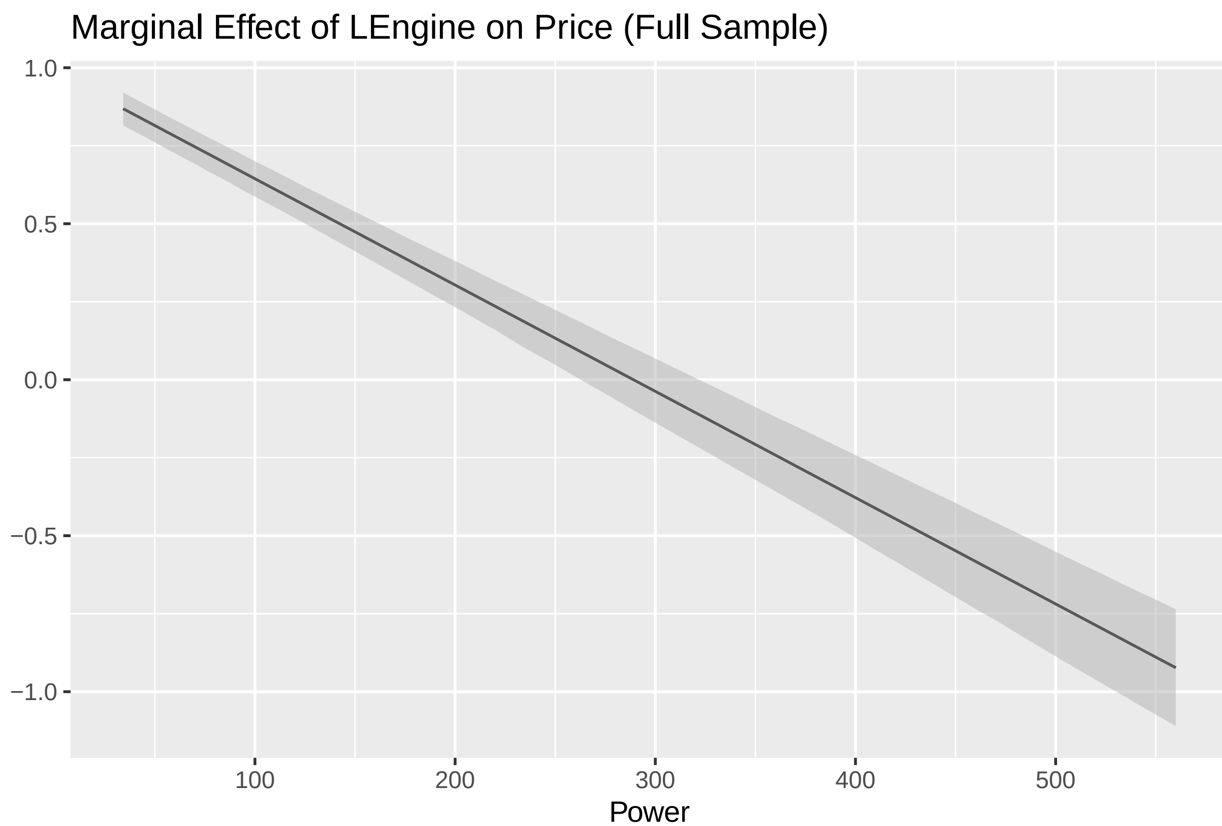
```
##                      Name    Make   Model Location Year Kilometers_Driven
## 3052 Porsche Cayenne Base Porsche Cayenne    Kochi 2019             14298
##      Fuel_Type Transmission Owner_Type Mileage Engine Power Seats Price
## 3052    Petrol    Automatic      First   13.33   2995   340     5  2.02
##         LPrice LEngine
## 3052 0.7030975  8.0047
```

```
#This Porsche Cayenne Base has an especially low price. Especially given that the car has an MSRP aroun
```

Step 6: Modeling Continued

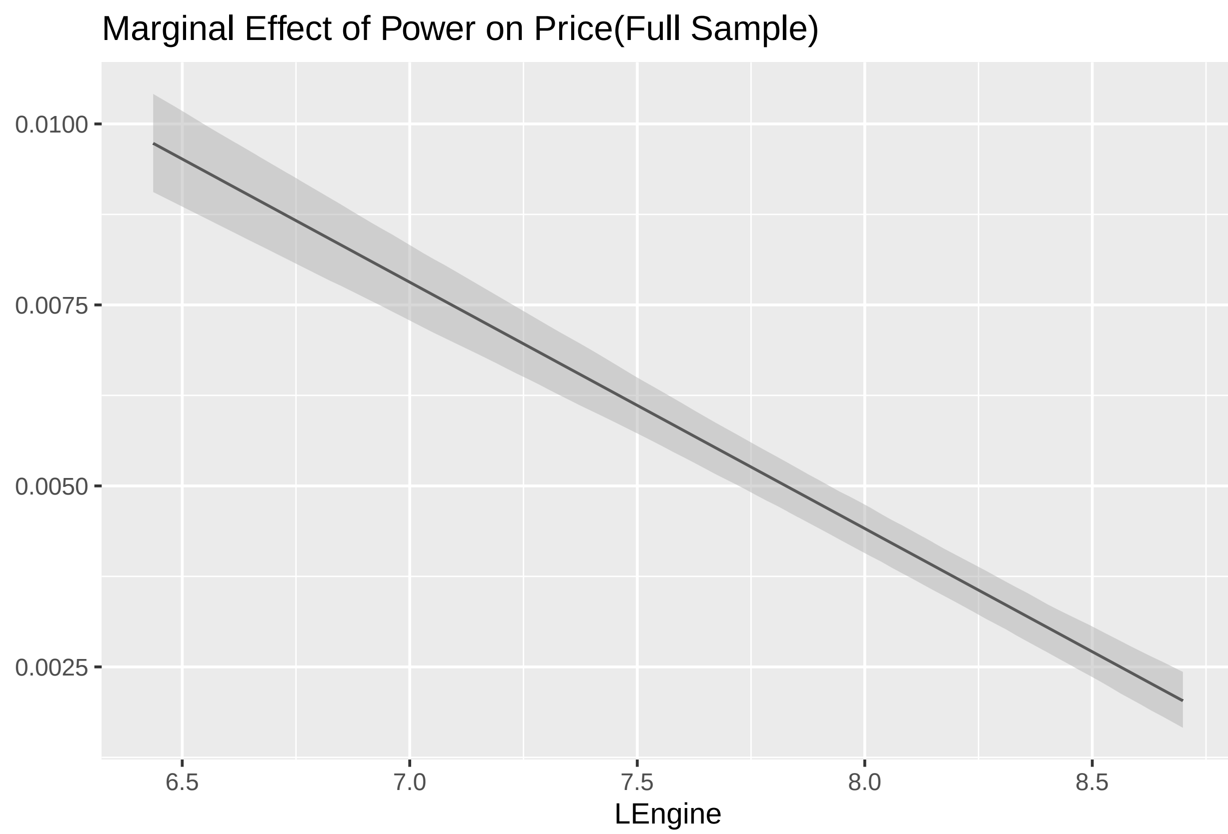
Inclusive Cars Model (Full Sample)

```
interplot(m = Reg_w_L, var1 = "LEngine", var2 = "Power") +
   xlab('Power') +
  ggtitle('Marginal Effect of LEngine on Price (Full Sample)')
```



For cars with hp under 290, the marginal effect of engine size is positive. This means that for those cars, higher engine size leads to higher price. However, this effect goes down as power goes up such that for cars with power above 300 hp, the effect of engine size on car price is negative.

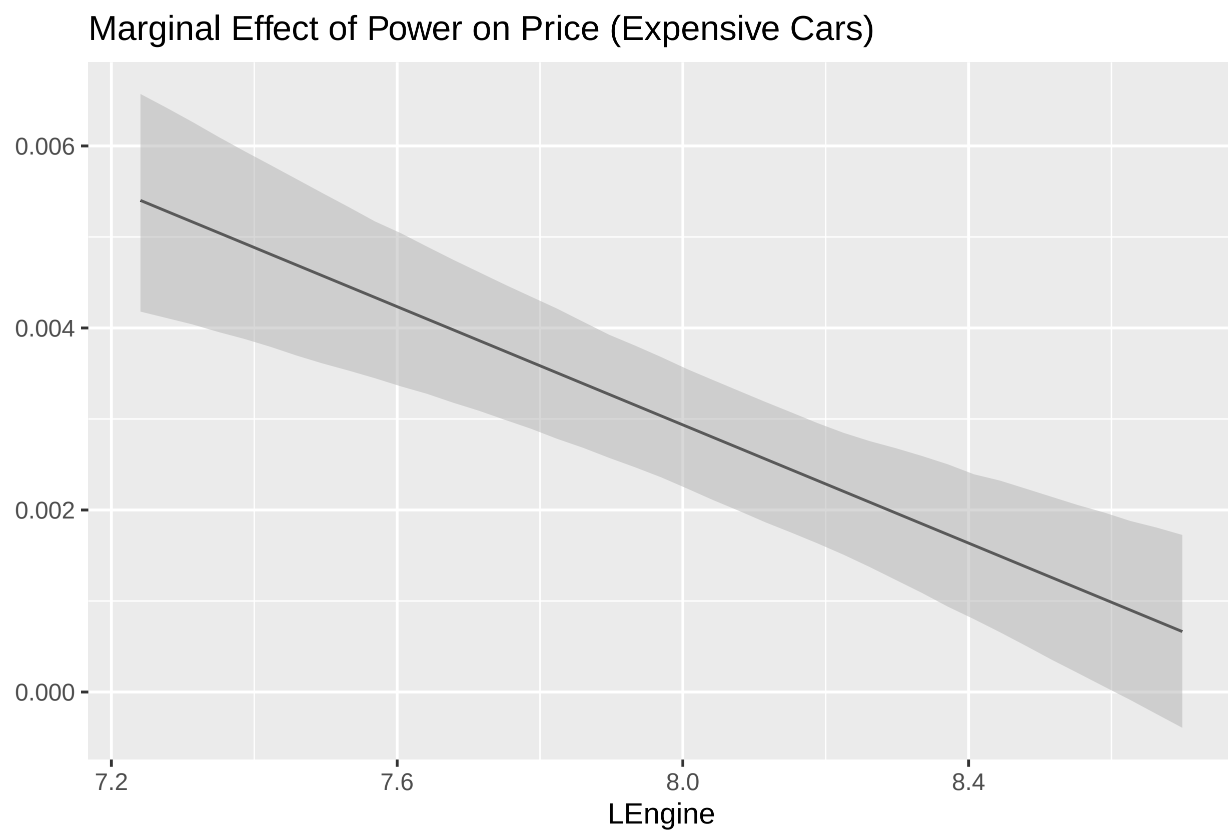
```
interplot(m = Reg_w_L, var1 = "Power", var2 = "LEngine") +
  xlab('LEngine') +
  ggtitle('Marginal Effect of Power on Price(Full Sample)')
```



Marginal effect of power on price is decreasing as engine size of the car increases. However, this effect is always positive for the range of engine sizes in the sample (it will turn into negative only if LEngine is larger than 11 which implies engine size above 59,000 cc). Therefore, the Marginal Effect of power will always be positive regardless of engine size.

Expensive Car Models

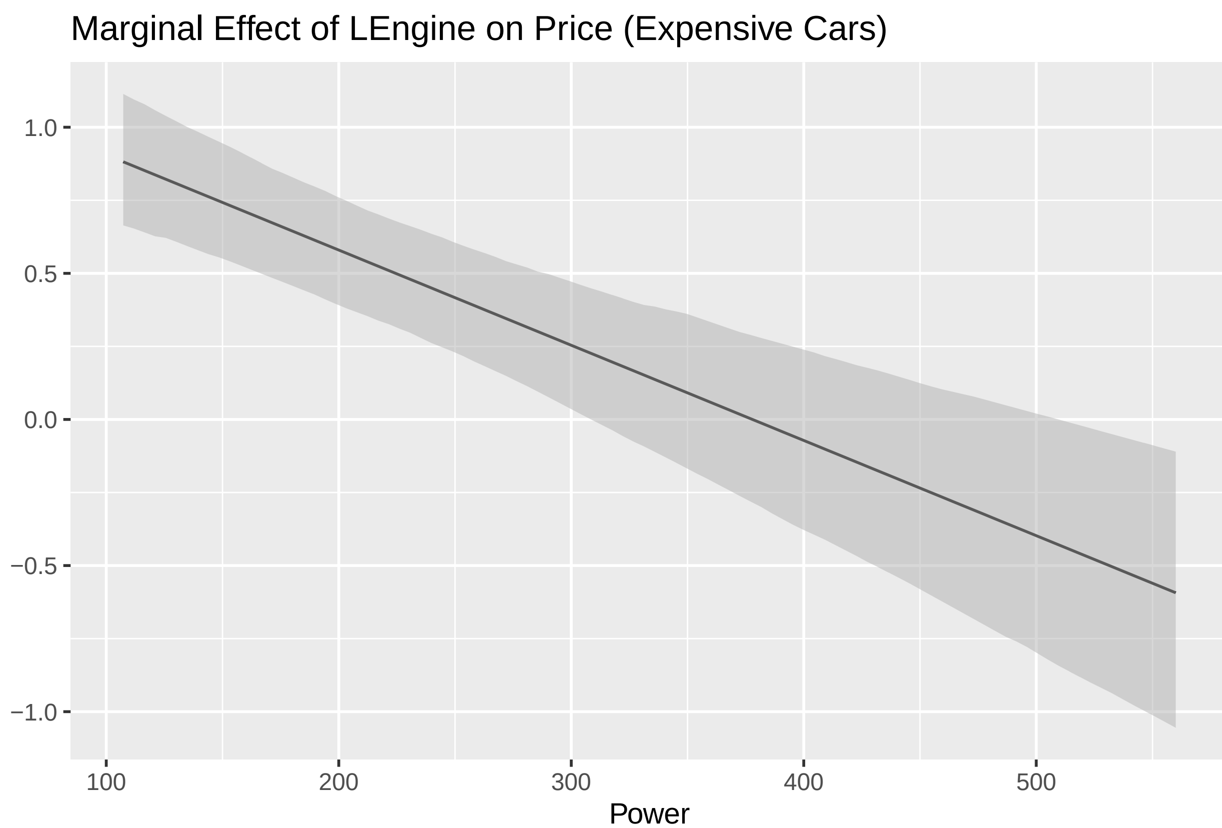
```
interplot(m = Reg_w_L_expensive, var1 = "Power", var2 = "LEngine") +
  xlab('LEngine') +
  ggtitle('Marginal Effect of Power on Price (Expensive Cars)')
```




```
expensive_1 <- interplot(m = Reg_w_L_expensive, var1 = "Power", var2 = "LEngine") +
  xlab('LEngine') +
  ggtitle('Marginal Effect of Power on Price (Expensive Cars)')
```

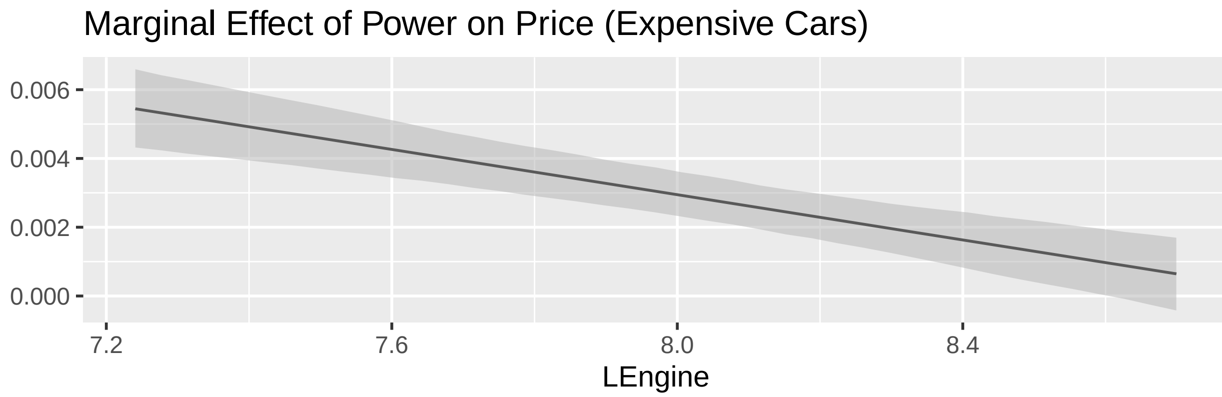
At every engine size, the marginal effect of power on price is higher for affordable cars compared to that effect for expensive cars.

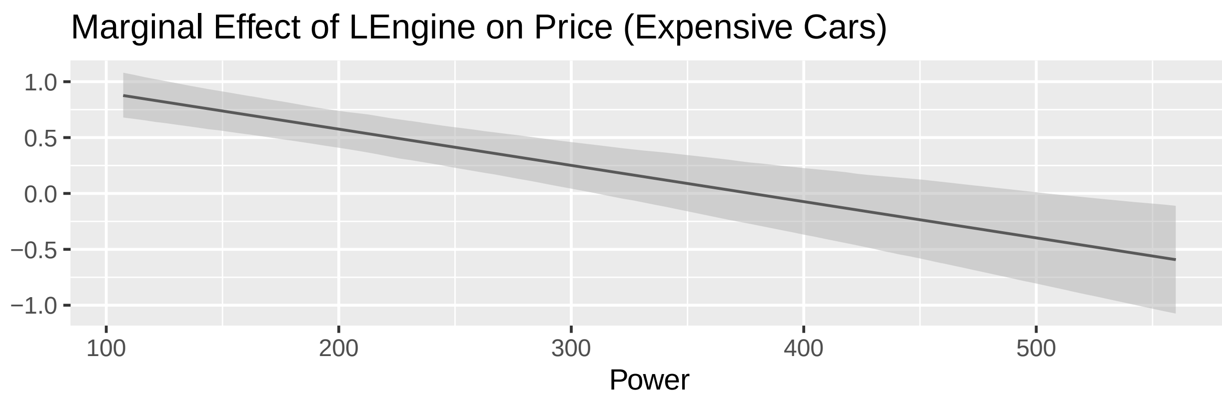
```
interplot(m = Reg_w_L_expensive, var1 = "LEngine", var2 = "Power") +
  xlab('Power') +
  ggtitle('Marginal Effect of LEngine on Price (Expensive Cars)')
```




```r
expensive_2 <- interplot(m = Reg_w_L_expensive, var1 = "LEngine", var2 = "Power") +
  xlab('Power') +
  ggtitle('Marginal Effect of LEngine on Price (Expensive Cars)')
```

```r
ggarrange(expensive_1, expensive_2, nrow = 2)
```

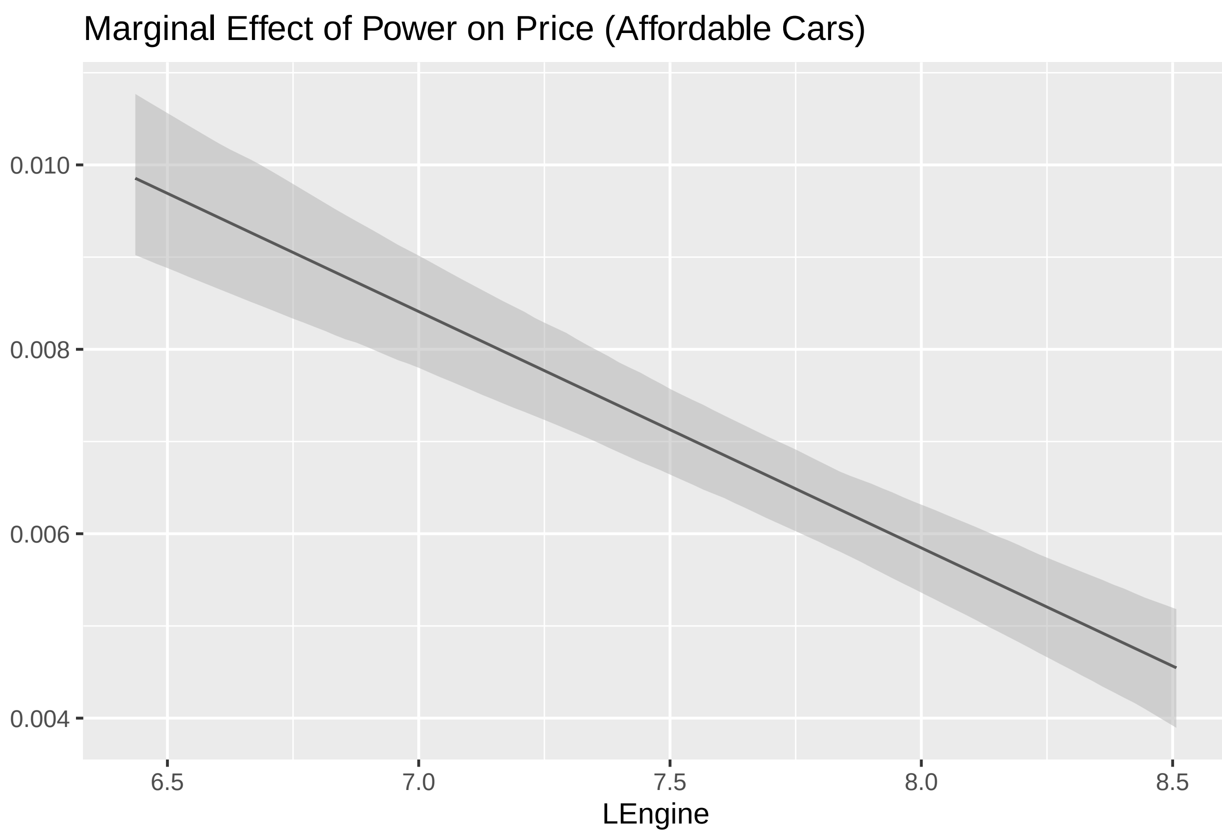




For expensive, we can observe that the Marginal Effect of LEngine is positive for cars up with horsepower up to approximately 360.

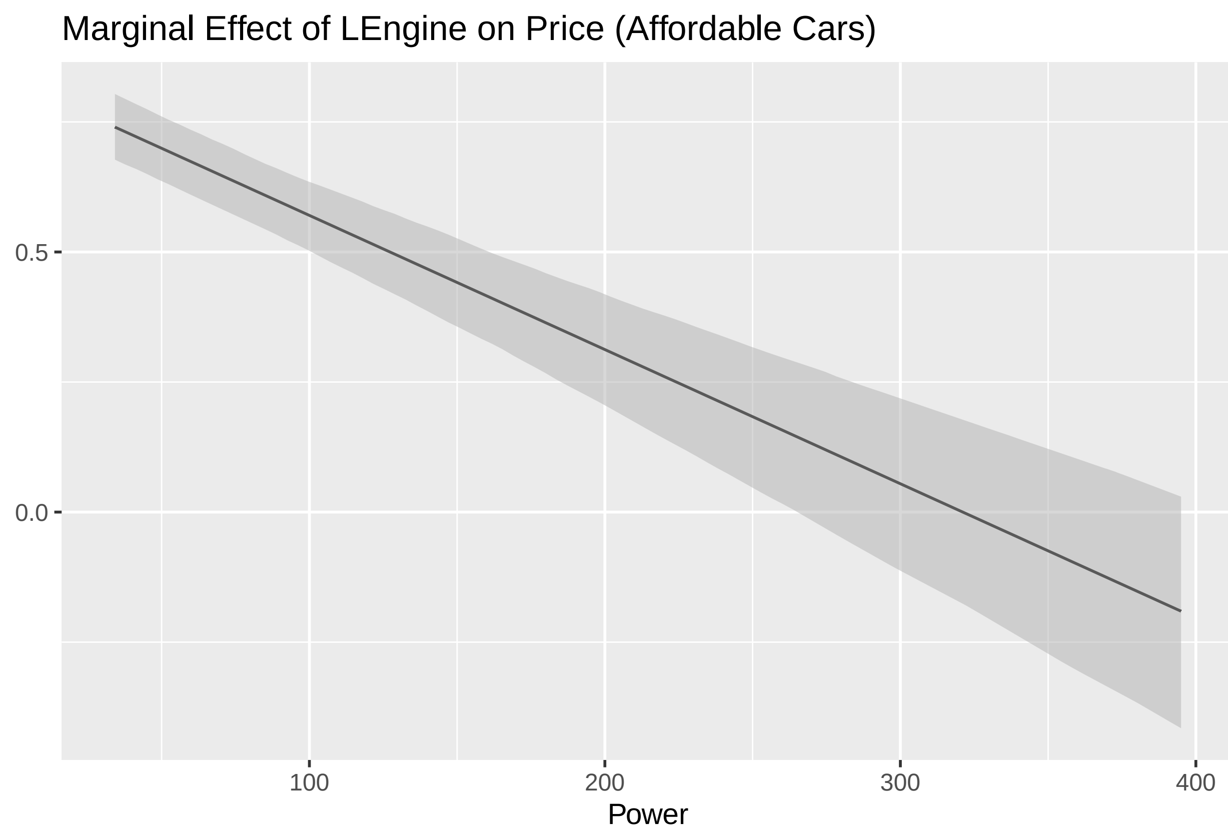
Affordable Car Models

```
interplot(m = Reg_w_L_affordable, var1 = "Power", var2 = "LEngine") +
  xlab('LEngine') +
  ggtitle('Marginal Effect of Power on Price (Affordable Cars)')
```



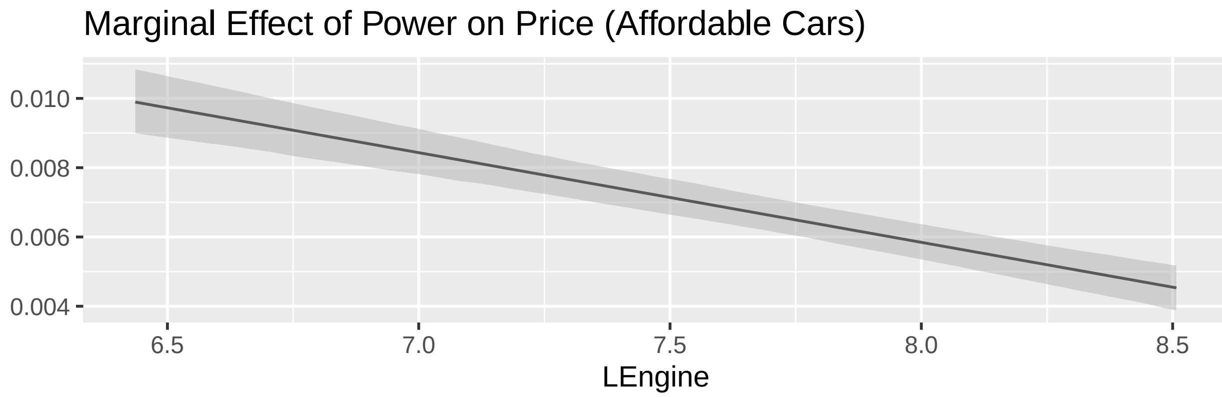

```r
affordable_1 <- interplot(m = Reg_w_L_affordable, var1 = "Power", var2 = "LEngine") +
  xlab('LEngine') +
  ggtitle('Marginal Effect of Power on Price (Affordable Cars)')
```

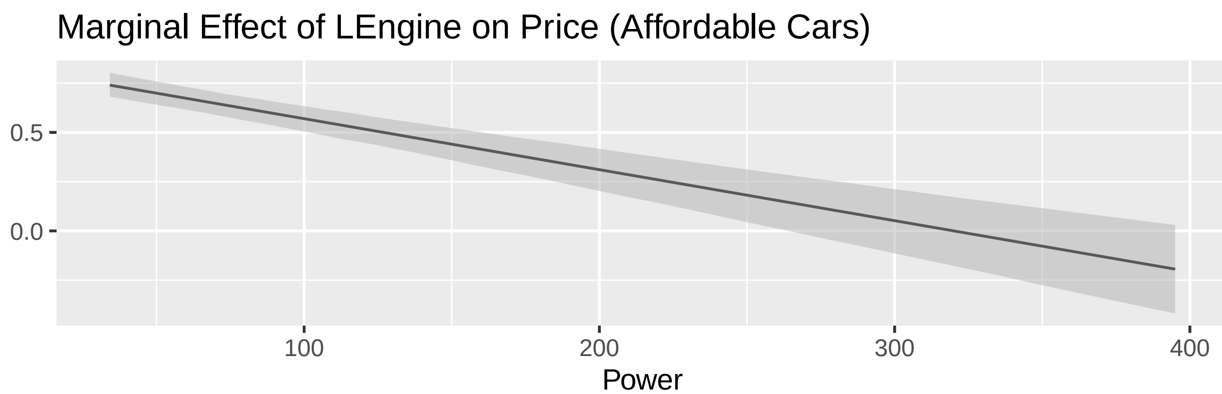
```r
interplot(m = Reg_w_L_affordable, var1 = "LEngine", var2 = "Power") +
  xlab('Power') +
  ggtitle('Marginal Effect of LEngine on Price (Affordable Cars)')
```




```
affordable_2 <- interplot(m = Reg_w_L_affordable, var1 = "LEngine", var2 = "Power") +
  xlab('Power') +
  ggtitle('Marginal Effect of LEngine on Price (Affordable Cars)')
```

```
ggarrange(affordable_1, affordable_2, nrow = 2)
```





For affordable cars, we can observe that the Marginal Effect of LEngine is positive for cars up with horsepower up to approximately 320.

Conclusion

Overall, people wish to buy cars with smaller sized engines in cc's that deliver optimal horsepower. This effect is even larger for affordable cars compared to expensive cars. This is likely because when customers go to buy affordable cars, they at least wish for power. In the case of expensive cars, they are less particular about how much power the car has due to the luxury features.