HW5 - Penalized, All Subsets Regression

William Morgan, Mitch O'Brien, Jared Scolaro

March 9, 2018

Contents

1. Data with 5 x's 1
2. Used Cars Data, Lasso vs. Step/All subsets 2

---

1. Data with 5 x's

Use `leaps::regsubsets` and the lasso to see what variable subsets they select

```
##   (Intercept)    x1    x2    x3    x4    x5
## 1        TRUE FALSE FALSE FALSE FALSE  TRUE
## 2        TRUE FALSE  TRUE FALSE FALSE  TRUE
## 3        TRUE  TRUE  TRUE FALSE FALSE  TRUE
## 4        TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## 5        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) -0.02465237
## x1           0.02470953
## x2           0.03468792
## x3           1.84435483
## x4           .
## x5           0.03625435
```

`leaps::regsubsets` selects all five variables, while the lasso only excludes $x4$

2. Used Cars Data, Lasso vs. Step/All subsets

Outline:

We begin the second half of this homework with a restatement of the problem, followed by a quick description of the data and an exploratory analysis to get a firm grasp on the available information. Next, we discuss our methods for preprocessing the data, including any cleaning and feature engineering, before moving on to model selection techniques. This question concludes with model evaluation and the selection of the best performing model.

Problem:

Build a linear model to predict the price of a used car using penalized regression and stepwise/all subsets selection methods. Evaluate the models from each methodology and determine which gives the most accurate predictions.

Data Description:

We use the `usedcars.csv` data set on the course web page to build the models, which contains descriptions of about 20,000 used cars. There are three continuous fields, `displacement` and `mileage`, and `price`. `displacement` measures the combined swept volume of the pistons inside the cylinders of an engine and can be seen as a measure of engine power, and `mileage` is simply the existing mileage on the car at the time of purchase. `price` is measured in USD. Other fields in the data set include `trim`, `isOneOwner`, `year`, `color`, `fuel`, `region`, `soundSystem`, and `wheelType`.

Most of the categorical variables are self-explanatory, so we won't define them here. `trim` isn't as obvious for non-car enthusiasts, so we looked up a basic explanation of it: "Trim levels identify a vehicle by a set of special features. Higher trim levels add to the base model features or replace them with something else.." (edmunds.com). Also, `trim` values containing "AMG" is apparently a trim option from Mercedes-Benz that are tuned in a specific way to upgrade the power of the car (mercedesbenznaples.com)

Exploratory Analysis:

The primary purpose of this analysis is to evaluate the quality of the variables included in the data set. Specifically, we want to observe their distributions and verify that they are appropriate to use in a regression scenario. We do not anticipate to find any problems with the continuous variables that would warrant exclusion, but we believe there might be some issues with the categorical variables. Namely, we want to avoid variables that 'pile up' in one level of a factor instead of being evenly distributed amongst the classes. This might become an issue when we are splitting the data and doing any sort of k-fold CV, because we could by chance end up with folds that have no observations for the minority class within a particular variable.

A less important (but still relevant) goal of this analysis is to look for observations that should be dropped from the data. As this was already prepared by someone else, it is unlikely any dramatic changes will have to be made.

Finally, we use this section to create a couple of visualizations and cross-tabulations to inform which variables and interactions might be relevant in modeling price.

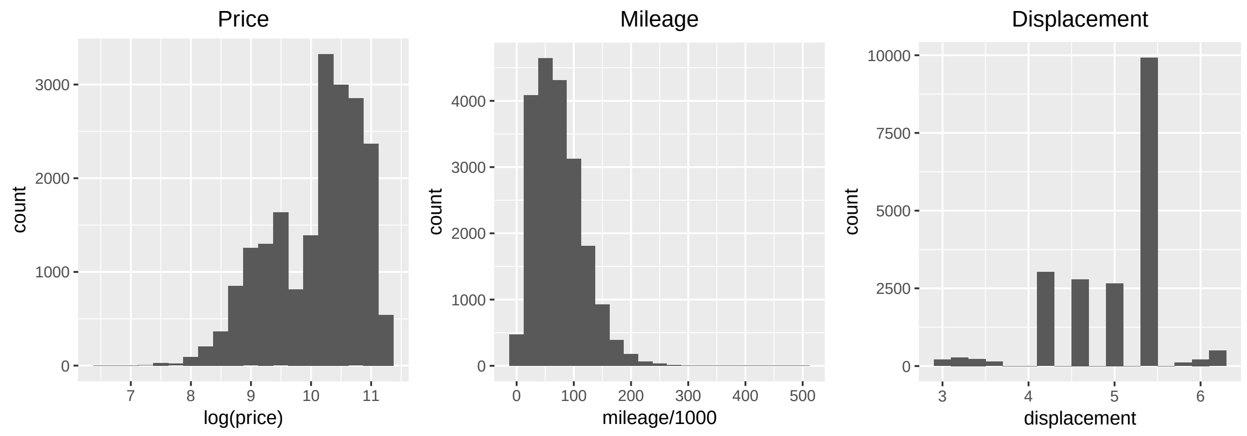
We begin by looking at frequency counts for the non-numeric variables:

```
##       trim       isOneOwner      year         color           fuel
##  550    :11825   f:16594    2007   :3607   Black :8194   Diesel  :  211
##  430    : 2787   t: 3469    2008   :2352   Blue  : 914   Gasoline:19632
##  500    : 2661              2012   :1872   Gray  :2168   Hybrid  :  220
##  63 AMG :  599              2011   :1625   other : 961
##  600    :  572              2010   :1624   Silver:4353
##  55 AMG :  356              2013   :1169   unsp  : 832
##  (Other): 1263              (Other):7814   White :2641
##      region            soundSystem     wheelType
##  SoA    :6492   Bang Olufsen : 104   Alloy  :11111
##  Pac    :3252   Bose         :1261   other  :  120
##  Mid    :2671   Harman Kardon:4278   Premium:  428
##  WSC    :2493   Premium      :5320   unsp   : 8404
##  ENC    :1984   unsp         :9100
##  Mtn    :1006
##  (Other):2165
```

More than 95% of the observations in the data have `fuel = 'Gasoline'`, so that will certainly cause some problems when we split up the data. `trim` and `wheelType` also have lopsided distributions, but not as nearly as bad as `fuel`. We have almost on observations with `soundSystem = Bang Olufsen`, so it would be wise to drop those observations to avoid the problem we described earlier. Further investigation is needed for variables with values of `unsp` because it is not immediately clear what that signifies. It is especially important for `soundSystem` because `unsp` is the majority class of that variable.

Let's now inspect the continuous variables with some histograms. We adjust `mileage` to be reported in thousands of dollars and miles (resp.) for easier interpretation.

`price` is log-transformed for the same reason, and we will keep `price` in this form for modeling as well.



We mainly want to identify outliers with these plots. `price` appears to have very few on the low end but for the most part seems reasonable. `mileage` on the other hand has some on the far right of the distribution. Finally, `displacement` does not appear to be continuously distributed, so it will likely serve better as a categorical variable.

Before moving on, we will look at the observations containing these outliers to briefly check if they appear to be reporting errors or actual sales. We will wait on factorizing `displacement` and do it in the preprocessing section.

Price outliers:

```
## # A tibble: 10 x 11
##    price trim   isOneOwner mileage year  color  displacement fuel   region
##    <int> <fct>  <fct>        <dbl> <fct> <fct>         <dbl> <fct>  <fct>
##  1   599 430    f           356116 2000  unsp           4.30 Gasol~ Mid
##  2   699 430    f           141039 2002  Silver         4.30 Gasol~ SoA
##  3   800 320    f           138060 1997  unsp           3.20 Gasol~ SoA
## 4    995 55 AMG f           120232 2002  Blue           5.40 Gasol~ SoA
## 5    995 500    f           200000 2001  unsp           5.00 Gasol~ ENC
## 6    995 320    f           175744 1995  Blue           3.20 Gasol~ WSC
## 7   1000 320    f           149226 1999  Silver         3.20 Gasol~ ESC
## 8   1450 320    f           225814 1995  Black          3.20 Gasol~ WSC
## 9   1500 420    f           130000 1996  unsp           4.20 Gasol~ WSC
## 10  1500 550    f            46325 2010  White          5.50 Gasol~ WSC
## # ... with 2 more variables: soundSystem <fct>, wheelType <fct>
```

Mileage outliers:

```
## # A tibble: 10 x 11
##    mileage price trim  isOneOwner year  color displacement fuel     region
##      <dbl> <int> <fct> <fct>      <fct> <fct>        <dbl> <fct>    <fct>
## 1   488525 46995 550   f          2012  White         4.60 Gasoline Mid
## 2   467834 21995 500   f          2006  Black         5.00 Gasoline WSC
## 3   407725  8995 500   f          2000  other         5.00 Gasoline Pac
## 4   356116   599 430   f          2000  unsp          4.30 Gasoline Mid
## 5   326768 19000 550   f          2007  Blue          5.50 Gasoline ENC
## 6   319626  3999 430   f          2003  other         4.30 Gasoline Mid
## 7   315340 11925 550   f          2007  Black         5.50 Gasoline New
## 8   314190  5994 430   f          2003  Gray          4.30 Gasoline ESC
## 9   312831  2900 320   f          1995  White         3.20 Gasoline SoA
## 10  310000  7999 430   f          2000  other         4.30 Gasoline SoA
## # ... with 2 more variables: soundSystem <fct>, wheelType <fct>
```

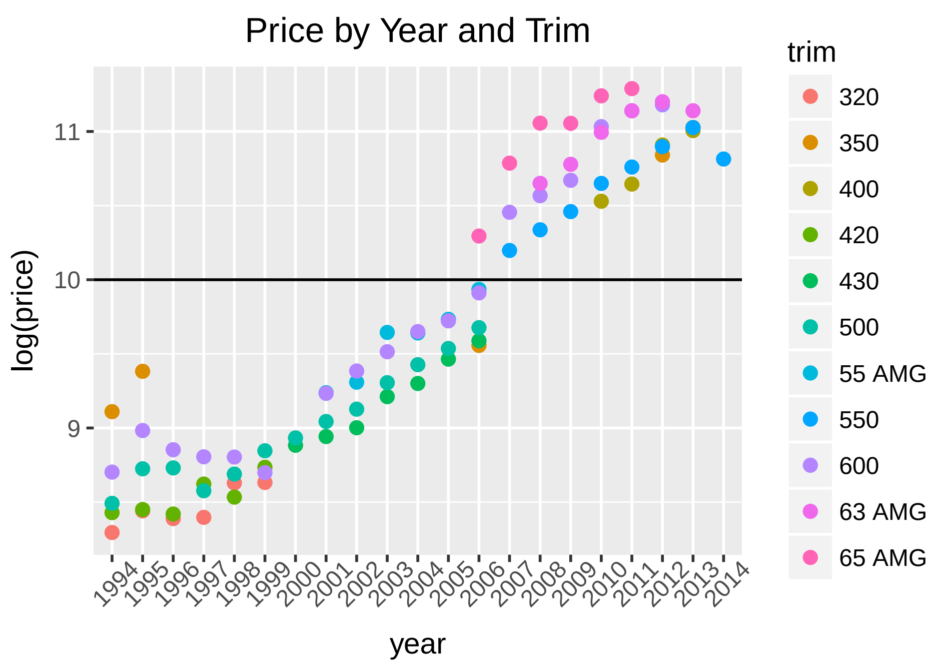
There doesn't seem to be anything out of place with either set of outliers (although the `mileage` outliers seem extremely large), so we will keep them in the data set.

In summary, the data is by and large ready to be processed for modeling. The only problems we found were the very low variation in `fuel` and the `unsp` values in a couple of the columns. We remedy the first by deleting the column altogether, and we discuss our approach to dealing with the irregular values in the following section.

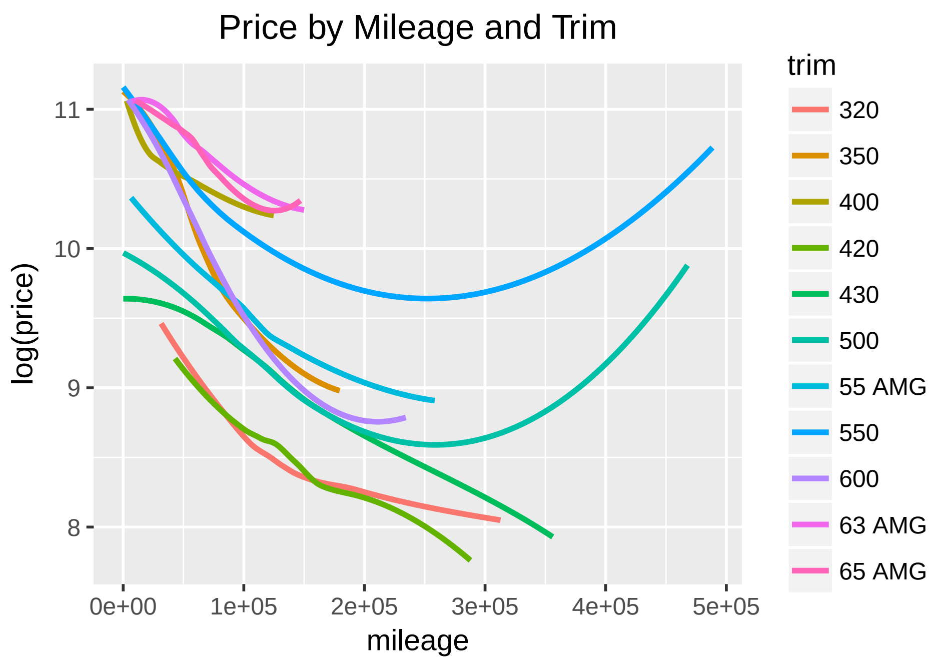
We conclude this section with a quick investigation into possible interactions and higher-order relationships to include in the model. For brevity we have excluded some of the cross-tabulations we thought were unimportant for the report.

Investigating Interactions:

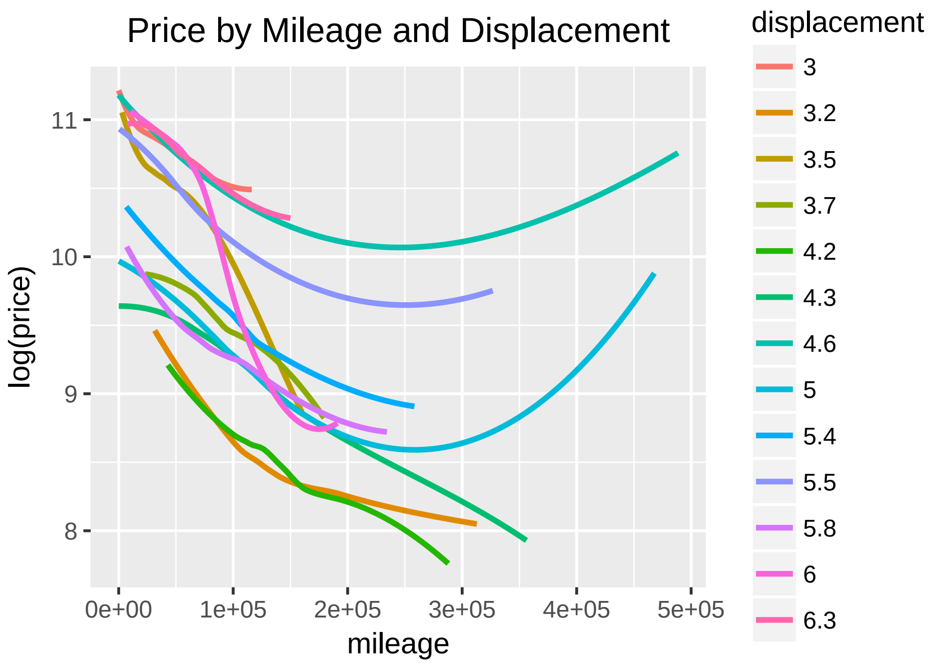
- The first interaction to check is `year` and `trim`. The figure below plots the relationship between price, year, and trim. Points on the figure represent class means for a given year. As expected, higher trim values are associated with higher sell prices. This plot reveals that certain trim values only exist within a given time frame. 65 AMG for instance was seemed to be introduced in 2005, having no observations before that time. This suggests that we should not use the interaction in the regression, as there would be many instances of entire columns of interactions being completely zero.



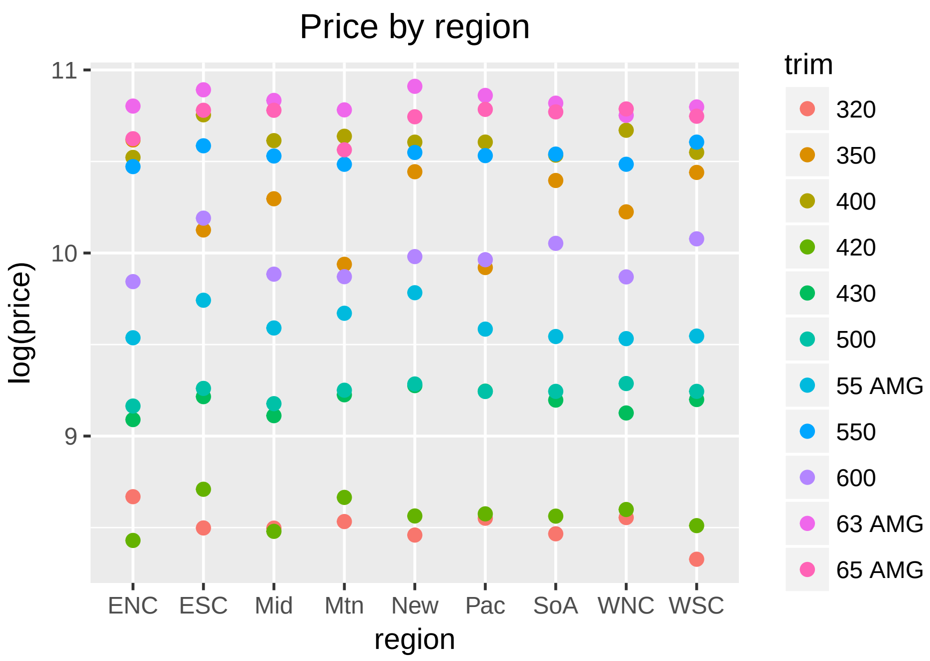
- We now check the interaction between `mileage` and `trim`. To make the relationships a bit more obvious we've used smoothing curves for each trim style instead of points. These reveal clear differences in selling prices for different trims at a given mileage, so we should include this interaction.



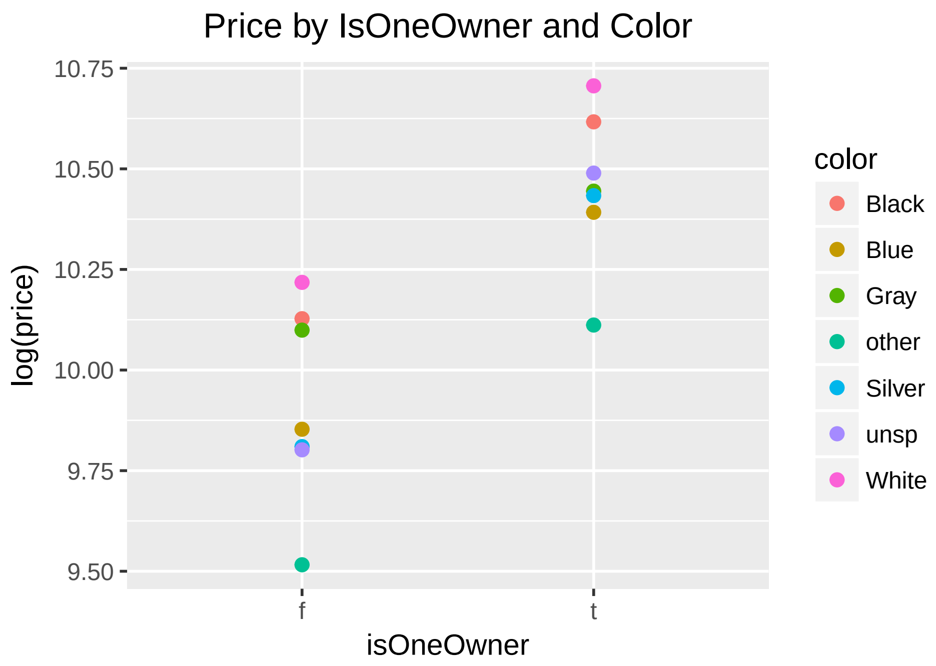
- For the same reason as the above interaction, we have evidence to include an interaction between `mileage` and `displacement` in our model.



- Our next pair of variables to check is `region` and `trim`. We would have support for their interaction being included in the model if price varied significantly for a given trim across regions, but that does not seem to be the case. For example, for `trim = 600` we can see that the price is pretty consistent across region. Hence, adding the region for a given trim style does not reveal any more information about the selling price.

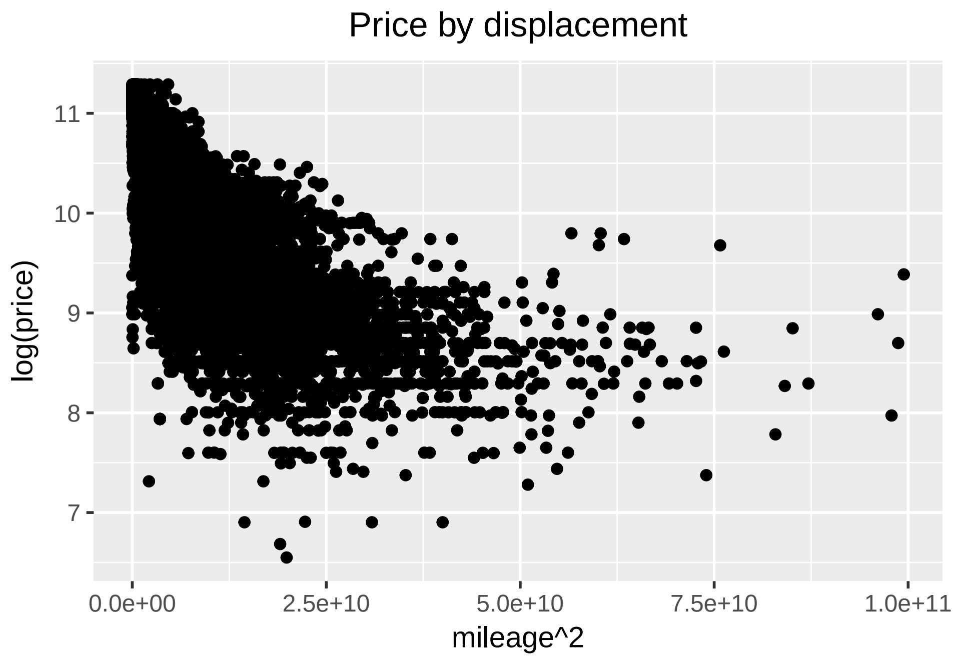


- The final interaction we will check for is `isOneOWner` and `color`. This plot does not give much support for putting the interaction in, as the slope for each color moving from `isOneOwner = F` to `isoneOwner = T` appears to be the same across most colors. In any case, it might be worth considering including anyway.



Higher Order Terms

- The only continuous value we are using in the regression is `mileage`, so we plot $mileage^2$ to see if there is evidence to include it in the model. We also checked its cubic form, but have excluded it here because there was no meaningful relationship. Based on the plot it seems that there is at least some relation, so we will include it in the model.



In short, the extra terms we think are worth including are:

- `mileage` and `trim`
- `mileage` and `displacement`

- `color` and `isOneOwner` (maybe)
- `mileage^2`

Preprocessing

As we noted earlier, `fuel` does not seem like an appropriate variable to include in the models because of its poor variation. For the same reason, observations with `soundSystem` values of `Bang Olufsen` will be dropped. In addition, the `unsp` values present a difficult problem. Assuming that this value means "unspecified", the interpretation of any coefficients containing this variable won't be very meaningful. It could very well be the case that two cars reporting `unsp` for `soundSystem` have entirely different sound systems in reality. If that were the case, any signal we were trying to extract would be drowned out. An obvious solution is to drop any observations containing this value, however this would amount to dropping a significant portion of the data. On the other hand, our goal isn't finding an accurate description of the relationship. Rather, we just want to build the best predictive model. So, we will move forward by duplicating our data and running two of the same analyses - one on the entire data set and one on the data set with no `unsp` values. This will allow us to move forward without sacrificing a majority of the data and also investigate the amount of noise `unsp` adds.

Once we drop `fuel`, factorize `displacement`, and create the subset without `unsp` values, we can set up the data for modeling. This will entail splitting the data into training/testing sets, centering and scaling the continuous columns, and deciding on formulas to use.

Our training/testing split will be the standard 80/20 split. We go about making This is done with the use of `caret::createDataPartition`, which allows us to split our data based on the distribution of the outcome variable `price`. We scale both sets of data by the sample means and standard errors of the training set.

The only 'feature engineering' we will do on this data set is to collapse the `year` into larger categories in order to reduce the number of coefficients we will have to estimate. This will speed up computation considerably because `year` already has 20 levels and we won't be sacrificing much in terms of interpretability if we condense that a little bit. (After checking it out, this reduces the number of estimates from 1500 to 840 in the 'interact everything' model). The new levels of `year` will be:

- anything before and including 2000
- 2001-2005
- 2006-2009
- 2010-2013

Model Selection and Estimation:

We take two approaches to feature selection: a data-informed perspective on important features and a 'kitchen sink' method. We were able to gain some insight on the relations between price and various columns in the data, giving us a list of extra terms that seemed relevant to include. On the other hand, the low dimensionality of the original data means that we can be pretty liberal in deciding what to include before we start running into computational issues. So, we will run a set of models based on our initial findings from the previous sections, and a set of models that are much broader and less founded in the data.

We first define the everything we need for the penalized regression. For any given formula, our goal is to run the cross-validated `glmnet` function with $\alpha = 0, .5, 1$ (ridge, enet, lasso), create predictions for each model, and calculate the root mean square error. This function, called `glmnetPrediction`, will take a formula in the form of a string (this is done to easily keep track of it), the training data frames, and the testing data frames. The actual code is not included in this report to save space. Next, we repeat the same thing for the stepwise selection. It will have the same basic setup: for a given formula, run the algorithm, find the best model, and report the test error.

Once the regression architecture is set up, we can test it by defining a ton of formulas and shoving it through the functions we just defined. This should give us some preliminary results on which formulas will work best. To be explicit, we include the formulas we decided to test in the following code chunk

```r
# define formulas and vectorize
f0 <- 'lprice ~ mileage + trim'                    # Example model (for testing code)
f1 <- 'lprice ~ .'                          # ALL
f2 <- 'lprice ~ .+I(mileage^2)'             # ALL + mileage^2
f3 <- 'lprice ~.+I(mileage^3)'              # ALL + mileage^2
f4 <- 'lprice ~.+I(1/mileage)-mileage'      # 1/mileage instead of mileage
f5 <- 'lprice ~.+I(mileage^3)+I(mileage^2)' # mileage^2 and mileage^3
f6 <- 'lprice ~.+mileage:trim'                 #mileage * trim
f7 <- 'lprice ~.+mileage:displacement'         #mileage * displacement
f8 <- 'lprice ~.+mileage:trim+mileage:displacement' #mileage * (trim, displacement)
f9 <- 'lprice ~.+color:isOneOwner'                  #color * oneowner
f10 <- 'lprice ~.*isOneOwner'                       #everything interacted with isOneOwner
f11 <- 'lprice ~.^2'                                #everything

formulas <- c(f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11)
rm(f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11)
```

Evaluation

At this point we have the RMSE of the demeaned `lprice` variable for each model that was run. To be more specific, for every formula specified above, we have the RMSE for a elastic net, ridge, and lasso regression along with the RMSE of the best subset selection method. Combining the results will allow us to determine the highest performing model on each set of test data.

A secondary question we had was to see if the excluding observations containing `unsp` values made our models any more accurate. In order to make this across-set comparison, we estimate an intercept-only model on the full data set and use the resulting RMSE as a way to measure how much better the subset performs. In essence, we want to answer the question: Does the predictive power of the model increase when we eliminate 'fuzzy' observations?

We now use the ratio:

$$\frac{RMSE_0}{RMSE_t},$$

where the numerator is the RMSE from the null model and the denominator is from the model specification we are testing. This will give us an approximation of how much 'better' the current model performs relative to the null. For example, a ratio of 3 implies that the error of the tested model is 3 times smaller than that of the null. Furthermore, it allows us to move away from the interpretation of the RMSE, which is currently calculated with regards to the demeaned log of price. The outcome of the null model is stated below:

```
## [1] "The RMSE of the null model using the full data is: 0.7499"
```

We now present the highest-performing model specification for each method and its associated performance ratio, in decreasing order.

```
## Step:  3.838462 , lprice ~.+I(mileage^3)+I(mileage^2)

## Elastic Net:  3.825327 , lprice ~.+I(mileage^3)+I(mileage^2)

## Lasso:  3.825595 , lprice ~.+I(mileage^3)+I(mileage^2)

## Ridge:  2.994859 , lprice ~.^2
```

The stepwise-selection method outperformed all forms of the penalized regression we tested, but by a very slim margin. We would be hesitant to conclude that one particular method is outweighed by another from these results. A broader set of formulas could potentially reveal differences among the methods. The ridge model clearly underperformed relative to the rest of the bunch.

Finally, we present the performance of the subsetted data models in the same fashion. (The order is the same as above for ease of comparison)

```
## Step:  4.6248 , lprice ~.+mileage:trim+mileage:displacement
```

```
## Elastic Net:  4.638755 , lprice ~.^2
```

```
## Lasso:  4.642935 , lprice ~.^2
```

```
## Ridge:  3.787752 , lprice ~.^2
```

Interestingly, the results from this data set are quite different both in terms of model specification and test error results. The ridge still performs the worst, but its performance is pretty much on par with the results of the full data set. As for the other three methods, all selected model specifications were different from the previous iteration and their relative performance has increased considerably. Based on this evidence, we think there is reason to believe that the `unsp` values added in a significant amount of noise to the full data, making models based on that set less reliable.

Final Notes

- A lot has been excluded from this report; pretty much all the code has been left out because we felt that it wouldn't be particularly helpful for the reader. It also would have likely doubled the length of this report.
- There is still much that could have been in terms of testing different methods; it would have been much smarter to write an algorithm that would update an existing formula instead of having to specify each individual formula
- We could be more clear on what exactly the best-performing models turned out to be. As it stands, we omit that entirely and never specify the actual coefficients of the model. We felt that it wasn't crucial to state them explicitly, but it probably would have been interesting to report.
- Finally, we're not entirely sure if the relative performance ratio was the best metric to convey our results. It did allow us to ignore converting the RMSE back to USD and the interpretation was pretty simple, but it may not have been the best way of going about it.