Visual selection of variables vs. Lasso model

When working data for a prediction model we can chose one of two paths, reducing the number of variables for the most important to simplify and better explain the model, or work with all variables in order to take the advantage of all the data.

The third option is also present, in the form of elastic net for example, however we will keep our interest in linear regression and only working with all or subset of variables to obtain a more interpretable model.

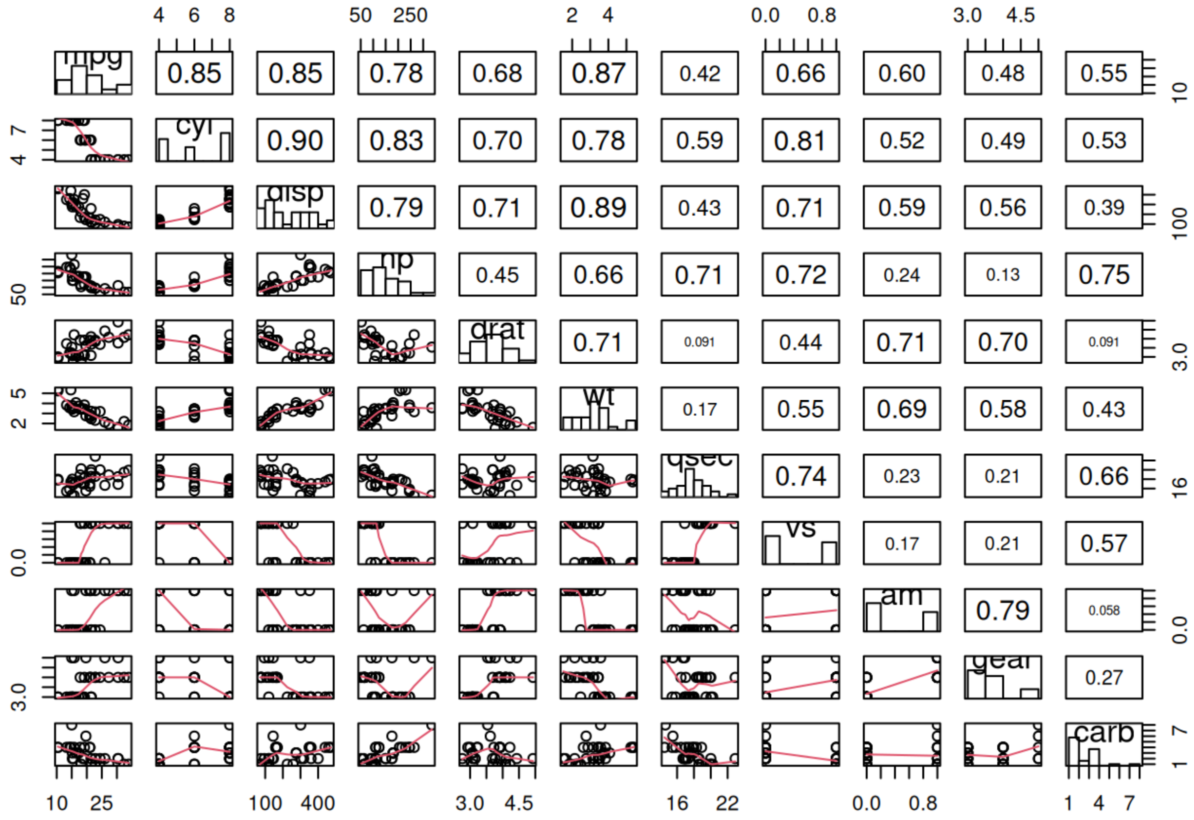
We'll check to examples with standard data-sets.

Example 1

Visualizing mtcars variables to predict mpg

mtcars dataset of R is described as: '…extracted from the 1974 *Motor Trend* US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).'

Visualizing the variables correlation with the output variable gives a good idea of the importance of them for the model, indicating the direction and strength of the relation, for example in the case of 'mtcars' data-set in an exercise to predict a car's mpg (Miles per gallon).



Visually it seems that gear (number of forward gears), qsec (1/4 mile time) and carbs (number of carburetors) can be discarded of the final model as the correlation to mpg is similar or lower than average.

And definitely some variables can be discarded to reduce data and simplify the model due to their low correlation: vs (Engine (0 = V-shaped, 1 = straight)) and am (Transmission (0 = automatic, 1 = manual)).

What combination would you choose?

I suggest make a note of your decisions in this matter and then…

Let’s see what does Lasso suggest.

The Lasso for mtcars

According to the creators of Lasso: ‘models generated from the lasso are generally much easier to interpret than those produced by ridge regression. We say that the lasso yields sparse models—that is, sparse models that involve only a subset of the variables.’. For more information check: https://link.springer.com/book/10.1007/978-1-0716-1418-1

Let’s go directly to model and check the coefficients obtained for the different variables, coefficients of zero indicate not to use the indicated variable.

```r
mtcars <- mtcars
factors <- c('cyl', 'vs', 'am', 'gear', 'carb')
for (var in factors) {
  mtcars[ , var] <- as.factor(mtcars[ , var])
}
x <- model.matrix(mpg ~ ., mtcars) [ ,-1]
y <- mtcars$mpg

grid <- 10^seq(10, -2, length = 100)
```

```r
train <- sample(1:nrow(x), nrow(x) / 2)
test <- (-train)
y.test <- y[test]
```

```r
model <- glmnet(x[train,], y[train], alpha = 1, lambda = grid)
```

```r
out <- glmnet(x, y, alpha = 1, lambda = grid)
lasso.coef <- predict(out, type = 'coefficients', s = bestlamda) [1:17, ]
lasso.coef
```

```
(Intercept)        cyl6        cyl8        disp          hp        drat 
28.82548961 -0.32424570  0.00000000  0.00000000 -0.02571643  0.78336859 
         wt        qsec         vs1         am1       gear4       gear5 
-2.59609757  0.00000000  0.91627981  0.97414573  0.00000000  0.00000000 
      carb2       carb3       carb4       carb6       carb8 
```

```
 0.00000000  0.00000000 -0.49093011  0.00000000  0.00000000
```

The model discards variables for 8cyls, disp, qsec, gear4, carb2, carb3, carb6 and carb8, a rather different decision to the initial visual impression.

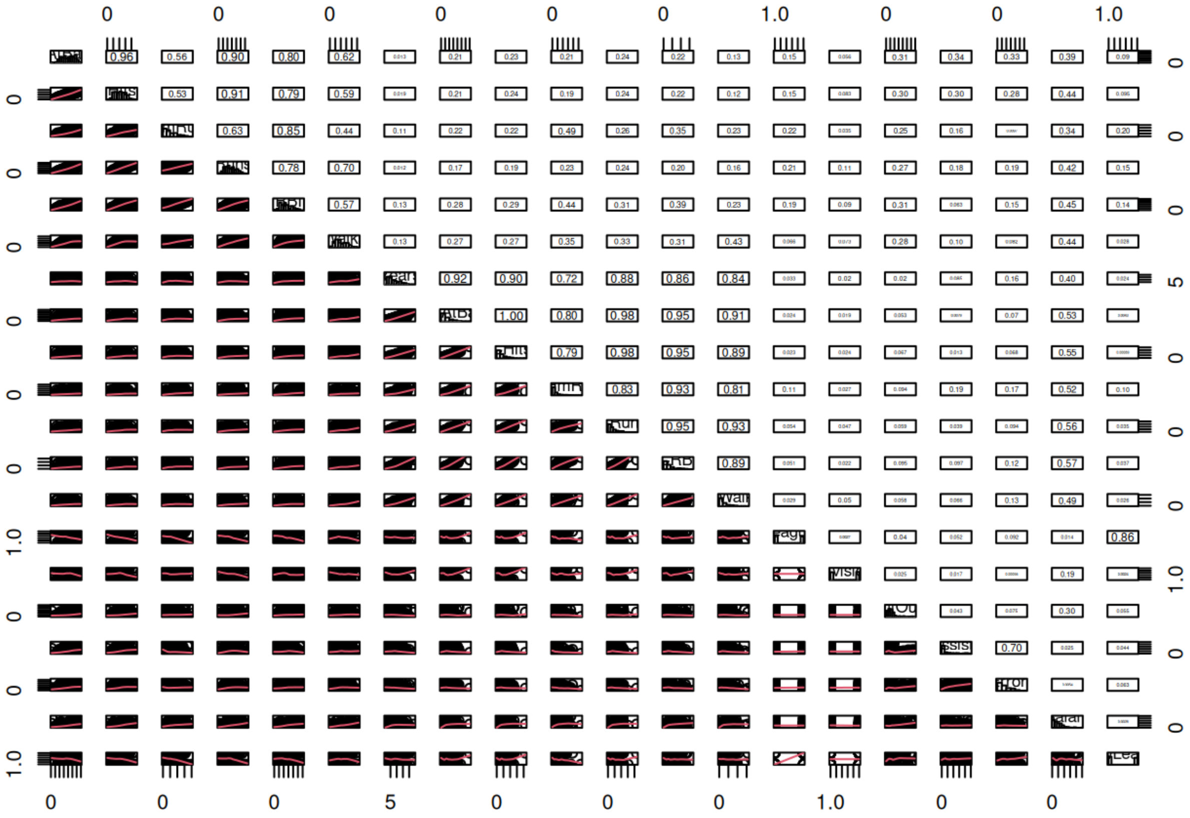
Note that the first run was made without factorizing cyl, vs, am, gear and carb variables and the model discarded variables disp (displacement), qsec, vs and gear but was unable to distinguish between different sub-factors.

So we could work a model with a decent accuracy using six variables instead of 9.

Example 2

Visualizing Hitters variables to predict Salary

Hitters data-set can be found in the ISLR2 library, ‘Major League Baseball Data from the 1986 and 1987 seasons.’ It has 20 variables as AtBat, Hits, Runs, Walks and more, the exercise is to predict the Salary of a player given this and other parameters of its carer.



Now the number of variables makes even reading the information a challenge, some of the variables are clearly too low correlated to be considered good candidates to discard from a final model and thus obtain a simpler final version.

Years, CAtBat (Number of times at bat during his career), CHits (Number of hits during his career), CHmRun (Number of home runs during his career), CWalks (Number of walks during his career), CHmRun (Number of home runs during his career), CWalks (Number of walks during his career), League (A factor with levels `A` and `N` indicating player's league at the end of 1986), Division, NewLeague.

What combination would you choose?

I suggest make a note of your decisions in this matter and then…

Let's see what does Lasso suggest.

The Lasso for Hitters

```
x <- model.matrix(Salary ~ ., Hitters) [ ,-1]
y <- Hitters$Salary

grid <- 10^seq(10, -2, length = 100)
```

```
train <- sample(1:nrow(x), nrow(x) / 2)
test <- (-train)
y.test <- y[test]
```

```
model <- glmnet(x[train,], y[train], alpha = 1, lambda = grid)
```

```
out <- glmnet(x, y, alpha = 1, lambda = grid)
lasso.coef <- predict(out, type = 'coefficients', s = bestlamda) [1:19, ]
lasso.coef
```

```
 (Intercept)        AtBat         Hits        HmRun         Runs          RBI 
 113.4801605   -1.4369816    5.4221983    0.0000000    0.0000000    0.0000000 
       Walks        Years       CAtBat        CHits       CHmRun        CRuns 
   4.5284815   -8.8178168    0.0000000    0.0000000    0.4705686    0.6201586 
        CRBI       CWalks      LeagueN    DivisionW      PutOuts      Assists 
   0.3951242   -0.4820820   31.3887001 -119.1611146    0.2693146    0.1516149 
      Errors 
  -1.8589716 
```

We see the coeficient with value 0, meaning not important for the model in HmRun, Runs, RBI (Runs batted), CAtBat (Times at bat) and CHits (Number of hits).