

Lab06

Channing_503128649

20/09/2021

Import

1. Produce a tidier file automobile.csv

```
o <- read.csv("automobile-original.csv", na.strings = "?")
o <- na.omit(o)
o <- subset(o, select = -c(engine.location))
write.csv(o, "automobile.csv", row.names = FALSE)

d1 <- read.csv("automobile.csv")
d2 <- read.csv("automobile-subset.csv")
all(d1 == d2)
```

```
## [1] TRUE
```

Explore

2. Write R code to answer the following questions:

What is the mean price (price) of all vehicles?

```
mean(d1$price)
```

```
## [1] 11445.73
```

How many vehicles have 4 doors (num.of.doors)?

```
nrow(d1[d1$num.of.doors == "four",])
```

```
## [1] 95
```

What are the different engine types (engine.type) among the observations?

```
unique(d1$engine.type)
```

```
## [1] "ohc" "l" "dohc" "ohcv" "ohcf"
```

How many vehicles have a price (price) higher than \$20000?

```
nrow(d1[d1$price > 20000,])
```

```
## [1] 13
```

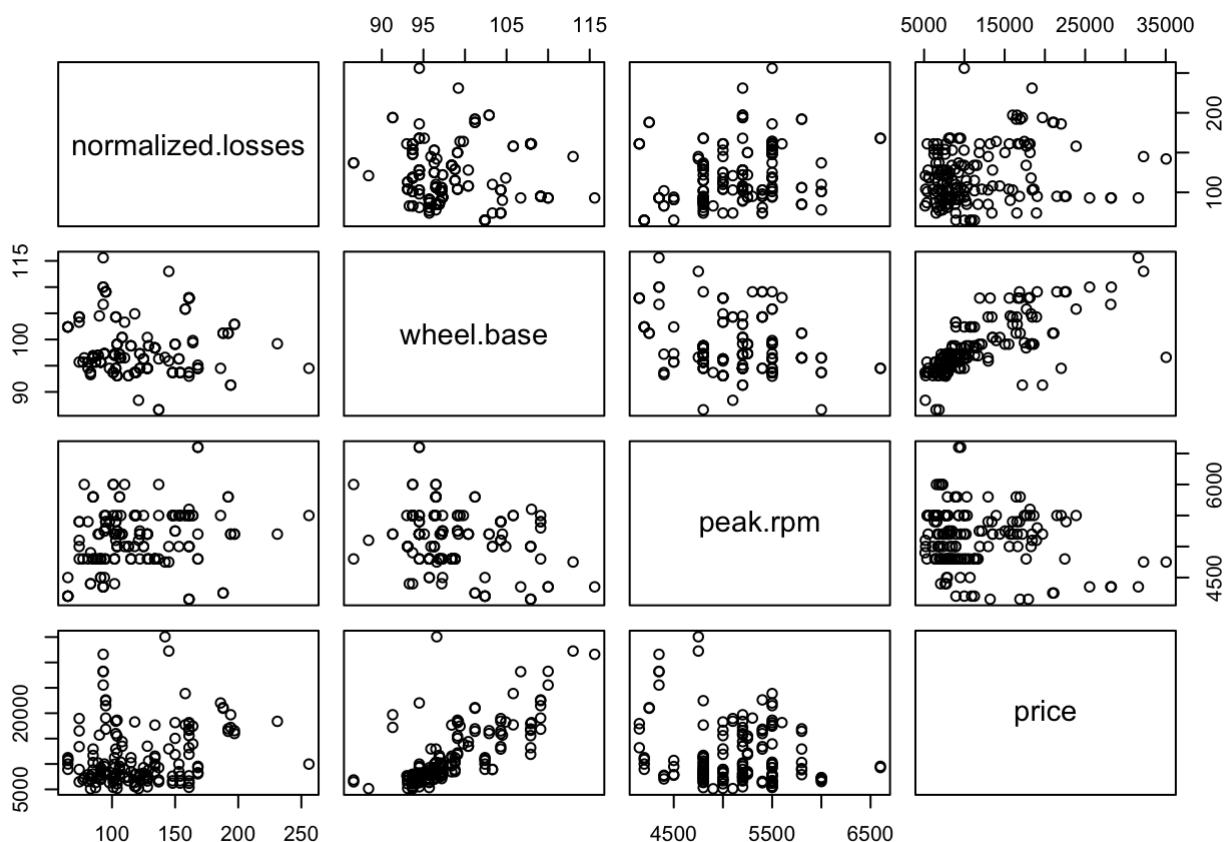
What is the mean price (price) for “4wd” (drive.wheels)?

```
mean(d1[d1$drive.wheels == "4wd",]$price)
```

```
## [1] 10241
```

3. Produce pairwise scatterplots between variables `normalized.losses`, `wheel.base`, `peak.rpm` and `price`

```
pairs(~normalized.losses + wheel.base + peak.rpm + price, data=d1)
```



Linear regression

For all of the following regression questions, we use the price (`price`) as the response variable.

4. Produce the full linear regression model with all variables included. Comment on the outcome.

```
fit <- lm(price~., d1)
summary(fit)
```

```
##
## Call:
## lm(formula = price ~ ., data = d1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2949.8  -587.9    0.0   650.3  2259.2
##
## Coefficients: (3 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.309e+04  1.560e+04   1.480 0.141731
## symboling      -5.068e+00  2.163e+02  -0.023 0.981351
## normalized.losses  5.577e+00  6.252e+00   0.892 0.374295
## makebmw        3.596e+02  1.791e+03   0.201 0.841251
## makechevrolet  -4.745e+03  1.856e+03  -2.556 0.011985 *
## makedodge      -6.209e+03  1.613e+03  -3.851 0.000201 ***
## makehonda      -1.583e+03  1.739e+03  -0.910 0.364854
## makejaguar      2.431e+03  2.918e+03   0.833 0.406615
## makemazda      -4.063e+03  1.568e+03  -2.591 0.010906 *
## makemercedes-benz  2.548e+03  1.608e+03   1.584 0.116050
## makemitsubishi -6.328e+03  1.578e+03  -4.010 0.000113 ***
## makenissan      -3.690e+03  1.470e+03  -2.510 0.013557 *
## makepeugot     -5.143e+03  4.732e+03  -1.087 0.279561
## makeplymouth   -6.025e+03  1.618e+03  -3.723 0.000316 ***
## makeporsche     4.830e+03  2.091e+03   2.310 0.022810 *
## makesaab       -4.040e+02  1.552e+03  -0.260 0.795107
## makesubaru     -7.317e+03  2.301e+03  -3.180 0.001927 **
## maketoyota     -5.869e+03  1.594e+03  -3.682 0.000364 ***
## makevolkswagen -4.297e+03  1.355e+03  -3.170 0.001986 **
## makevolvo      -2.871e+03  1.838e+03  -1.563 0.121105
## fuel.typegas   -1.065e+04  5.139e+03  -2.072 0.040708 *
## aspirationturbo  2.171e+03  5.956e+02   3.646 0.000413 ***
## num.of.doorstwo -8.381e+02  3.795e+02  -2.208 0.029366 *
## body.stylehardtop -5.627e+03  1.375e+03  -4.091 8.33e-05 ***
## body.stylehatchback -5.736e+03  1.343e+03  -4.271 4.23e-05 ***
## body.stylesedan  -5.702e+03  1.398e+03  -4.080 8.71e-05 ***
## body.stylewagon  -5.647e+03  1.414e+03  -3.995 0.000119 ***
## drive.wheelsfwd  -2.936e+01  6.501e+02  -0.045 0.964064
## drive.wheelsrwd  1.977e+03  9.553e+02   2.070 0.040895 *
## wheel.base      3.184e+02  8.310e+01   3.832 0.000215 ***
## length         -7.663e+01  3.808e+01  -2.012 0.046700 *
## width           2.437e+02  2.048e+02   1.190 0.236673
## height          -3.352e+02  1.196e+02  -2.802 0.006035 **
## curb.weight      5.208e+00  1.297e+00   4.016 0.000110 ***
## engine.type1     -4.677e+03  3.668e+03  -1.275 0.205003
## engine.typeohc   -1.913e+03  9.958e+02  -1.921 0.057359 .
## engine.typeohcf      NA          NA          NA          NA
## engine.typeohcv   -1.337e+03  1.161e+03  -1.152 0.251949
## num.of.cylindersfive -4.108e+03  2.559e+03  -1.606 0.111308
## num.of.cylindersfour -4.688e+03  3.242e+03  -1.446 0.151067
## num.of.cylinderssix -2.976e+03  2.894e+03  -1.028 0.306105
## num.of.cylindersthree  NA          NA          NA          NA
## engine.size      -1.244e+01  2.356e+01  -0.528 0.598678
## fuel.system2bbl   2.070e+03  1.018e+03   2.032 0.044587 *
## fuel.systemidi      NA          NA          NA          NA
```

```
## fuel.systemmfi          3.468e+03  1.959e+03   1.770 0.079599 .
## fuel.systemmpfi        2.602e+03  1.081e+03   2.407 0.017808 *
## fuel.systemspdi        1.081e+03  1.292e+03   0.837 0.404621
## bore                   -8.817e+02  1.427e+03  -0.618 0.538041
## stroke                  -5.677e+02  9.524e+02  -0.596 0.552409
## compression.ratio      -7.000e+02  3.844e+02  -1.821 0.071380 .
## horsepower             -2.019e+01  1.911e+01  -1.057 0.292973
## peak.rpm                -5.377e-01  5.658e-01  -0.950 0.344070
## city.mpg                -1.564e+02  1.030e+02  -1.518 0.131872
## highway.mpg             1.284e+02  8.912e+01   1.441 0.152517
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1165 on 107 degrees of freedom
## Multiple R-squared:  0.9734, Adjusted R-squared:  0.9607
## F-statistic: 76.78 on 51 and 107 DF,  p-value: < 2.2e-16
```

There are too many variables in this linear model, where some of variables are significant but could be highly correlated. Also, 3 coefficients are not available.

5. Remove any variable(s) that seem to cause the linear regression to fail, i.e., some coefficients may become NA. Repeat this until you can produce a meaningful “full” linear regression model (it is okay if you remove slightly more variables than necessary).

How many variables are deemed to be significant by the t-tests (with a p-value less than 0.05)?

```
fit <- lm(price ~ . - fuel.system - num.of.cylinders - engine.type, d1)
coef <- summary(fit)$coefficients
table(coef[,4]<0.05)
```

```
##
## FALSE  TRUE
##      18    24
```

There're 24 significant variables.

6. Apply your “full” linear regression model to the data and compute the resulting mean squared error (MSE).

```
mse = function(y1, y2) mean( (y1 - y2)^2 )
mse(predict(fit, d1), d1$price)
```

```
## [1] 1107500
```

Subset selection

```
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

7. For the data (using your “full” set of variables), produce a subset linear regression model, using the backward selection and the AIC.

```
r.bwd = regsubsets(price ~ . - fuel.system - num.of.cylinders - engine.type, nvmax =
38, data=d1, method="backward")
r.s <- summary(r.bwd)
bic <- r.s$bic
aic <- bic - (log(nrow(d1)) - 2) * (38:1)
j = which.min(aic)
beta = coef(r.bwd, j)
beta
```

```
##      (Intercept)      makechevrolet      makedodge      makehonda
##      17908.63085      -5980.10911      -7615.82056      -5505.11491
##      makemazda      makemitsubishi      makenissan      makepeugot
##      -4507.70019      -8073.20659      -4907.11451      -8360.61669
##      makeplymouth      makesubaru      maketoyota      makevolkswagen
##      -7520.61552      -6636.69665      -6798.15202      -5215.54416
##      makevolvo      aspirationturbo      body.stylehardtop      body.stylehatchback
##      -4490.72017      1625.24581      -7088.22297      -6209.99436
##      body.stylesedan      body.stylewagon      drive.wheelsrwd      wheel.base
##      -5516.20648      -5293.49476      1471.33885      380.48283
##      length      height      curb.weight
##      -138.78747      -519.48153      7.50285
```

8. Apply the AIC-selected model to the data and compute the resulting MSE.

```
d1.matrix <- model.matrix(fit)
yhat = drop(d1.matrix[,names(beta)] %*% beta)
mse(yhat, d1$price)
```

```
## [1] 1434429
```

9. Create a plot that shows the predictions of your AIC-selected model against the response variable (price), using different colors for different levels of drive.wheels

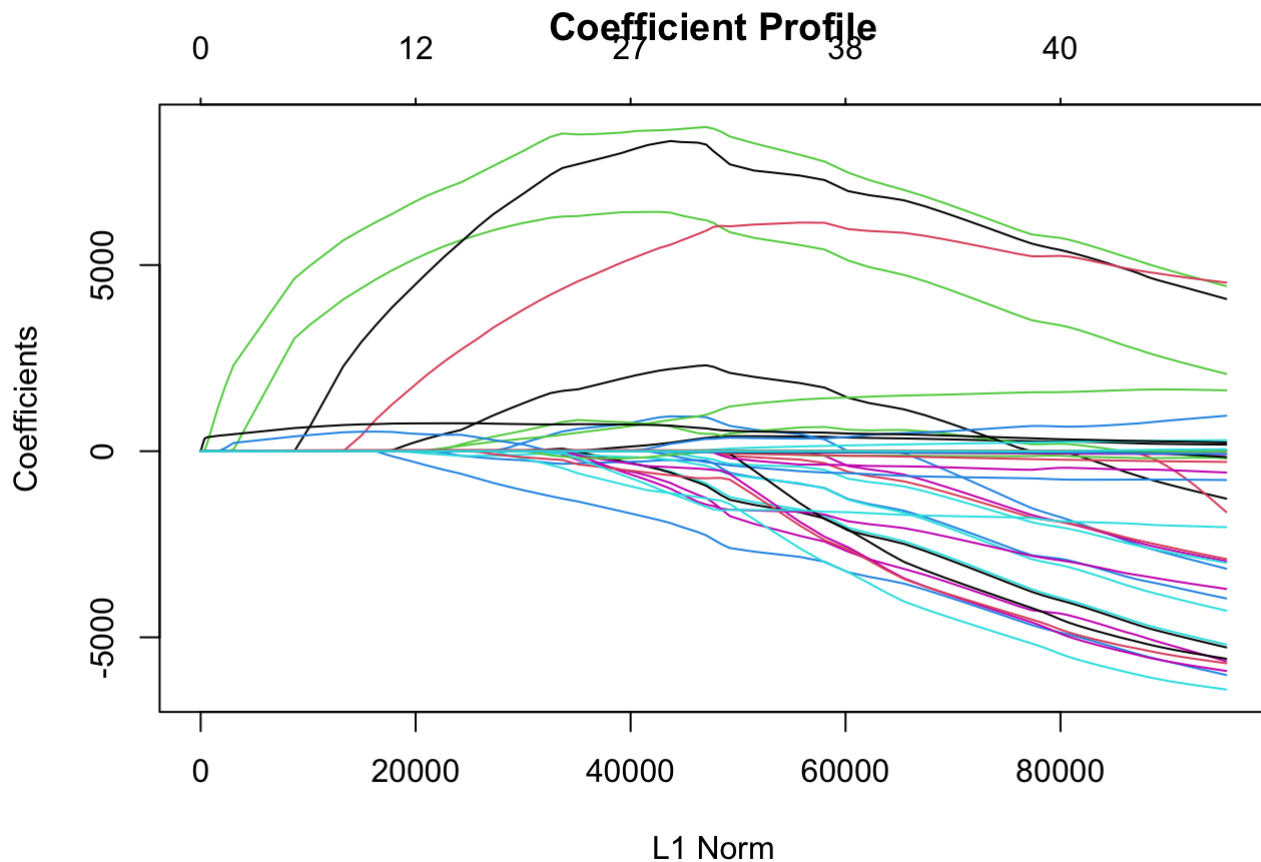
LASSO

10. For the data (using your “full” set of variables), compute the Lasso model.

```
x <- d1.matrix[,-1]
y <- d1$price
r.lasso = glmnet(x, y, alpha=1)
```

11. Create a coefficient profile plot of the coefficient paths that varies with the value of λ (or $\log(\lambda)$).

```
plot(r.lasso, main="Coefficient Profile")
```



(@) Choose 5 different λ -values within a seemingly reasonable range (with roughly 5 to 30 variables included) and compute the MSEs of the corresponding 5 Lasso subset models. Write R code to find out how many variables (excluding the intercept) are included in each Lasso subset model.

Summary

In this lab, we went through the data science process with a focus on the model with lots of variable. We used two main techniques to choose the best submodel, model selection criteria and regularisation.

There're 2 main criteria, namely, AIC and BIC, they based the maximum likelihood with the only different on the penalty terms. AIC tends to preserve the variables while BIC penalise heavily on the numbers of variables with the growth of observations.

The regularisation approaches shrink the coefficients instead. To that, Lasso can do it more efficient by shrinking the coefficients to exactly 0.