Model Selection with BMA

Grant Esparza

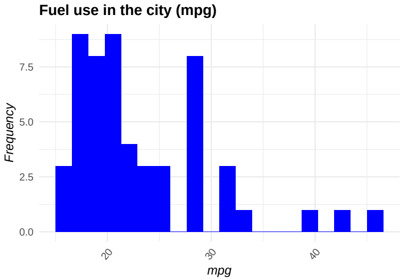
December 19, 2018

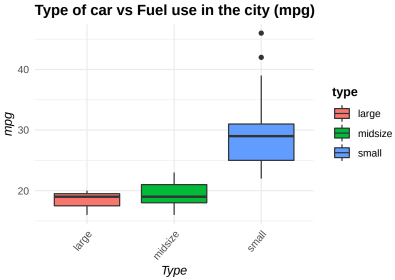
Introduction

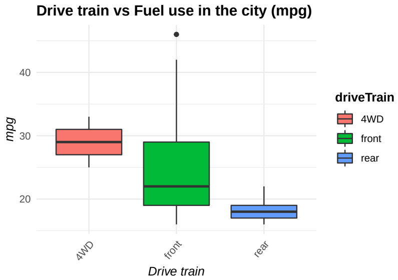
Picking models and ensuring that you end up using the right predictors can be a difficult task. Bayesian Model Averaging is a method that can be used to conduct Bayesian regression which is similar to linear regression however with some exceptions.

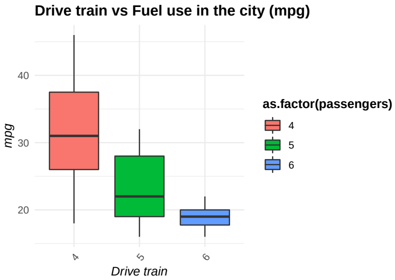
We'll go through the model building process for both linear regression and Bayesian regression and see which produces the better model at predicting mpg.

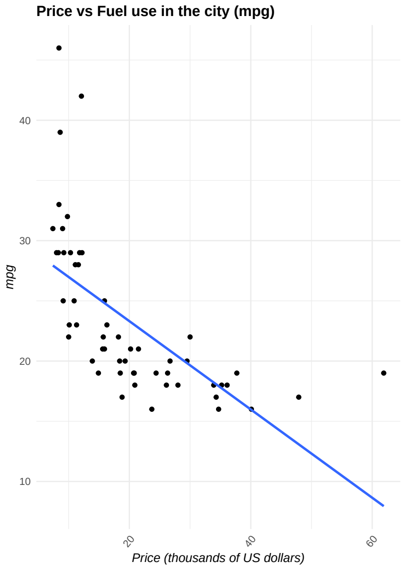
Plots

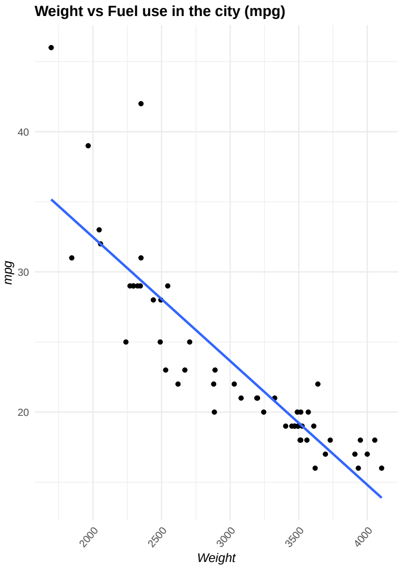

Fuel use in the city (mpg)
Frequency
7.5
5.0
2.5
0.0
20
30
40
mpg


Type of car vs Fuel use in the city (mpg)
mpg
40
30
20
large
midsize
small
Type
type
large
midsize
small


Drive train vs Fuel use in the city (mpg)
mpg
40
30
20
4WD
front
rear
Drive train
driveTrain
4WD
front
rear


Drive train vs Fuel use in the city (mpg)
mpg
40
30
20
4
5
6
Drive train
as.factor(passengers)
4
5
6


Price vs Fuel use in the city (mpg)
mpg
40
30
20
10
20
40
60
Price (thousands of US dollars)


Weight vs Fuel use in the city (mpg)
mpg
40
30
20
2000
2500
3000
3500
4000
Weight

Multiple Linear Regression

```
##
## Call:
## lm(formula = mpgCity ~ type + price + driveTrain + passengers +
##     weight, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0730 -0.8915  0.0308  1.0116 10.9097
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     65.254607   7.690710   8.485 5.76e-11 ***
## typemidsize     -3.570184   1.473808  -2.422   0.0194 *
## typesmall       -4.075422   2.571819  -1.585   0.1199
## price            0.038124   0.060002   0.635   0.5283
## driveTrainfront  1.716006   2.248698   0.763   0.4493
## driveTrainrear   3.272107   2.699716   1.212   0.2317
## passengers      -2.207348   0.981565  -2.249   0.0294 *
## weight          -0.009973   0.001947  -5.123 5.81e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.02 on 46 degrees of freedom
## Multiple R-squared:  0.8197, Adjusted R-squared:  0.7922
## F-statistic: 29.87 on 7 and 46 DF,  p-value: 4.43e-15
```

```
##
## Call:
## lm(formula = mpgCity ~ type + driveTrain + passengers + weight,
##     data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1143 -0.8901 -0.0432  0.8496 10.9229
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     64.519655   7.554842   8.540 4.03e-11 ***
## typemidsize     -3.341317   1.420019  -2.353   0.0229 *
## typesmall       -3.916735   2.543375  -1.540   0.1303
## driveTrainfront  1.718977   2.234383   0.769   0.4455
## driveTrainrear   3.366905   2.678436   1.257   0.2149
## passengers      -2.272014   0.970061  -2.342   0.0235 *
## weight          -0.009428   0.001737  -5.429 1.95e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3 on 47 degrees of freedom
## Multiple R-squared:  0.8181, Adjusted R-squared:  0.7948
## F-statistic: 35.22 on 6 and 47 DF,  p-value: 8.625e-16
```

```
##
## Call:
## lm(formula = mpgCity ~ type + passengers + weight, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1212 -1.1376  0.0124  0.9672 10.7977
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 65.280739   7.298025   8.945 7.12e-12 ***
## typemidsize -3.595922   1.392009  -2.583  0.01282 *
## typesmall   -3.749750   2.528120  -1.483  0.14442
## passengers  -2.674123   0.884711  -3.023  0.00398 **
## weight      -0.008354   0.001459  -5.725 6.18e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.995 on 49 degrees of freedom
## Multiple R-squared:  0.811,  Adjusted R-squared:  0.7955
## F-statistic: 52.56 on 4 and 49 DF,  p-value: < 2.2e-16
```

Removing more variables results in a smaller AdjR^2

Bayesian Regression

Bayesian regression is similar to linear regression but it has the benefit of supplying a *prior* distribution to the coefficents. By using the *posterior*, the conditional distribution of the weights given a dataset, we can update our prior for another iteration.

Using the package `BMA`, we can sample from out dataset to generate inclusion probabilities for each of the coefficents in our model. This process will help us select a model with coefficents that are most likely to be in the "true" model.

Bayesian Model Averaging

```r
car_bays <- BAS::bas.lm(mpgCity ~., data=cars, method="MCMC", prior="ZS-null",
                        modelprior=uniform())
```

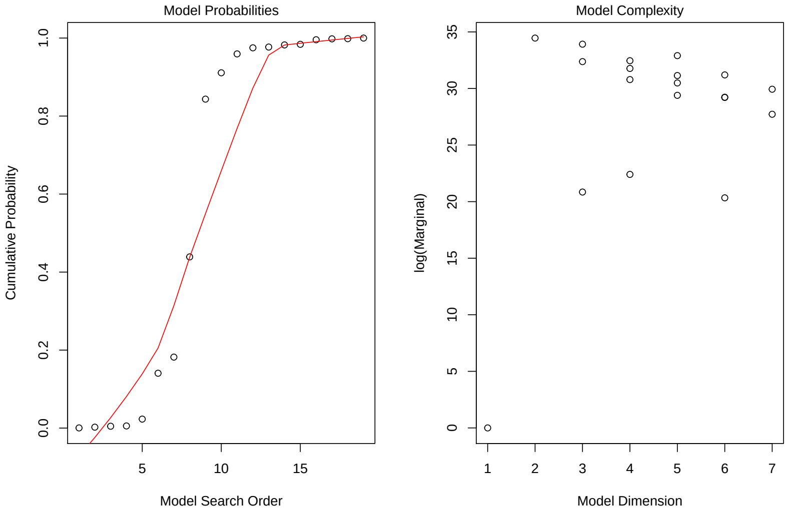
`method` - Sampling method to use for Bayesian Model Averaging. `MCMC` samples with replacement using the Markov chain Monte Carlo algorithm

`prior` - Prior distribution for regression coefficents. `ZS-null` uses the Cauchy distribution

`modelprior` - Family of prior distribution on the models. `uniform()` assigns equal probabilities to all models

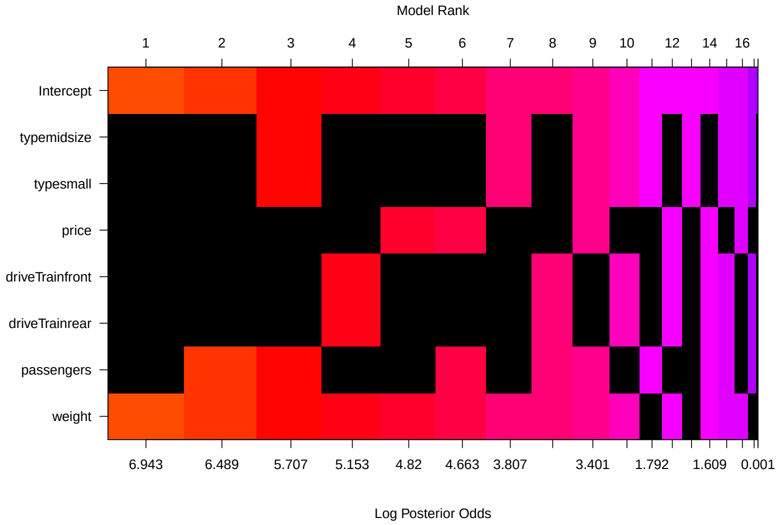
```
##                  P(B != 0 | Y)  model 1    model 2    model 3    model 4
## Intercept           1.00000000  1.00000  1.0000000  1.0000000  1.0000000
## typemidsize         0.16093750  0.00000  0.0000000  1.0000000  0.0000000
## typesmall           0.16093750  0.00000  0.0000000  1.0000000  0.0000000
## price               0.10781250  0.00000  0.0000000  0.0000000  0.0000000
## driveTrainfront     0.09570312  0.00000  0.0000000  0.0000000  1.0000000
## driveTrainrear      0.09570312  0.00000  0.0000000  0.0000000  1.0000000
## passengers          0.45000000  0.00000  1.0000000  1.0000000  0.0000000
## weight              0.99492187  1.00000  1.0000000  1.0000000  1.0000000
## BF                          NA  1.00000  0.5826445  0.2116529  0.1352449
## PostProbs                   NA  0.40450  0.2569000  0.1175000  0.0676000
## R2                          NA  0.76900  0.7845000  0.8110000  0.7905000
## dim                         NA  2.00000  3.0000000  5.0000000  4.0000000
## logmarg                     NA 34.45492 33.9147434 32.9021138 32.4542534
##                   model 5
## Intercept       1.0000000
## typemidsize     0.0000000
## typesmall       0.0000000
## price           1.0000000
## driveTrainfront 0.0000000
## driveTrainrear  0.0000000
## passengers      0.0000000
## weight          1.0000000
## BF              0.1247244
## PostProbs       0.0484000
## R2              0.7708000
## dim             3.0000000
## logmarg        32.3732729
```

Looking at the model



Model Ranking

```
image(car_bays, rotate=F)
```



Predictions

Let's try to predict the mpg for a 1995 Ford F-150 with front wheel drive. The actual city mpg is **15 mpg**.

```
linear.pred[1]
```

```
## [1] 26.24022
```

```
bay.pred$Ybma
```

```
##          [,1]
## [1,] 17.79937
```

Conclusion

We were able to create both linear regression and Bayesian models that aimed to predict the mpg consumed in the city. While we settled for a model with four predictors for the linear model, the Bayesian Model Averaging performed on our data decided the best model was the variable that only used weight as a predictor.