Bayesian learning and Montecarlo Simulation - final project

Valentina Abbattista
Jana El Khoury
Ossama El Oukili
Matteo Regge

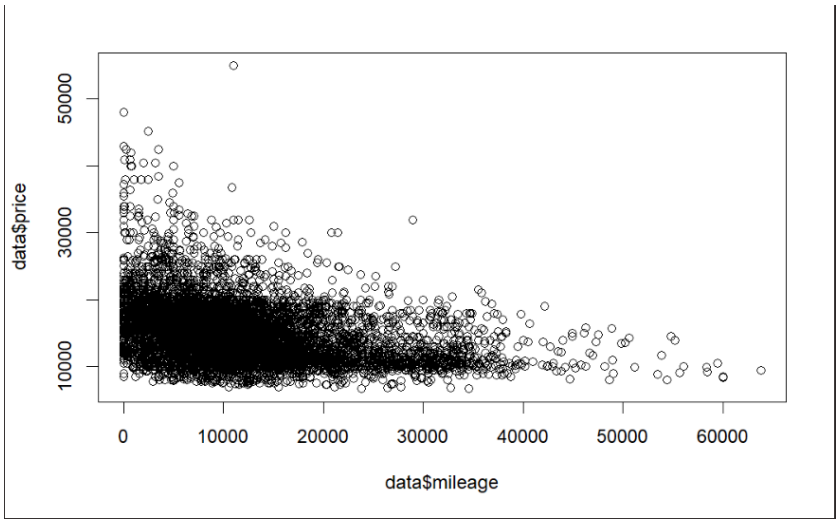
July 2022

1 Introduction

We start by reading the given dataset. The covariates are:

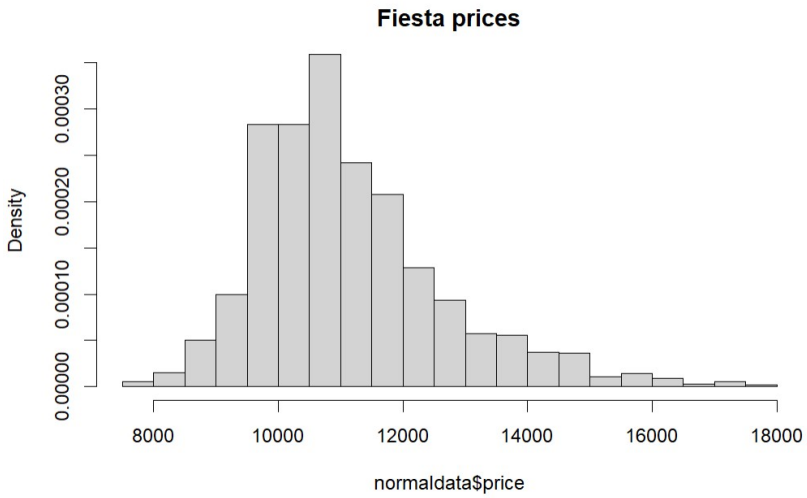
- Car model *(categorical variables)*
- Transmission *(categorical variables)*
- Fuel type *(categorical variables)*
- Year of production
- Mileage
- Miles per gallon
- Price
- Engine size
- Tax

Our goal is to predict the price of a Ford car from the variables present in the dataset. We import the data from the "ford.txt" file, set up the categorical variables as factors and print a summary of the covariates while plotting them. On the Y Axis we have the Price label while on the X Axis we have each one of the variables. This gives us a general understanding of the relevancy of the variables and their values.



We can make a few observations from what we see: more recent cars are consistently more expensive, manual shift tends to be cheaper than semi-automatic and automatic, which are priced similarly, lower mileage means higher prices and hybrid cars generally cost more than diesel or petrol ones.

Then we plot an histogram of the prices, taking the 2018 Fiesta cars as an example because this is the most common car type of the dataset. Observing the plot we can assume that the prices are normally distributed.



Now we proceed by pre-processing and normalizing our data. We reckon that these few steps may yield good results when using some of the variables of our dataset. We start by removing entries that have **engineSize** = 0 or **tax** = 0 since they appear to be mistakes.
Then we normalize by:

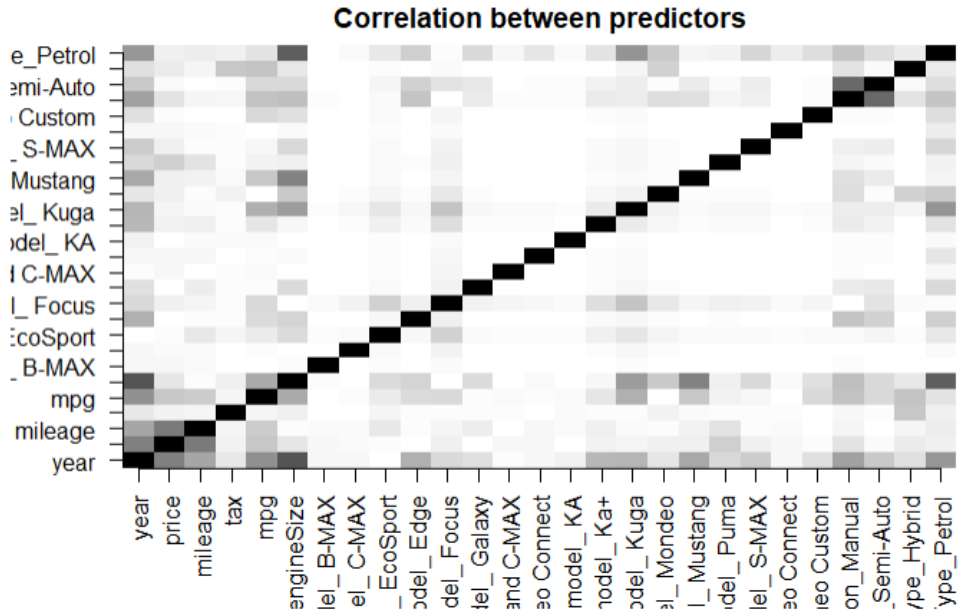
- Computing **Year** = **Year** - 2018, since year values start from 2018
- Computing **Mileage** = **Mileage**/10000
- Computing **Tax** = **Tax**/100
- Computing **Mpg** = **Mpg**/100

We will better clarify the role of normalization later on in the report.
Lastly, we need to remember that we have some categorical variables, thus we need to use dummy variables to work properly: we will have 3 dummy variables for **transmission**, 3 dummy variables for **fuelType** and 18 dummy variables for **model**, for a total of 30 covariates (including **price**).

Now we use the **relevel** function to take the Fiesta Model as reference for the various models. This is important since in the next phase we will use **BAS** Library, we will notice that three categorical variables will be missing, this happens because **BAS** automatically takes one of each categorical variable as reference for all the others and insert them in the Intercept calculation.
This is done automatically in alphabetical order, so the reference model will be **B-MAX**. Looking at the dataset though, we notice that there are few **B-MAX models** , therefore we'd rather take the **Fiesta model** as reference since it is the most common one. This will also best reduce the confidence intervals of the beta distributions.

Before starting we can make some considerations about our data and the covariates. We can verify if there are some linear dependencies plotting an image that represents the correlation between the predictors. In particular, we see that there is correlation between **year**, **price** and **mileage** because the relative squares are darker than the others. In the next section we will also analyze the multi-collinearity between predictors.

Lastly we separate our dataset in two parts: the training set will be used to fit the model, while the test set will be used for the prediction.
The split is 80-20 and is computed randomically (seed 123 for replicability).



2 Prior selection

First, we start fitting a frequentist model. This is done just for comparison and for some additional analysis. This is the summary of the model:

```
Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               18100.78    1818.34   9.955  < 2e-16 ***
model B-MAX                -585.91     611.44  -0.958 0.337976
model C-MAX                  19.17     260.90   0.073 0.941421
model EcoSport             1739.30      93.14  18.673  < 2e-16 ***
model Edge                10423.57     246.49  42.288  < 2e-16 ***
model Focus                2392.12      60.00  39.866  < 2e-16 ***
model Galaxy               5661.43     280.48  20.185  < 2e-16 ***
model Grand C-MAX          1230.57     345.15   3.565 0.000366 ***
model Grand Tourneo Connect  2689.11     406.90   6.609 4.21e-11 ***
model KA                  -4439.87     500.52  -8.870  < 2e-16 ***
model Ka+                 -4374.50     115.32 -37.932  < 2e-16 ***
model Kuga                 2783.27     112.98  24.635  < 2e-16 ***
model Mondeo               1056.04     186.80   5.653 1.65e-08 ***
model Mustang              5750.25     446.67  12.874  < 2e-16 ***
model Puma                 5845.45     230.52  25.357  < 2e-16 ***
model S-MAX                7433.77     258.11  28.801  < 2e-16 ***
model Tourneo Connect      3027.00     583.90   5.184 2.24e-07 ***
model Tourneo Custom       5768.26     343.23  16.806  < 2e-16 ***
year                       2071.77      49.41  41.930  < 2e-16 ***
transmissionManual         -989.81      87.60 -11.299  < 2e-16 ***
transmissionSemi-Auto       503.41     125.60   4.008 6.20e-05 ***
mileage                    -845.51      30.89 -27.369  < 2e-16 ***
fuelTypeHybrid             6608.67     583.17  11.332  < 2e-16 ***
fuelTypePetrol              953.25     115.40   8.260  < 2e-16 ***
tax                       -5303.98    1221.49  -4.342 1.43e-05 ***
mpg                       -3672.31     364.54 -10.074  < 2e-16 ***
engineSize                 4361.33     114.98  37.931  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1725 on 5924 degrees of freedom
Multiple R-squared:  0.8562,    Adjusted R-squared:  0.8556
F-statistic:  1357 on 26 and 5924 DF,  p-value: < 2.2e-16
```

We can make some considerations about the estimate values of β posterior related to our data:

- All the numerical variables seem to be relevant for the computation, while some dummy variables seem to be less relevant: in particular, we can notice that the coefficients of **modelB-MAX** and **modelC-MAX** are smaller than the coefficients of the other models.
- Also the p-values are good indicators, for example **modelB-MAX** and **modelC-MAX** have high p-values, thus as said before they are probably not very significant for our model. We will see later if this consideration is correct.
- A negative coefficient means that the related predictor reduces the price of a car. For example,

the higher is the mileage, the lower is the price of the car, therefore mileage has a negative coefficient. On the other hand, a positive coefficient means that the related predictor increases the price of a car: for example, if the car is hybrid it will probably have a high price.

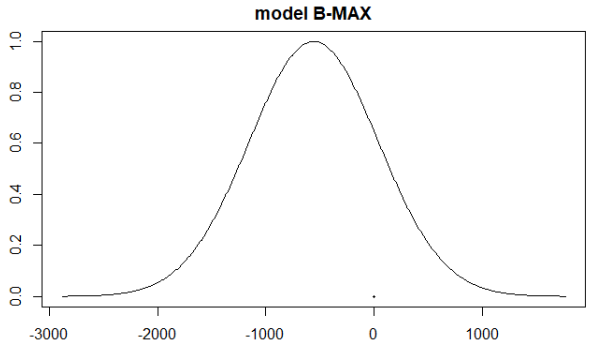
We can analyze the multicollinearity between predictors with the frequentist model. We can suppose that there are some variables that may be dependent, for example we could derive **mpg** knowing **year**, **model**, **fuelType**, **transmission** and **engineSize** and maybe it could be the same for other variables. We use the VIF indicator: if it is greater than 9 or 10 for a predictor it means that the predictor depends on other predictors. In our case we see that the VIF of **model** is greater than 9, so this is an evidence of the presence of multicollinearity. Based on our considerations we can remove **mpg** from our model and the VIF value of **model** will become smaller. But our primary goal is prediction and we now that multicollinearity does not affect prediction, so we will remove it later and for the moment we will continue our analysis; at the end we will do a comparison.

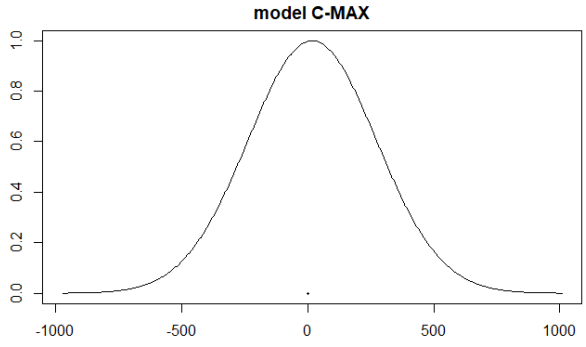
	GVIF	Df	GVIF^(1/(2*Df))
model	9.209738	17	1.067481
year	1.574981	1	1.254982
transmission	1.348305	2	1.077574
mileage	1.464717	1	1.210255
fuelType	5.803498	2	1.552110
tax	1.062916	1	1.030978
mpg	3.032628	1	1.741444
engineSize	5.109022	1	2.260315

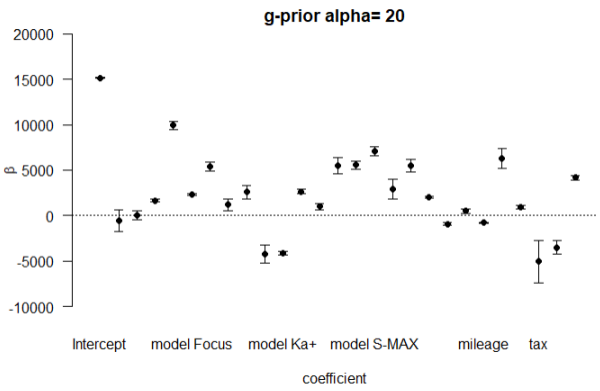
	GVIF	Df	GVIF^(1/(2*Df))
model	5.421546	17	1.050974
year	1.498949	1	1.224316
transmission	1.219601	2	1.050883
mileage	1.459111	1	1.207937
fuelType	2.817102	2	1.295539
tax	1.062902	1	1.030971
engineSize	3.972551	1	1.993126

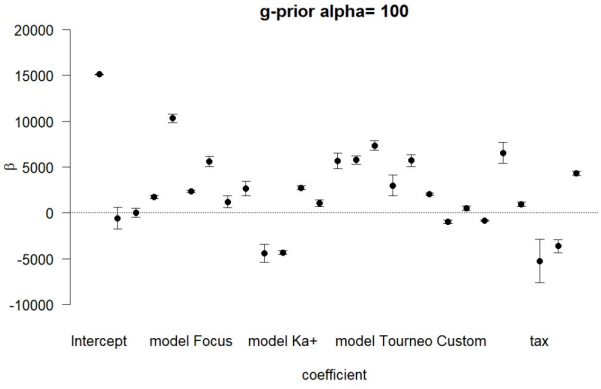
Now we can use a bayesian approach and select a prior. Since we have no prior knowledge over our dataset, we will choose a non-informative prior. In particular we will compare G-prior and Zellner-Siow.

For the G-prior we chose $\alpha = 20$ so the prior will be a bit informative. We tried also with greater values, but even with $\alpha = 100$ the result is quite similar.
If we run the code we can see a summary of the model, the plots of β distributions and the plot of the 95% confidence intervals for β. In particular, we can observe that the posterior means of the β distributions for **modelB-MAX** and **modelC-MAX** are quite close to the zero with respect to the posterior means of the β distributions for the other covariates: this means that probably **modelC-MAX** and **modelB-MAX** are not relevant for our model, as we supposed before. This makes sense because, if we compute the mean of the prices for B-MAX cars and C-MAX cars, we notice that they are very similar to the mean of the prices for Fiesta cars, which is our reference for the car model as we said before.
Moreover, thanks to data normalization, we can visualize better the β coefficients with their own confidence intervals, because if the input is smaller the relative β will be greater and so its posterior mean and its confidence interval will be spaced from the X axis.

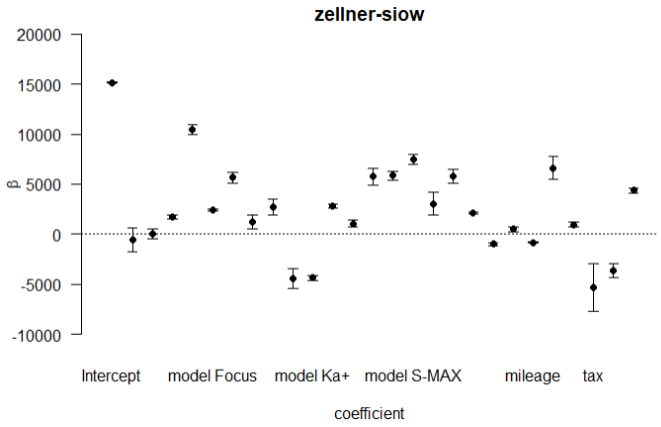








Now we can try to select a Zellner-Siow prior, which is completely non informative and there are no hyperparameters to be set.
As before, if we run the code we can observe a summary of the model, the plots of β distributions and the plot of the 95% confidence intervals for β. The plots are very similar to the G-prior case, the only difference is that in the Zellner-Siow case the values of β are a bit smaller. Even in this case we can suppose that **modelC-MAX** and **modelB-MAX** are not significant for our model for the same reason as before.
Despite G-prior and Zellner-Siow lead us to similar results, we must remember that there is a big difference between the two: Zellner-Siow prior is a mixture of G-priors and does not provide a closed form solution for the posterior, so bas.lm does some MCMC to compute a result. Therefore bas.lm using a Zellner-Siow prior provides an approximation.



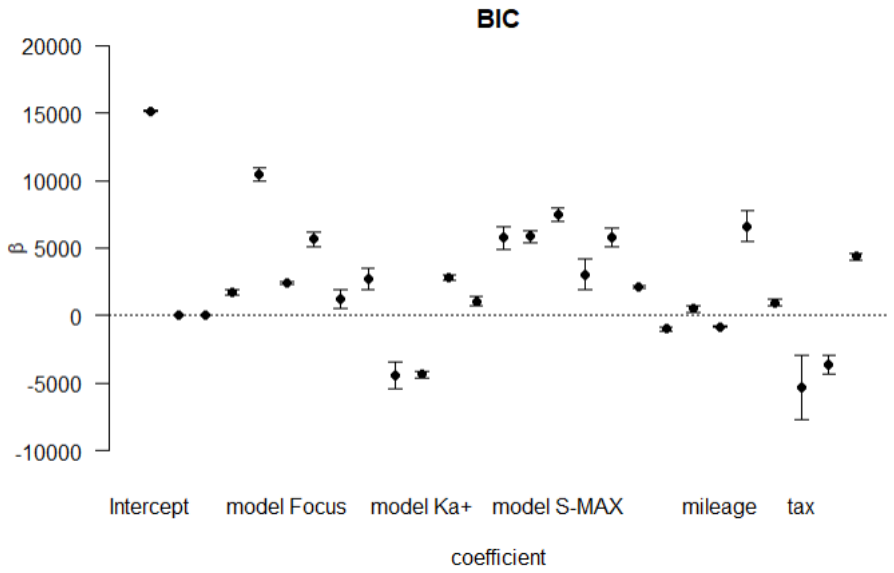
-

3 Model selection

We observed that the posterior mean values of the coefficients of **modelC-MAX** and **modelB-MAX** are close to the zero, and so we supposed that these two predictors may be not relevant for our model.
From the plot of the correlation between predictors we also noticed that some predictors are correlated between them, and so maybe any of them may be not very significant for the model. We can verify these assumptions through model selection.

Now we fit a BIC model and we visualize a summary of it. We can notice that the probability of $\beta! = 0$ is almost zero for **modelB-MAX** and **modelC-MAX**, as we supposed. Moreover, **modelB-MAX** and **modelC-MAX** are not present in the first model (which is the best one), and they appear only once in the five best models that are shown in the summary: thus our consideration about these predictors was correct. The other predictors instead seem to be all relevant because their probability of $\beta! = 0$ is very close to 1.
Then we select the best model, which is the one that does not contain **modelB-MAX** and **modelC-MAX**, and we plot the confidence intervals of β in the best BIC model.

	P(B != 0 | Y)	model 1	model 2	model 3	model 4	model 5
Intercept	1.000	1.000	1.000	1.000	1.000	1.000
model B-MAX	0.020	0.000	0.000	0.000	1.000	0.000
model C-MAX	0.013	0.000	0.000	0.000	0.000	1.000
model EcoSport	1.000	1.000	1.000	1.000	1.000	1.000
model Edge	1.000	1.000	1.000	1.000	1.000	1.000
model Focus	1.000	1.000	1.000	1.000	1.000	1.000
model Galaxy	1.000	1.000	1.000	1.000	1.000	1.000
model Grand C-MAX	0.888	1.000	0.000	1.000	1.000	1.000
model Grand Tourneo Connect	1.000	1.000	1.000	1.000	1.000	1.000
model KA	1.000	1.000	1.000	1.000	1.000	1.000
model Ka+	1.000	1.000	1.000	1.000	1.000	1.000
model Kuga	1.000	1.000	1.000	1.000	1.000	1.000
model Mondeo	1.000	1.000	1.000	1.000	1.000	1.000
model Mustang	1.000	1.000	1.000	1.000	1.000	1.000
model Puma	1.000	1.000	1.000	1.000	1.000	1.000
model S-MAX	1.000	1.000	1.000	1.000	1.000	1.000
model Tourneo Connect	1.000	1.000	1.000	1.000	1.000	1.000
model Tourneo Custom	1.000	1.000	1.000	1.000	1.000	1.000
year	1.000	1.000	1.000	1.000	1.000	1.000
transmissionManual	1.000	1.000	1.000	1.000	1.000	1.000
transmissionSemi-Auto	0.976	1.000	1.000	0.000	1.000	1.000
mileage	1.000	1.000	1.000	1.000	1.000	1.000
fuelTypeHybrid	1.000	1.000	1.000	1.000	1.000	1.000
fuelTypePetrol	1.000	1.000	1.000	1.000	1.000	1.000
tax	0.994	1.000	1.000	1.000	1.000	1.000
mpg	1.000	1.000	1.000	1.000	1.000	1.000
engineSize	1.000	1.000	1.000	1.000	1.000	1.000
BF	NA	1.000	0.126	0.024	0.021	0.013
PostProbs	NA	0.833	0.105	0.020	0.017	0.011
R2	NA	0.856	0.856	0.856	0.856	0.856
dim	NA	25.000	24.000	24.000	26.000	26.000
logmarg	NA	-70308.561	-70310.631	-70312.289	-70312.445	-70312.903



We also fitted a model with Lasso to see the differences with another model selection method and we implemented it with **JAGS**. We implemented the following Lasso model:

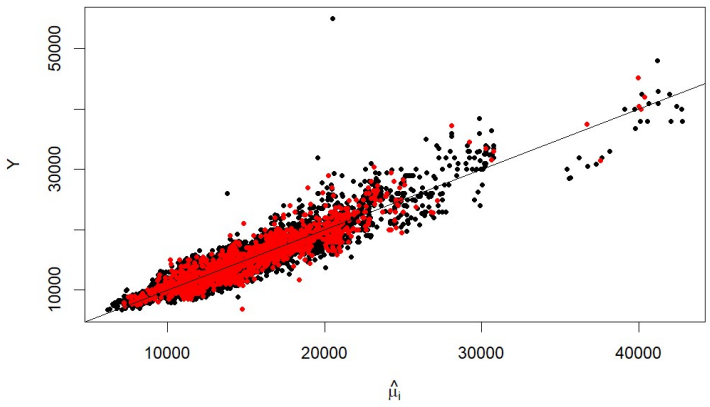
$$
\begin{aligned}
y_i &\sim \mathcal{N}(\alpha + X_i\beta, \sigma^2) \\
\alpha &\sim \mathcal{N}(0, 100) \\
\beta_i &\sim \mathcal{DE}(0, \sigma_b^2\sigma^2) \\
\sigma^2 &\sim \mathcal{IG}(0.01, 0.01) \\
\sigma_b^2 &\sim \mathcal{IG}(0.01, 0.01)
\end{aligned}
$$

We observe that the posterior mean values for β are a bit different from the ones computed with BIC and we also observe that Lasso does not shrink **modelB-MAX** (beta[6] in the image) and **modelC-MAX** (beta[7] in the image).

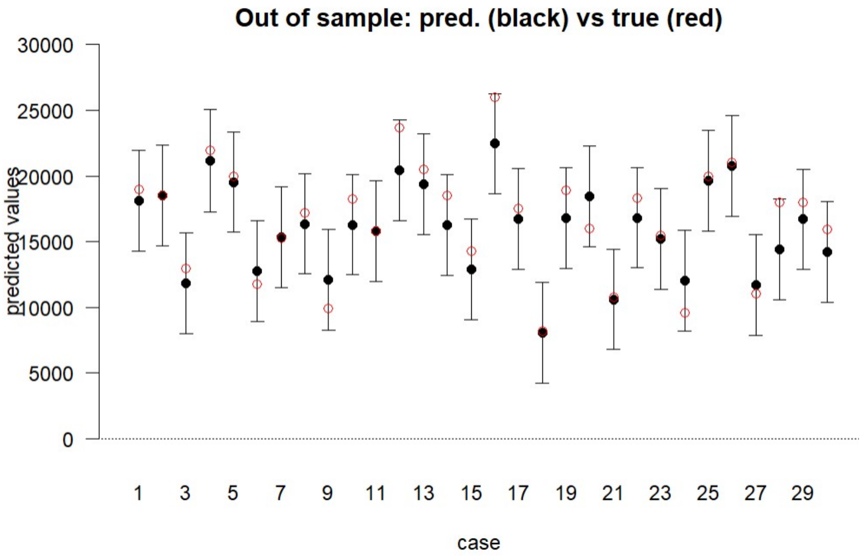
```
Iterations = 2010:3000
Thinning interval = 10
Number of chains = 1
Sample size per chain = 100

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean      SD Naive SE Time-series SE
alpha        0.615   9.881   0.9881          0.9881
beta[1]   2110.628  53.514   5.3514         11.1829
beta[2]   -841.702  28.772   2.8772          4.8002
beta[3]   6526.864 242.728  24.2728        164.0217
beta[4]  -3043.538 348.772  34.8772        155.8179
beta[5]   4619.550 105.271  10.5271         34.1375
beta[6]   -503.865 599.345  59.9345         59.9345
beta[7]    114.598 256.439  25.6439         25.6439
beta[8]   1761.305 103.558  10.3558         22.8667
beta[9]  10487.400 256.300  25.6300         47.5105
beta[10]  2390.776  54.097   5.4097          5.4097
beta[11]  5652.983 268.138  26.8138         12.5656
beta[12]  1346.415 301.697  30.1697         35.8866
beta[13]  2745.299 385.367  38.5367         38.5367
beta[14] -4561.154 466.273  46.6273         46.6273
beta[15] -4321.121 106.986  10.6986         10.6986
beta[16]  2827.661 115.188  11.5188         18.0079
beta[17]  1039.409 168.307  16.8307         20.6345
beta[18]  5218.374 392.451  39.2451         53.4969
beta[19]  5842.368 235.034  23.5034         18.5627
beta[20]  7432.186 237.691  23.7691         23.7691
beta[21]  3134.372 604.004  60.4004         60.4004
beta[22]  5918.407 347.325  34.7325         64.1560
beta[23] -1016.236 100.936  10.0936         21.4263
beta[24]   486.949 147.133  14.7133         17.7506
beta[25]  7301.511 586.493  58.6493        135.9803
beta[26]  1164.690 113.460  11.3460         27.3767
```

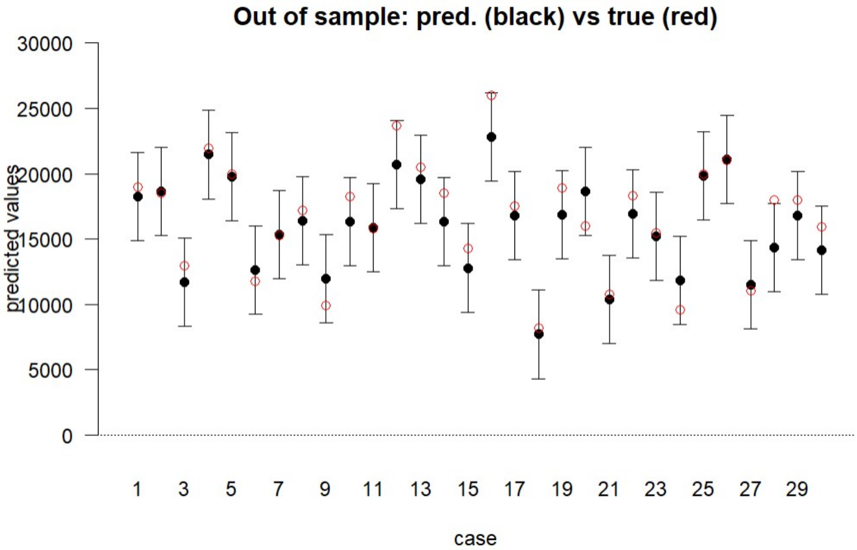
4 Prediction

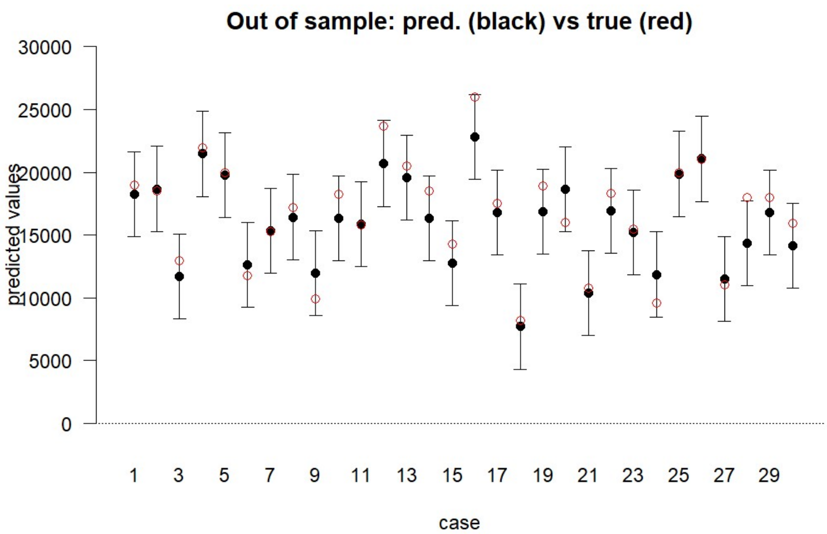
We will do the prediction phase first on the training set to verify how the model overfits the data, and then on the test set to verify the accuracy of our model. We will compare the predictions for the G-prior model, the Zellner-Siow model and the BIC model.
We plot all the data on a plane where the X axis is the predicted price and the Y axis is the real price: the more the point is near to the function Y = X, the more is good the prediction. We show on the report just the plot for the BIC model because it is very similar to the plots for G-prior and Zellner-Siow prior.



We also plot the real price and the prediction with its confidence interval, but in this case it would be impossible to visualize so much data, so we consider just few data of the test set for this plot.
The following images are respectively for the model fitted with G-prior, the model fitted with Zellner-Siow prior and the model fitted with BIC. We can see that the prediction is very good in every case because the predicted value is almost always within the confidence interval.







5 Conclusions

We can summarize what we done and provide some conclusions:

- First we normalized and cleaned our data and we commented them making some considerations to have a general idea of the problem. In particular, we tried to understand how our data are

distributed, the relevance of the predictors and the correlation between them, and we supposed that some predictors could be not significant for our model. We also distinguished between numerical and categorical variables and we split our data set into training and test set.

- We fitted a frequentist model and we analyzed the multicollinearity of the predictors. Then we fitted some bayesian models using different priors like G-prior and Zellner-Siow prior, and we observed that there are no significant differences between them, even if G-prior is a bit informative and has a closed form solution while Zellner-Siow prior is not informative and has no closed form solution.

- We noticed in the previous step that some predictors, in particular **modelB-MAX** and **modelC-MAX**, had a low posterior mean value of β. We verified this assumption and the assumption based on correlation between predictors through model selection.
  We started with BIC and we found out that the best model does not contain **modelB-MAX** and **modelC-MAX**, but the performance did not seem to be very much better than before. We also fitted a model with Lasso and it did not even shrink any predictors.

- We did the prediction and we analyzed the predicted values to understand which model could be the better. They had all very good performances, and there were not big differences between the first two models and the model found through model selection. Therefore we can conclude that in our problem all the predictors are more or less relevant to predict the price of a new car and we can simply use a bayesian model with a non informative prior.

Lastly, based on our consideration on multicollinearity we can try fitting a model eliminating **mpg** and compare it to the model with all the predictors. As expected the prediction with this model leads us to the same result, because the presence of a linear dependent covariate affects only the statistical meaning of the coefficients and not the prediction. The fact that the predictions are equivalent means that it is possible to remove **mpg** for a more clear and statistically significant model.