POLITECNICO
MILANO 1863

Bayesian learning

and

Monte Carlo Simulations

Project Work

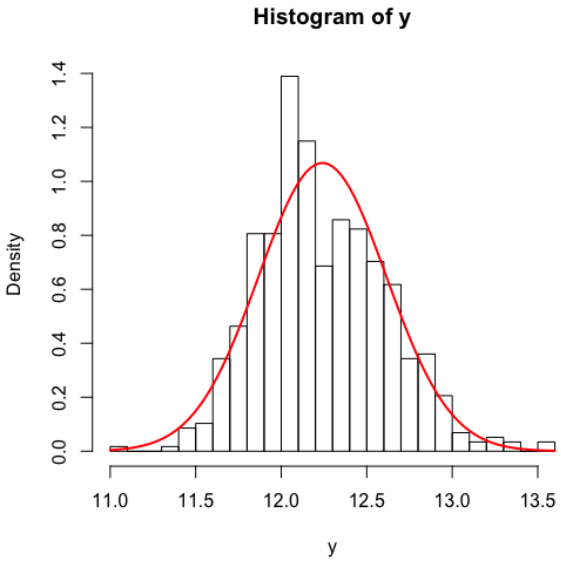
Ames House Price data

Family name	Given name	Student ID
Peretti	Carlo	945171
Pessina	Manuel	941667
Pisani	Matteo	944056
Secchi	Alessandro	944668

Academic Year: 2019/2020

1. Description of the problem and the data

In the following report it is analysed the dataset "Ames House Price" with the statistical methods seen during the course "Bayesian learning and Monte Carlo simulation". The R code for the first part is all reported in the appendix.
First, a brief analysis concerning the variables used to compute the final price of the house has been done.
The starting 76 covariates could be nominal, ordinal, continuous and discrete; they include all the information required by a typical home buyer. Further on in the report, the variables more in depth are analysed.
The first step is to load the data and eliminate all the cases in which there are some missing data because they are not suitable for the statistical analysis. Doing this operation, a dataset with 583 observations instead of 1460 has been obtained.
The main response variable is the value of the house and in the analysis, the logarithm has been considered.
The following graph shows the histogram of logarithm of the sale prior compared with a normal distribution (red curve) with, as mean, the mean of y and, as standard deviation, the σ^2 of y.
It can be seen that they follow approximately the same distribution. So, the sale price is distributed as a Normal.



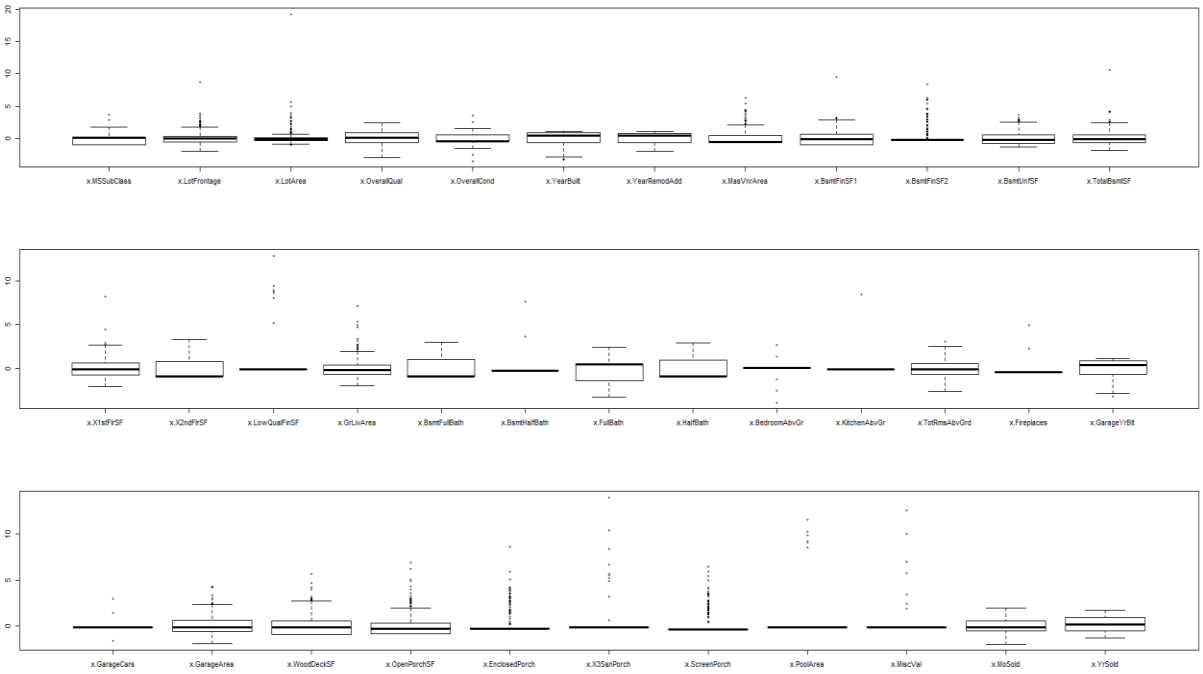
This is clearly a linear model composed by a response y (the sale price of the house) and by several explanatory variables. The structure of the linear model follows the following formula:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \varepsilon_i$$

The goal of computing a regression model is to make statistical inference, in order to understand the dependence of the response over some variables.

2. Covariates analysis

Now the covariates are analysed. Taken into account only the numeric variables, they have been scaled. Before starting the statistical analysis, some considerations about the 36 numeric covariates have been done, in order to visualize their boxplot to see how the values of the variables are distributed.



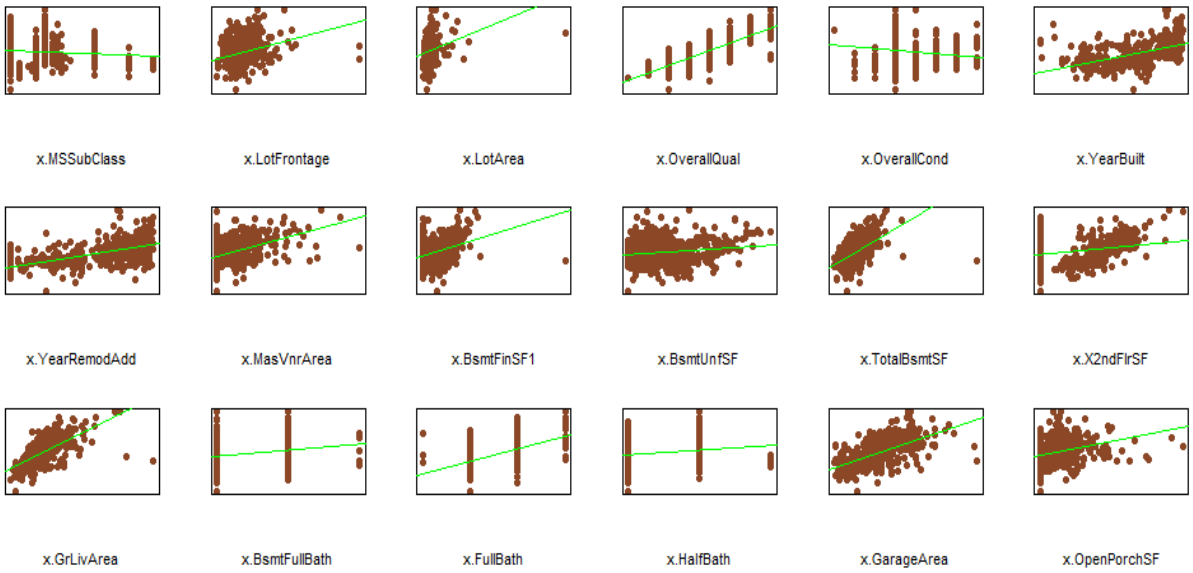
Using this graphical representation, it can be seen easily that the boxplot of some covariates is very compact with respect to others. This means that their values are always very similar, and they do not have a high influence in the regression, so they can be eliminated in order to clean the model.

The remaining numeric covariates are 21 but a further analysis can be considered by computing the correlation between the covariates. Looking at the heatmap [APPENDIX] of the correlation matrix it's clear that some variables are highly correlated and this is not suitable for a linear model because if the correlation coefficient is near to one it means that they are bringing the same information to the final response and moreover can cause problems in the inversion of the coefficients matrix. So, one of the two can be eliminated.

In correlations -1 means there is a perfect negative linear relationship, 0 means there is no linear relationship while +1 means there is a perfect positive linear relationship.

Specifically, it can be noticed that the most critical correlation coefficients are between "GrLivArea" and "TotRmsAbvGrd" with a value of 0.81, between "TotalBsmtSF" and "X1stFlrSF" with a value of 0.9 and between "YearBuilt" and "GarageYrBlt" with a value of 0.86. One of the two is kept, while the second has been removed from the model.

Once obtained the final dataset with the numeric covariates a plot can show (graphically) the relationship between them and the response.



From this graphic representation a couple of considerations can be done. The first one is that the beta coefficient is quite always positive (the response increases if the covariate increases). Furthermore, it looks clear the discrete nature of some variables, e.g. the "Overall Conditions" and the "Overall Quality". If the scatterplot doesn't indicate there's at least somewhat of a linear relationship, the correlation doesn't mean much. In fact, correlation only applies to linear relationships and if a strong relationship exists but it's not linear, the correlation factor may be misleading, because in some cases a strong curved relationship exists. That's why it's critical to check also the condition of linearity.

Now the categorical variables are considered; in this case the analysis is a little more complicated for a linear model, because they don't have numeric values, but they have names or words. So, it has been used the command "FastDummies" in order to divide one factorial variable in several numeric variables that take only 0 and 1 as value. For example, for the variable "Street" there are two possible values: "Pave" and "Gravel"; with this function the software will create to covariates "Street_Paved" and "Street_Gravel" (If the street is paved the first variable will take value 1 and the second 0). Looking at the summary, it is worth to notice that some generated dummies take the null value for every observation; it means that they are useless for the dataset. So, they have been eliminated.

Now the remaining numeric variables have been added with the dummies in order to obtain a clean final dataset with 190 covariates.

3. Bayesian model choice with BIC, AIC, Zellner G priors

At this point there is a clean data frame with many covariates (190) in which a model selection should be performed.

To perform this operation, the BAS library has been used that with the given data frame creates 2^p different models, where p is the number of covariates and because of computer space

allocation and time is set as default $p<25$. This is also the reason why, rerunning the code, the variables selected change every time, usually by no more than 2 or 3.

By using the algorithm from the BAS library, a criterion must be assumed to select the covariates that can affect more the response variable. 3 different types of priors have been considered: BIC, AIC and Zellner G-Prior that are explained better in the following subchapters.

3.1 BIC Prior

The Bayesian Information Criterion (BIC) is defined as following:

$$BIC = -2\ln(\widehat{likelihood}) + (p+1)\ln(n)$$

where n is the number of observations and p is the number of covariates.
The BIC considers, as shown, the likelihood that in general, with a model M, can be written as:

$$likelihood = p(y_i|\alpha, \beta, \sigma^2) = L(\alpha, \beta, \sigma^2)$$

that is the probability for the observed data y occurring under the given parameters α, ß, σ^2. A model that fits better the observations will have a higher value of maximized likelihood. But the other part of the sum introduces a penalization based upon the number of covariates, so too many predictors may result in overfitting the data. When
p increases, the second term increases as well. This provides a trade-off between the goodness of fit given by the first term and the model complexity represented by the second term.
A key index provided by the summary is the log marginal likelihood: in the model selection with BIC criterion the best model has the highest logmargin, that is associated to the lowest BIC value.
Running this part of the code we obtained a model with 38 covariates and logmargin -682,11. [The code is written in the appendix]

3.2 AIC Prior

The Akaike's Information Criterion (AIC) is very similar to the BIC and works in the same way, but the penalty rate function is lower, in fact is defined as $2 \cdot p$ where p is still the number of covariates.

Exactly for this reason the final model obtained by BAS with this criterion has almost double of covariates considered inside.

Running BAS with AIC prior it has been obtained a model with 91 covariates.

3.3 Zellner G-Prior

Zellner's G prior is useful whenever we do not have enough information about the prior in order to reduce the influence of the hyper-parameters. In this case for the prior variance we use a non-informative prior

$$\sigma^2|X \sim \pi(\sigma^2) = \sigma^{-2}$$

This is an improper prior: it is not a proper probability but can be used if it allows to obtain a finite distribution for the posterior.
On the other hand, the posterior of β depends also on the observation

$$\beta|\sigma^2, X \sim N_{k+1}(\tilde{\beta}, c\sigma^2(X^tX)^{-1})$$

This is not a pure Bayesian prior, but it is very useful because it allows not to use the complex M matrix. The main advantage is, therefore, that the Zellner's G prior allows the experimenter to introduce information about the location parameter of the regression while bypassing the most difficult aspects of the prior specification, namely the derivation of the prior correlation structure: so, the experimental prior determination is restricted to the choices of $\tilde{\beta}$ and of the constant c. c can be interpreted as a measure of the amount of information available in the prior relative to the sample. For instance, setting 1/c = 0.5 gives the prior the same weight as 50% of the sample.

At last running this BAS code a model with 41 covariates has been obtained.

4. Comparison between models obtained through DIC criterion

After being obtained 3 different models based on different number of covariates, there is the choice of which is the best one that fits better the response.
To do this it has been used the Deviance Information Criterion (DIC) through the library JagsUI and samples generated through Monte Carlo Markov Chains [APPENDIX].
DIC is defined as:

$$DIC = p_D + \overline{D(\vartheta)}$$

or equivalently:

$$DIC = 2p_D + D(\bar{\theta})$$

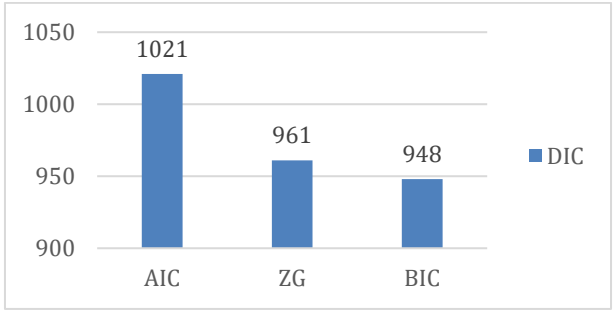
where D(θ) represents the deviance

$$D(\theta) = -2\log(p(y|\theta)$$

and ρ_D the effective number of parameters of the model, which is computed as:

$$\rho_D = \overline{D(\theta)} - D(\bar{\theta})$$

where $\bar{\theta}$ is the expectation of θ and $\overline{D(\theta)}$ is the expectation of the deviance of θ.
Both can be easily computed by sampling from a MCMC.
The larger the effective number of parameters is, the easier it is for the model to fit the data, and so the deviance needs to be penalized. Of course, a better model shows a lower DIC index.
The DIC values are then reported in the table below.



In the appendix is also reported the diagnostic check of the trace plots and density.
Based upon this value the decision is to use the BIC model, which is one of the most popular, that still has around ~40 covariates considered.
Since the high number of variables, the BIC procedure is reiterated twice.
The final model obtained with this procedure have 32 covariates and a logmarg -672,87.
Next, because of the high number of covariates from this run, x and with no possibility of a further reduction as probably the best model with BIC criterion has already been obtained, it has been decided to fit our model with no more than 20 covariates.
To do this the covariates are ordered by considering the highest posterior probability of inclusion of the covariates in the model and kept the best 20. [APPENDIX]
At last a BIC model (*ames.finale*) is recreated to understand through *logmargin* (-690 compared to -673 of X.BIC2) and the DIC value if this model drops down in performance.
The summary of the JagsUI for the DIC process is reported in the appendix and tells that through *overlap0* which are the most important variables that may not contain the 0 (at 95% credible region), through $\widehat{R}$ if it is <1.1 the chain converges, and through the number of *n.eff* how many samples it has to be taken from the chain to be independent observations.
The DIC value is comparable (915 respect to 948) to the model with almost double covariates considered. The final model is a trade-off between the general goodness of the fit of the covariates and its simplicity.

5. Linear Regression

Now we have obtained the final dataset with our 20 covariates and we can compute a linear model. The likelihood of the ordinary normal linear model is

$$f(y|X,\beta,\sigma 2) = (2\pi\sigma^2)^{-\frac{n}{2}} * \exp\left[-\frac{1}{2\sigma^2}(y - X\beta)^t (y - X\beta)\right]$$

In frequentist linear regression, the best explanation is taken to mean the coefficients, β, that minimize the residual sum of squares (RSS). This is known as the maximum likelihood estimate (MLE) of β because it is the value that is the most probable, given the inputs, X, and outputs, y. We can find the coefficients in R and look at their estimates and other basic info.

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               9.047197   0.153663  58.877  < 2e-16 ***
x.linx.LotConfig_CulDSac  0.090814   0.023313   3.895  0.00011 ***
x.liny.x.LotFrontage      0.032812   0.006291   5.216 2.57e-07 ***
x.linx.SaleCondition_Normal 0.112206 0.021618   5.190 2.93e-07 ***
x.linx.SaleType_New       0.180392   0.026741   6.746 3.79e-11 ***
x.linx.Neighborhood_NridgHt 0.081053 0.019203   4.221 2.84e-05 ***
x.liny.x.TotalBsmtSF      0.088497   0.007814  11.326  < 2e-16 ***
x.linx.RoofMatl_Membran   3.170917   0.195767  16.197  < 2e-16 ***
x.linx.RoofMatl_WdShake   2.960018   0.175698  16.847  < 2e-16 ***
x.linx.RoofMatl_Tar.Grv   2.973587   0.157684  18.858  < 2e-16 ***
x.liny.x.BsmtUnfSF       -0.042596   0.005583  -7.630 1.01e-13 ***
x.linx.RoofMatl_CompShg   3.057982   0.151429  20.194  < 2e-16 ***
x.linx.RoofMatl_WdShngl   3.190073   0.156860  20.337  < 2e-16 ***
x.liny.x.OverallCond      0.053399   0.006190   8.626  < 2e-16 ***
x.liny.x.YearBuilt        0.091862   0.007838  11.719  < 2e-16 ***
x.linx.Neighborhood_Crawfor 0.174045 0.023441   7.425 4.21e-13 ***
x.liny.x.OverallQual      0.105455   0.008808  11.972  < 2e-16 ***
x.liny.x.GarageArea       0.044929   0.007566   5.938 5.05e-09 ***
x.linx.Neighborhood_StoneBr 0.135490 0.031343   4.323 1.82e-05 ***
x.linx.Condition2_PosN   -1.002576   0.090754 -11.047  < 2e-16 ***
x.liny.x.GrLivArea        0.140386   0.007279  19.286  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1228 on 562 degrees of freedom
Multiple R-squared:  0.8955,    Adjusted R-squared:  0.8918
F-statistic: 240.9 on 20 and 562 DF,  p-value: < 2.2e-16
```

In the first column we can see the estimate of the beta coefficients, and as expected, looking at the previous analysis in chapter 2, they are positive for almost all the variables.

We can also assess that our selection was good because all the t-values have an absolute value bigger than 2: it means that they bring a significant effect to the response.
The p-value is small for all the covariates: the data support the null hypothesis β=0; only at the significant level 0; we can see it looking at the 3 stars on the right. Since the threshold is very low, the null hypothesis is non-supported by the data. We can reject the null coefficient, so they are all good covariates in order to explain the model.
The last consideration is about the R-squared indexes, very high, around 0.9. It indicates how well our covariates are able to explain the response.
The second way for the estimation of the is done through the "bas.lm" function, that provides the bayesian posterior distribution of the coefficients. The results are:

	post mean	post sd	2.5%	97.5%
Intercept	12.24098818	0.005090365	12.23085343	12.25088517
x.LotConfig_CulDSac	0.08990523	0.024853954	0.04400523	0.13962076
y.x.LotFrontage	0.03273850	0.006364123	0.02016093	0.04505046
x.SaleCondition_Normal	0.11212644	0.021889859	0.06846939	0.15414353
x.SaleType_New	0.18029608	0.027248571	0.12986003	0.23591910
x.Neighborhood_NridgHt	0.08064506	0.019815908	0.04129295	0.11832054
y.x.TotalBsmtSF	0.08849814	0.008041770	0.07278164	0.10369828
x.RoofMatl_Membran	3.17009562	0.212265835	2.77819160	3.55177972
x.RoofMatl_WdShake	2.95859275	0.190588916	2.60755290	3.30100283
x.RoofMatl_Tar.Grv	2.97224422	0.173855440	2.66495053	3.28734760
y.x.BsmtUnfSF	-0.04259710	0.005630884	-0.05376505	-0.03176517
x.RoofMatl_CompShg	3.05660044	0.168752077	2.75393400	3.35186608
x.RoofMatl_WdShngl	3.18834888	0.175060720	2.87539777	3.49449912
y.x.OverallCond	0.05338862	0.006269151	0.04106658	0.06544140
y.x.YearBuilt	0.09186944	0.007990475	0.07627025	0.10711855
x.Neighborhood_Crawfor	0.17394852	0.023620861	0.12739683	0.21953691
y.x.OverallQual	0.10553067	0.009015177	0.08763405	0.12254982
y.x.GarageArea	0.04495286	0.007643497	0.03015391	0.05998328
x.Neighborhood_StoneBr	0.13517112	0.031888862	0.07633433	0.20084529
x.Condition2_PosN	-1.00229168	0.093343026	-1.18414939	-0.82644482
y.x.GrLivArea	0.14038434	0.007420417	0.12579441	0.15451260

Comparing these values with the previous ones they look almost identical, except for the intercept (bas.lm computes the intercept as the mean of the response). None of the quantiles of the contains the 0, so all the regressors are likely to explain effectively the response. Graphically this aspect is shown in the following figures, the first associated to the 4 covariates with the highest postProb, the second to the 4 covariates with the smallest postProb. In both pictures the point corresponding to the null coefficient is in the tails of the parameter distribution as can be appreciated look in the APPENDIX.

Jags allow the process of linear regression with the Bayesian method, in an alternative way (respect to bas.lm). For the computation, the model is defined with a hierarchical prior for β. Since the prior isn't conjugate, Jags runs a MCMC where the posteriors of the are the target stationary distribution. The model is the following:

$$y_i \sim N(\alpha + X\beta, \sigma^2)$$

$$\alpha \sim N(0,100)$$

$$\beta_j \sim N(0, {\sigma_b}^2)$$

$${\sigma}^{-2} \sim G(0.01, 0.01)$$

$${\sigma_b}^{-2} \sim G(0.01, 0.01)$$

	Mean	SD		2.5%	25%	50%	75%	97.5%
alpha	9.20006	0.158381	alpha	8.89035	9.09268	9.20002	9.30680	9.51145
beta[1]	0.09240	0.023368	beta[1]	0.04670	0.07661	0.09237	0.10801	0.13855
beta[2]	0.03164	0.006316	beta[2]	0.01928	0.02738	0.03162	0.03589	0.04411
beta[3]	0.11088	0.021862	beta[3]	0.06792	0.09616	0.11079	0.12570	0.15389
beta[4]	0.17765	0.027056	beta[4]	0.12471	0.15950	0.17756	0.19557	0.23105
beta[5]	0.08307	0.019296	beta[5]	0.04514	0.07014	0.08317	0.09604	0.12078
beta[6]	0.08503	0.007878	beta[6]	0.06958	0.07970	0.08505	0.09033	0.10039
beta[7]	2.99886	0.199304	beta[7]	2.60788	2.86393	2.99994	3.13391	3.38676
beta[8]	2.79736	0.180099	beta[8]	2.44283	2.67575	2.79776	2.91885	3.14707
beta[9]	2.82164	0.162345	beta[9]	2.50277	2.71204	2.82264	2.93145	3.13909
beta[10]	-0.04182	0.005593	beta[10]	-0.05283	-0.04556	-0.04183	-0.03805	-0.03093
beta[11]	2.90595	0.156080	beta[11]	2.59943	2.80020	2.90578	3.01136	3.21245
beta[12]	3.03876	0.160652	beta[12]	2.72279	2.93105	3.03888	3.14791	3.35147
beta[13]	0.05331	0.006216	beta[13]	0.04109	0.04907	0.05336	0.05751	0.06543
beta[14]	0.09198	0.007820	beta[14]	0.07664	0.08670	0.09197	0.09725	0.10733
beta[15]	0.17395	0.023429	beta[15]	0.12821	0.15797	0.17405	0.18964	0.21981
beta[16]	0.10667	0.008791	beta[16]	0.08941	0.10076	0.10666	0.11256	0.12394
beta[17]	0.04543	0.007589	beta[17]	0.03048	0.04032	0.04541	0.05051	0.06041
beta[18]	0.13730	0.031474	beta[18]	0.07566	0.11602	0.13724	0.15846	0.19920
beta[19]	-0.98735	0.090598	beta[19]	-1.16395	-1.04840	-0.98809	-0.92633	-0.80891
beta[20]	0.13926	0.007301	beta[20]	0.12493	0.13435	0.13925	0.14420	0.15346

6. Sensitivity Analysis

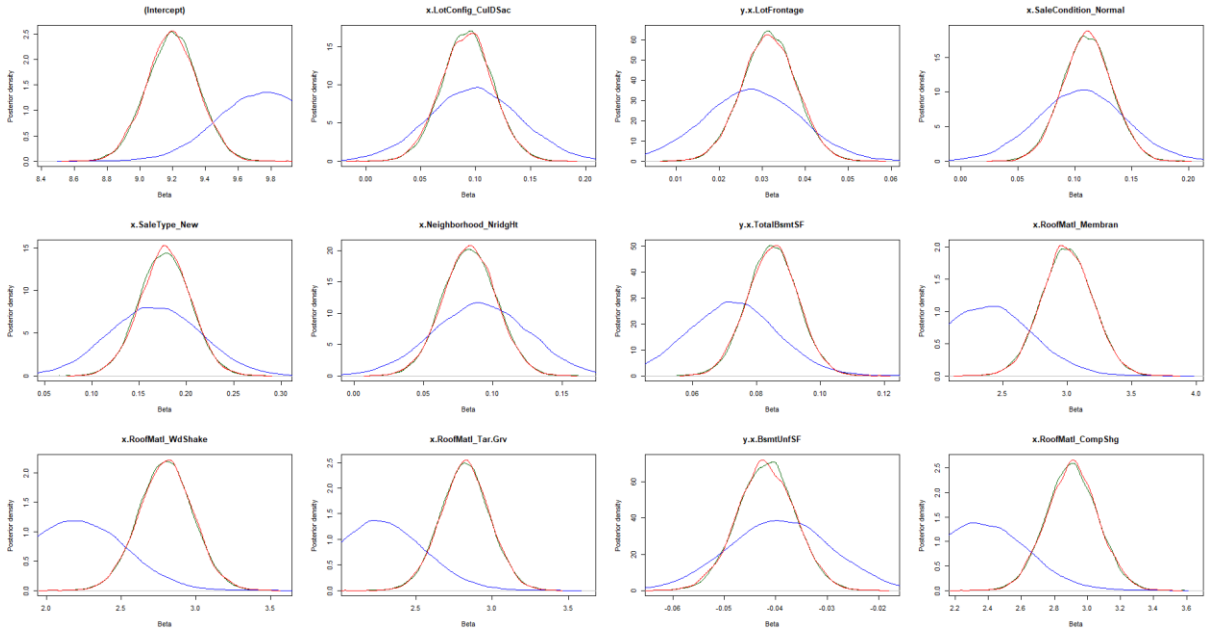
Implementing the Normal-Normal Inverse Gamma model in Jags, it is interesting to evaluate how the output is influenced by the choice of the hyperparameters. In the first sensitivity test the variance of the prior will be strongly reduced.

In the second test, the changes will affect the tuning values of the MCMC like the burn-in, the number of iterations of the chain and the number of chains, running in parallel.

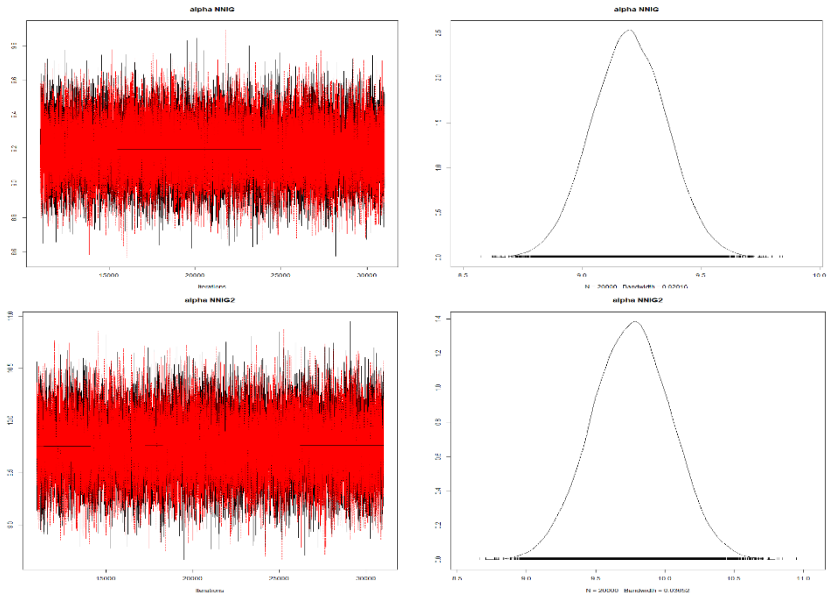
Model	First	Second	Third
$\boldsymbol{y_i} \sim$	$N(\alpha + X\beta, \sigma^2)$		
$\boldsymbol{\alpha} \sim$	$N(0{,}100)$		
$\boldsymbol{\beta_i} \sim$	$N(0, {\sigma_b}^2)$		
$\boldsymbol{\sigma^{-2}} \sim$	$G(0.01{,}0.01)$	$G(10{,}10)$	$G(0.01{,}0.01)$
$\boldsymbol{{\sigma_b}^{-2}} \sim$	$G(0.01{,}0.01)$	$G(10{,}10)$	$G(0.01{,}0.01)$
burn in	10000	10000	100
n.iteration	20000	20000	10000
n.chains	2	2	1

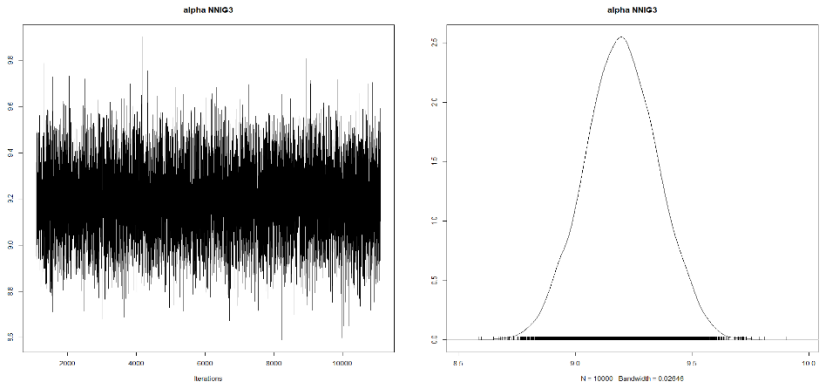
The values of the β, contained in the summary, of the three simulation are quite similar, but looking at their distribution, appear clear differences. The reduction of the burn-in and of the number of iterations doesn't affect the posterior distribution: the Markov Chain reaches anyway the stationarity. It's possible to appreciate large differences in the second case: acting on the variance, the chain takes more time to explore all the data space during the iterations.

The figure below shows graphically the previous assessments for alpha and the first β (the second set of the coefficients is in the APPENDIX).



Analysing the trace plots of the variable "alpha" shown in the APPENDIX, in all the simulations, included the last one with decreased burn-in, they show a good MCMC (mean reverting, not periodic, explore a large portion of space).





Another typical prior, without closed form posteriors, for the Bayesian linear regression, is the **Lasso prior**. The hierarchical structure is:

$$y_i \sim N(\alpha + X\beta, \sigma^2)$$

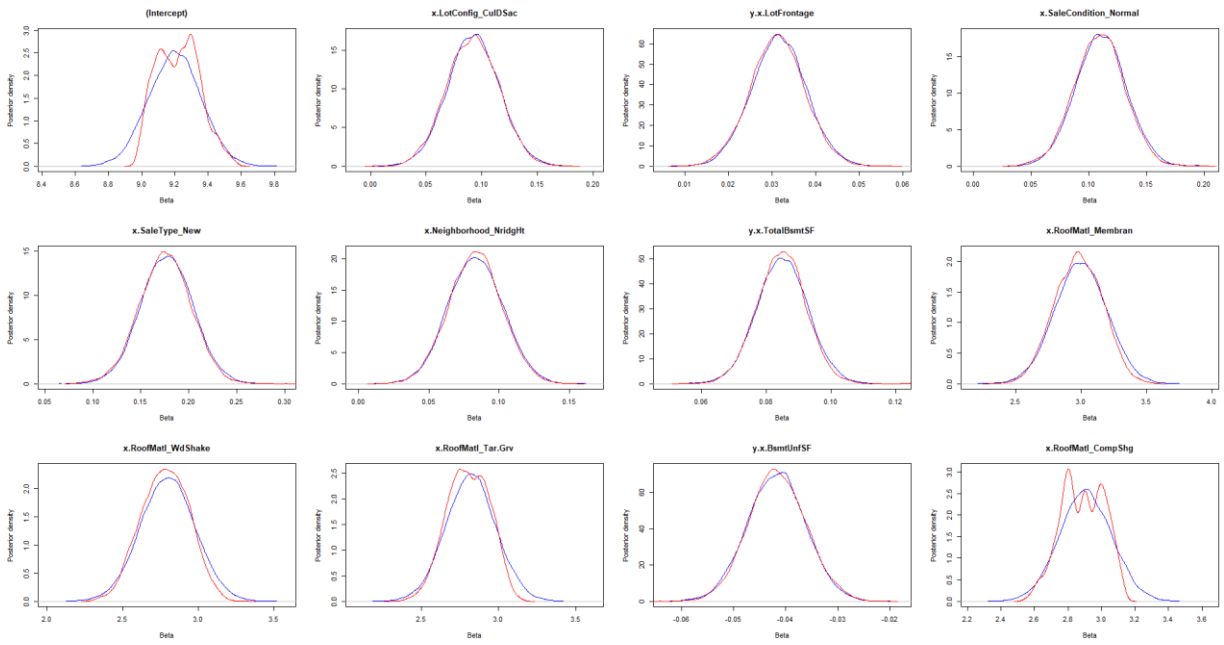
$$\alpha \sim N(0{,}100)$$

$$\boldsymbol{\beta_i \sim DE(0, \sigma_b^2)}$$

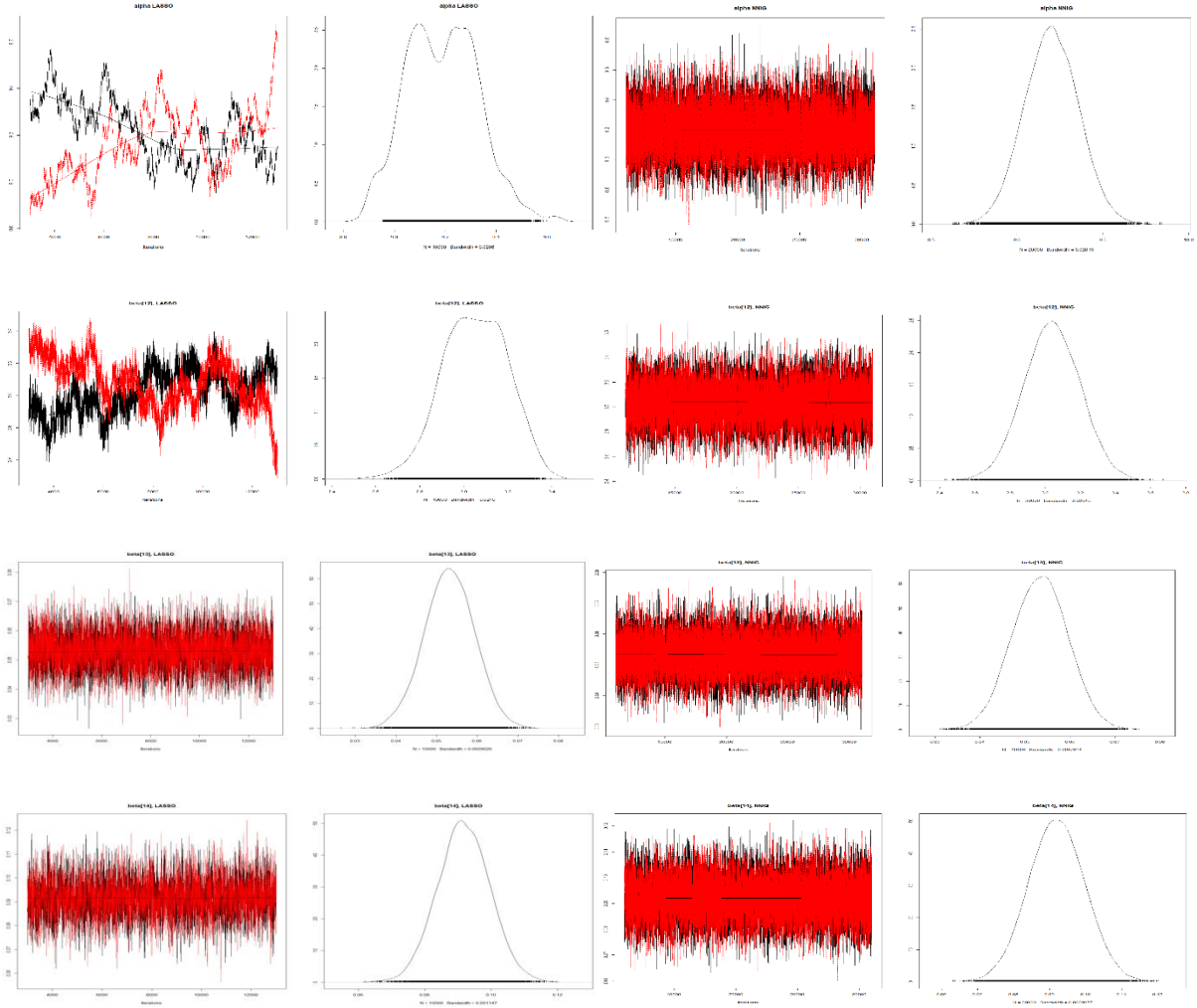
$$\sigma^{-2} \sim G(0.01{,}0.01)$$

$$\sigma_b{}^{-2} \sim G(0.01{,}0.01)$$

Due to the double exponential for β is more picked in the 0. The posterior distributions are quite similar; the only relevant differences are for those coefficients with posterior mean close to 0, where the model with the Lasso prior is effectively more picked around it.



The trace plots where the coefficient is closer to 0 have worse performances (tend to explore a minor portion of the space).



8. Prediction

The results obtained performing an analysis of the predictions is useful to assess the goodness of the model. In an out-sample prediction the dataset is divided in two parts: the first observations are used to fit the model, while the last ones (y_{real}, the actual prices of the last x houses) are assumed not to be measured and need to be predicted, by means of their covariates. The prior predictive distribution has the same distribution of the likelihood, so it's a Normal:

$$y_{new}|\sigma^2, \beta, X_{new} \sim N(X_{new}\beta, \sigma^2)$$

The posterior distributions of β and σ^2, obtained as

$$\prod_{i=1}^{n-x}(y_i|X_i, \beta, \sigma^2)\pi(\beta, \sigma^2) \propto \pi(\sigma^2, \beta|y, X)$$

are exploited to obtain y_{pred}. Hence, the posterior predicted distribution of the last $\boldsymbol{x}$ responses can be computed analytically by

$$\int \pi(y_{new}|\sigma^2, \beta, X_{new})\pi(\sigma^2, \beta|y, X)d\sigma^2, d\beta$$

The comparison between y_{real} and y_{pred} shows how much the model can give a good estimate of the houses prices, once the features are known. "As shown below" the model is simulated with two different numbers of responses to be predicted, according to the results to be emphasized. The simulations exploit JAGS (look at the code) with the typical Normal-Normal-IG prior for the parameters.

- In the first case $\mathrm{x} = \mathrm{length}(y_{pred}) = 15$. The results obtained are reported in the next tab

	Mean	SD	2.5%	25%	50%	75%	97.5%
Yp[1]	12.60	0.1248	12.36	12.52	12.60	12.69	12.85
Yp[2]	12.02	0.1247	11.78	11.94	12.02	12.11	12.27
Yp[3]	12.08	0.1262	11.84	12.00	12.08	12.17	12.33
Yp[4]	12.69	0.1248	12.45	12.61	12.69	12.78	12.94
Yp[5]	12.94	0.1266	12.69	12.85	12.94	13.02	13.19
Yp[6]	12.59	0.1257	12.35	12.51	12.59	12.68	12.84
Yp[7]	11.82	0.1236	11.58	11.74	11.82	11.91	12.06
Yp[8]	12.33	0.1238	12.09	12.25	12.33	12.41	12.57
Yp[9]	11.82	0.1238	11.58	11.74	11.82	11.90	12.06
Yp[10]	11.94	0.1244	11.70	11.86	11.95	12.03	12.19
Yp[11]	12.08	0.1257	11.83	12.00	12.08	12.17	12.33
Yp[12]	13.05	0.1284	12.80	12.96	13.05	13.13	13.30
Yp[13]	12.08	0.1248	11.84	12.00	12.08	12.16	12.32
Yp[14]	11.78	0.1239	11.54	11.69	11.78	11.86	12.02
Yp[15]	12.24	0.1243	12.00	12.16	12.24	12.32	12.48

[1,]	11.84940
[2,]	11.68688
[3,]	12.16598
[4,]	12.13619
[5,]	11.98293
[6,]	12.06681
[7,]	12.88567
[8,]	12.19096
[9,]	12.16003
[10,]	12.64433
[11,]	12.38839
[12,]	12.56755
[13,]	12.07254
[14,]	12.25486
[15,]	12.49313

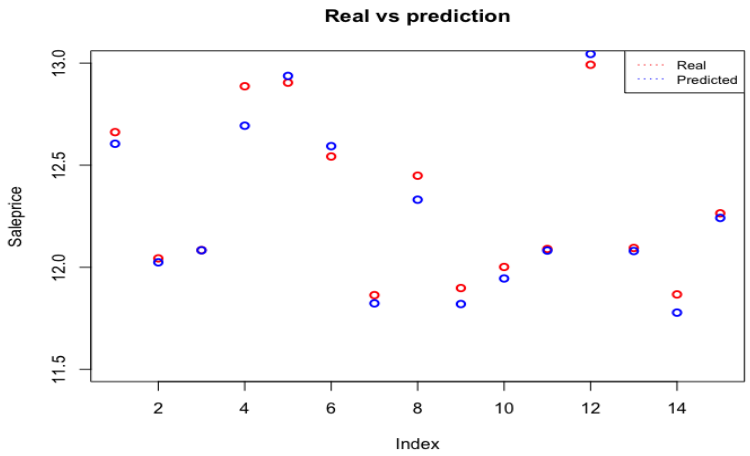
Fig: in blue the posterior mean of the prediction, in red the real value of the sale price

The predictor, y_{est} under squared loss is the posterior predictive mean, which is the product between matrix of explanatory variables X_{new} by the Bayes estimate of β. It's possible to evaluate the mean square error between y_{est} and y_{real} as

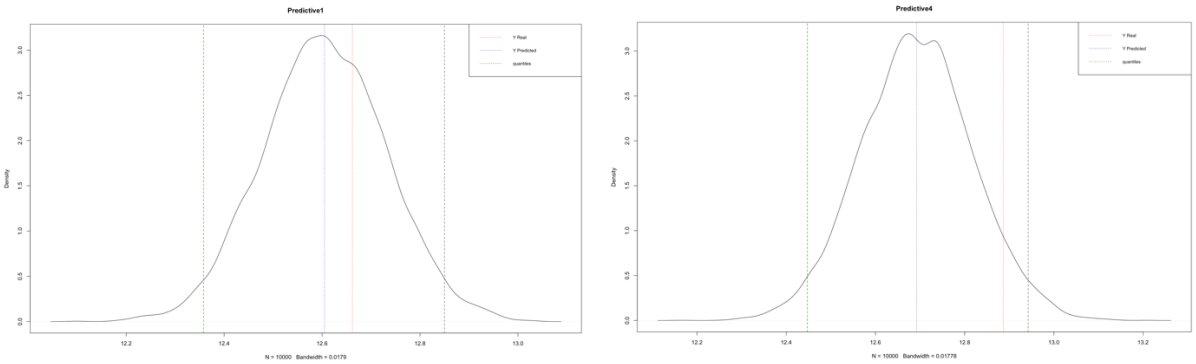
$$MSE = \frac{1}{x-2}\sum_{i}^{x}\left(y_{est,i} - y_{real,i}\right)^2$$

=0.0062286

Graphically:



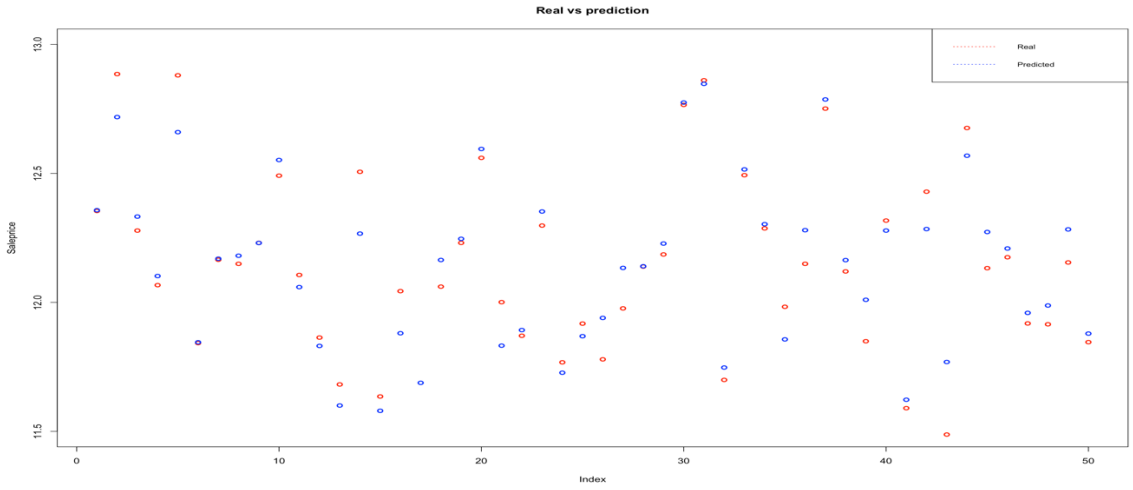
The predictions are very close the real values of the response, so the model is valid to provide an estimate of the price of a new house. The validity can be further checked, looking how y_{real} "almost" always fits inside the confidence intervals of y_{pred}.
This is demonstrated also with the following graphs where is shown the distribution of the probability of few ypred.



In the first plot the distribution of Yp[1] with its quantiles is shown. The red line indicates the real value of the Sale Price for this specific observation and we can see that it is clearly into the confidence interval. In the second graph is plotted the probability of the prediction with the biggest difference from the real value. Nevertheless, also in this case the red line is included in the confidence interval.

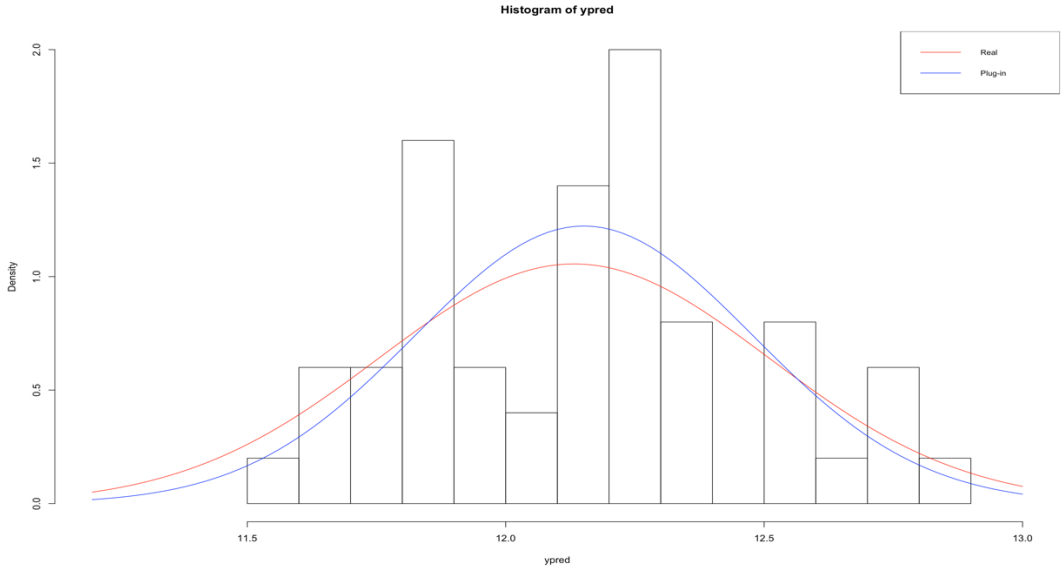
- In the second case $\mathrm{x} = \mathrm{length}(y_{pred}) = 50$. In such this way the fitting of the model is done through less observation (theoretically it will be a bit worse, actually doesn't change much).

$$MSE$$
0.0192917

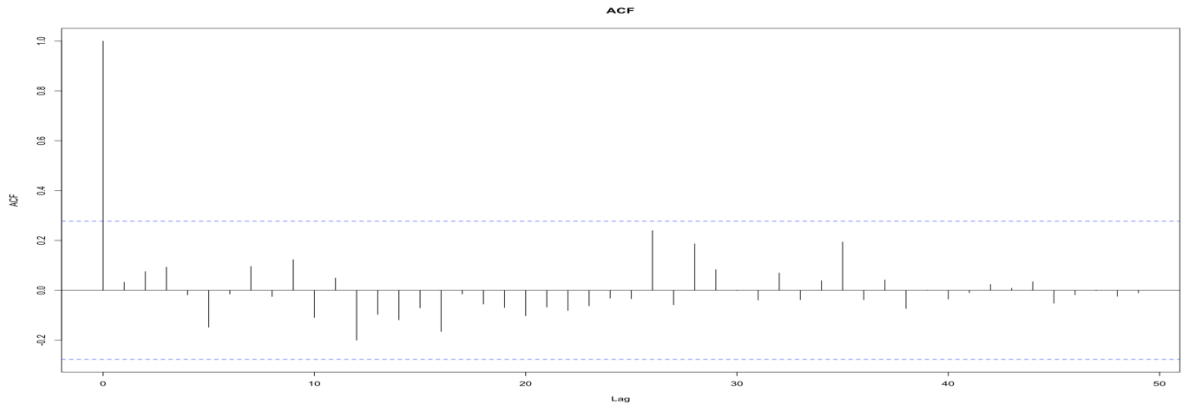


The estimate of the prediction of the response is like the real last 50 house prices. The MSE is a bit higher, as expected, but anyway this linear model confirms to describe well the process in order to evaluate a house price in Ames. This can be assessed, also, through the comparison

between the predictors, with the associated plug-in Normal distribution, and the Normal of the real sale prices.



Add comment



APPENDIX

Chapter 1

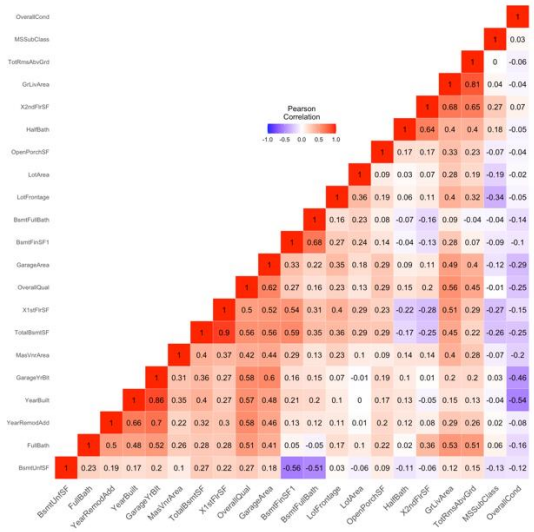
```
rm(list=ls())
library(fastDummies)
library(BAS)
library(rjags)
library(jagsUI)

#### LETTURA DATI####
ames <- read.csv("ameshouse.csv", sep=";")
house <- ames[which(complete.cases(ames)),] #pulizia missing values

#### PRIMO STUDIO COVARIATES####
#Formattazione dati per analisi
ames.numeric <- dplyr::select_if(house, is.numeric) #selezione variabili solo numeriche
x <- as.matrix(ames.numeric[,1:36])
x <- scale(x)
y <- ames.numeric$SalePrice
y <- log(y)

#istogramma e normale con media di y e sd di y
hist(y, prob=T, breaks=20)
curve(dnorm(x,mean=mean(y),sd=sd(y),log=FALSE),add=T, col='red', lwd=2)
```

Chapter 2

```
#Prima lettura dei dati per possibile eliminazione di alcune covariates poco influenti
summary(x)
dataframe <- data.frame( x=as.matrix(x))
par(mfrow=c(3,1))
boxplot(dataframe[1:12])
boxplot(dataframe[13:25])
boxplot(dataframe[26:36])

#Adesso scriviamo il nuovo modello inserendo solo le variabili numeriche piu' rilevanti
x <- x[,-c(15, 10 ,18, 21, 22, 24, 26, 28, 30:36)]
summary(x)

#### ANALISI CORRELAZIONE TRA LE 21 VARIABILI PER ULTERIORE PULIZIA ####
cormat <- round(cor(x),2)
head(cormat)

library(reshape2)
melted_cormat <- melt(cormat)

# Get lower triangle of the correlation matrix
get_lower_tri<-function(cormat){
  cormat[upper.tri(cormat)] <- NA
  return(cormat)
}
# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}
reorder_cormat <- function(cormat){
  # Use correlation between variables as distance
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
}
# Reorder the correlation matrix
cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
library(ggplot2)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
```

```
                      midpoint = 0, limit = c(-1,1), space = "Lab",
                      name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()
# Print the heatmap
ggheatmap +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    legend.justification = c(1, 0),
    legend.position = c(0.6, 0.7),
    legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                               title.position = "top", title.hjust = 0.5))
```

Correlation matrix between the 21 numeric covariates




```
#### DATA FRAME SCREMATO ####
x <- x[,-c(12, 18, 19)] #rimozione di ulteriori 3 covariates altamente correlate tra loro
numeric <- data.frame( x=as.matrix(x))

n <- nrow(numeric)
par(mfrow=c(3,6), mar=c(4.2,2,2,1.2))
for (j in 1:18){ plot(numeric[,j], y, xlab=names(numeric)[j],
                      pch=19, col="sienna4", xaxt="n", yaxt="n")
abline(lm(y~numeric[,j]), col="green")}


#Analisi ulteriori covariates categoriche
ames.factor <- dplyr::select_if(house, is.factor) #selezione delle sole variabili categoriche
summary(ames.factor) #chiarire  quali come sono distribuite le dummies
dummies <- fastDummies::dummy_cols(ames.factor,remove_first_dummy = TRUE, remove_selected_columns = TRUE)

dummies.pulito <- dummies[,-c(2, 12, 49, 51, 55, 56, 57, 73, 76, 77, 81, 82, 84, 98, 118, 122, 124, 129, 146,
                              148, 151, 155, 156, 165, 180, 194, 196 )]

#Unione dei data frame numerico+dummies
completa <- data.frame(y=numeric , x=dummies.pulito)
#DATAFRAME PULITO DA MISSING VALUES E COVARIATES POCO INFLUENTI
```

Chapter 3

```
#### EFFETTUARE CONFRONTO DIC CON I TRE METODI BIC, AIC E ZG PER VEDERE QUAL'E' IL PIU' EFFICACE####

ames.comparazione <- completa
#creare un nuovo modello con le covariates risultate dalla prima scrematura
#e poi fare i tre metodi per calcolare il DIC


# --BIC--
ames.BIC = bas.lm(y ~ ., data = ames.comparazione,
                      prior = "BIC", method = "MCMC", MCMC.iterations = 10000 ,
                      modelprior = uniform())
summary(ames.BIC)
best.BIC = which.max(ames.BIC$logmarg)
bestmodelBIC = ames.BIC$which[[best.BIC]]
bestmodelBIC

X.BIC <- ames.comparazione[,c(bestmodelBIC)]

ames.BIC$logmarg[best.BIC] #Troviamo il logmarg del modello BIC


# --AIC--
ames.AIC = bas.lm(y ~ ., data = ames.comparazione,
                       prior = "AIC", method = "MCMC", MCMC.iterations = 10000 ,
                       modelprior = uniform())
summary(ames.AIC)
best.AIC = which.max(ames.AIC$logmarg)
bestmodelAIC = ames.AIC$which[[best.AIC]]
bestmodelAIC

X.AIC <- ames.comparazione[,c(bestmodelAIC)] #Troviamo il logmarg del modello AIC

# --ZG-prior--
ames.ZG = bas.lm(y ~ ., data = ames.comparazione,
                      prior = "g-prior", method = "MCMC", MCMC.iterations = 10000 ,
                      modelprior = uniform())
summary(ames.ZG)
best.ZG = which.max(ames.ZG$logmarg)
bestmodelZG = ames.ZG$which[[best.ZG]]
bestmodelZG

X.ZG <- ames.comparazione[,c(bestmodelZG)] #Troviamo il logmarg del modello ZS
```

Chapter 4

```
#### USO DI JAGS PER DEFINIRE IL VALORE DIC TRAMITE METODI BIC, AIC, ZS####

model.string <- "model{
    # prior
    c2 <- n
    # prior means
    for (j in 1:P){ mu.beta[j] <- 0.0 }

    # calculation of xtx
    for (i in 1:P){ for (j in 1:P){
       inverse.V[i,j] <- inprod( x[,i] , x[,j] )
    }}
    for(i in 1:P){ for (j in 1:P){
      prior.T[i,j] <- inverse.V[i,j] * tau /c2
    }}

    # likelihood

        for (i in 1:n){
            y[i] ~ dnorm( mu[i], tau ) # stochastic componenent
            mu[i] <- inprod( beta[], x[i,] )
        }

    # prior distributions
    beta[1:P] ~ dmnorm( mu.beta[], prior.T[,] )
    tau    ~ dgamma( 0.01, 0.01 )
    s2    <- 1/tau
     s <-sqrt(s2)
}"
```

```
#DIC - AIC
library(jagsUI)

model <- textConnection(model.string)

x <- model.matrix(~., X.AIC)
n <- nrow(x)
p <- ncol(x)

jags.m <- jagsUI::jags(model,
                       inits=NULL,
                       data = list('n' = n,
                                   'P' = p,
                                   'x' = x,
                                   'y' = y),
                       n.chains = 2,
                       n.iter = 5000,
                       n.burnin = 2000,
                       n.adapt = 1000,
                       parameters.to.save="beta")
jags.m
summary(jags.m)
#VALORE DIC~1050
```

For the others two prior (BIC – Zellner's G Prior) the code is the same. The only difference is that a different input for the x (X.BIC – X.ZG) is set.

The results indicate BIC prior as the best method. So, twice iterations have been done in order to find the best dataset.

```
library(BAS)

#Prima iterazione

ames.BIC1 = bas.lm(y ~ ., data = X.BIC,
                     prior = "BIC", method = "MCMC", MCMC.iterations = 100000 ,
                     modelprior = uniform())
summary(ames.BIC1)
best.BIC1 = which.max(ames.BIC1$logmarg)
bestmodelBIC1 = ames.BIC1$which[[best.BIC1]]
bestmodelBIC1

X.BIC1 <- X.BIC[,c(bestmodelBIC1)]

#Seconda iterazione

ames.BIC2 = bas.lm(y ~ ., data = X.BIC1,
                   prior = "BIC", method = "MCMC", MCMC.iterations = 100000 ,
                   modelprior = uniform())
summary(ames.BIC2)
best.BIC2 = which.max(ames.BIC2$logmarg)
bestmodelBIC2 = ames.BIC2$which[[best.BIC2]]
bestmodelBIC2

X.BIC2 <- X.BIC1[,c(bestmodelBIC2)]

ames.BIC2$logmarg[best.BIC2]
```

```
#riordine variabili in funzione della probabilita' marginale di inclusione
ordinato = order(ames.BIC2[["probne0"]])
#scelta delle 20 covariates piu' influenti nella costruzione del modello
k <- ncol(X.BIC2)
ames.primeventiBIC = ordinato[(k-20):k]

ames.finale <- X.BIC2[,c(ames.primeventiBIC-1)]


#BIC FINALE PER STIMA LOGMARG CON LE 20 COVARIATES MIGLIORI
ames.BICfin = bas.lm(y ~ ., data = ames.finale,
                     prior = "BIC", method = "MCMC", MCMC.iterations = 100000 ,
                     modelprior = uniform())
summary(ames.BICfin)
best.BICfin = which.max(ames.BICfin$logmarg)
ames.BICfin$logmarg[best.BICfin]
#MODELLO FINALE TROVATO, adesso possiamo procedere con il calcolo dei vari coefficenti
```

Summary: "ames.BICfin"

In this summery the most important factor is the very high values of the posterior probability of inclusion. It indicates that these 20 covariates are included in quite all the models.

	P(B != 0 | Y)	model 1	model 2	model 3	model 4	model 5
Intercept	1.00000	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.LotConfig_CulDSac	0.99038	1.0000	0.00000000	1.000000e+00	1.000000e+00	0.000000e+00
y.x.LotFrontage	0.99946	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.SaleCondition_Normal	0.99906	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.SaleType_New	0.99928	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.Neighborhood_NridgHt	0.99643	1.0000	1.00000000	0.000000e+00	1.000000e+00	0.000000e+00
y.x.TotalBsmtSF	0.99974	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.RoofMatl_Membran	0.99943	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.RoofMatl_WdShake	0.99944	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.RoofMatl_Tar.Grv	0.99947	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
y.x.BsmtUnfSF	0.99981	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.RoofMatl_CompShg	0.99972	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.RoofMatl_WdShngl	0.99948	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
y.x.OverallCond	0.99968	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
y.x.YearBuilt	0.99975	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.Neighborhood_Crawfor	0.99975	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
y.x.OverallQual	0.99998	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
y.x.GarageArea	0.99986	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
x.Neighborhood_StoneBr	0.99819	1.0000	1.00000000	1.000000e+00	0.000000e+00	1.000000e+00
x.Condition2_PosN	0.99955	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
y.x.GrLivArea	0.99995	1.0000	1.00000000	1.000000e+00	1.000000e+00	1.000000e+00
BF	NA	1.0000	0.01023069	2.703554e-03	1.745644e-03	3.133537e-04
PostProbs	NA	0.9860	0.00850000	2.400000e-03	1.500000e-03	6.000000e-04
R2	NA	0.8955	0.89270000	8.922000e-01	8.921000e-01	8.902000e-01
dim	NA	21.0000	20.00000000	2.000000e+01	2.000000e+01	1.900000e+01
logmarg	NA	-689.9783	-694.56070171	-6.958915e+02	-6.963290e+02	-6.980465e+02

Here a quickly description of the 20 final covariates is reported.

1. **LotConfig_CulDSac:** Lot configuration - Cul-de-sac
2. **LotFrontage:** Linear feet of street connected to property
3. **SaleCondition_Normal:** Condition of sale - Normal Sale
4. **SaleType_New:** Type of sale - Home just constructed and sold
5. **Neighborhood_NridgHt:** Physical locations within Ames city limits - Northridge Heights
6. **TotalBsmtSF:** Total square feet of basement area
7. **RoofMatl_Membran:** Roof material - Membrane
8. **RoofMatl_WdShake:** Roof material - Wood Shakes
9. **RoofMatl_Tar.Grv:** Roof material - Gravel & Tar
10. **BsmtUnfSF:** Unfinished square feet of basement area
11. **RoofMatl_CompShg:** Roof material - Standard (Composite) Shingle
12. **RoofMatl_WdShngl:** Roof material - Wood Shingles
13. **OverallCond:** Rates the overall condition of the house
14. **YearBuilt:** Original construction date
15. **Neighborhood_Crawfor:** Physical locations within Ames city limits - Crawford
16. **OverallQual:** Rates the overall material and finish of the house

17. **GarageArea:** Size of garage in square feet
18. **Neighborhood_StoneBr:** Physical locations within Ames city limits - Stone Brook
19. **Condition2_PosN:** Proximity to various conditions - Near positive off-site feature--park, greenbelt, etc.
20. **GrLivArea:** Above grade (ground) living area square feet

In order to find the DIC of the final dataset "ames.finale" a JAGS model have been created to see the value (The code of the JAGS model is the same than the one showed one page before).

Summary of the JAGS model of "ames.finale"

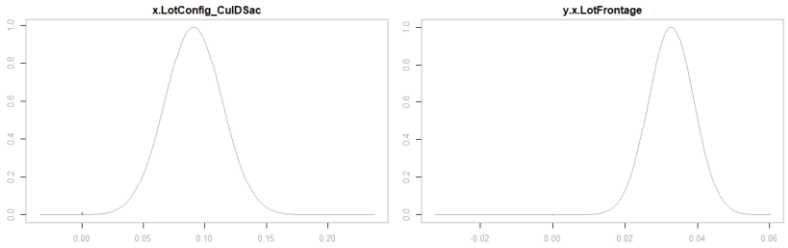
	mean	sd	2.5%	50%	97.5%	overlap0	f	Rhat	n.eff
beta[1]	9.028	0.648	7.766	9.026	10.289	FALSE	1.000	1.000	6000
beta[2]	0.090	0.099	-0.105	0.091	0.282	TRUE	0.817	1.000	6000
beta[3]	0.033	0.027	-0.020	0.033	0.086	TRUE	0.890	1.000	2682
beta[4]	0.111	0.092	-0.070	0.111	0.290	TRUE	0.888	1.001	2156
beta[5]	0.180	0.114	-0.043	0.179	0.404	TRUE	0.938	1.000	6000
beta[6]	0.081	0.081	-0.079	0.081	0.237	TRUE	0.838	1.000	6000
beta[7]	0.089	0.033	0.025	0.089	0.156	FALSE	0.996	1.000	6000
beta[8]	3.169	0.818	1.570	3.152	4.787	FALSE	1.000	1.000	6000
beta[9]	2.964	0.743	1.473	2.985	4.415	FALSE	1.000	1.000	6000
beta[10]	2.973	0.667	1.652	2.971	4.294	FALSE	1.000	1.000	6000
beta[11]	-0.043	0.024	-0.088	-0.043	0.004	TRUE	0.965	1.001	6000
beta[12]	3.057	0.639	1.814	3.059	4.309	FALSE	1.000	1.000	6000
beta[13]	3.189	0.665	1.872	3.187	4.492	FALSE	1.000	1.000	6000
beta[14]	0.053	0.026	0.003	0.053	0.105	FALSE	0.980	1.000	2633
beta[15]	0.092	0.034	0.027	0.091	0.158	FALSE	0.998	1.001	2064
beta[16]	0.172	0.099	-0.021	0.171	0.369	TRUE	0.959	1.000	6000
beta[17]	0.106	0.037	0.031	0.106	0.178	FALSE	0.998	1.000	6000
beta[18]	0.044	0.032	-0.020	0.044	0.106	TRUE	0.914	1.000	6000
beta[19]	0.134	0.132	-0.129	0.136	0.392	TRUE	0.845	1.001	1599
beta[20]	-0.993	0.388	-1.760	-0.993	-0.238	FALSE	0.994	1.001	2360
beta[21]	0.140	0.030	0.080	0.140	0.198	FALSE	1.000	1.000	6000
deviance	364.789	33.183	300.496	364.227	431.145	FALSE	1.000	1.001	3909

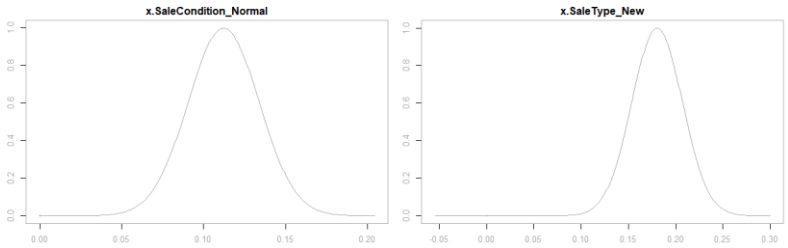
```
#Diagnostic check
par(mfrow=c(1,1))
plot(jags.m) # traceplots e densita'

jags.mcmc <- as.mcmc(jags.m$samples[1])
geweke.diag(jags.mcmc)
geweke.plot(jags.mcmc)

heidel.diag(jags.mcmc)
```

Chapter 5

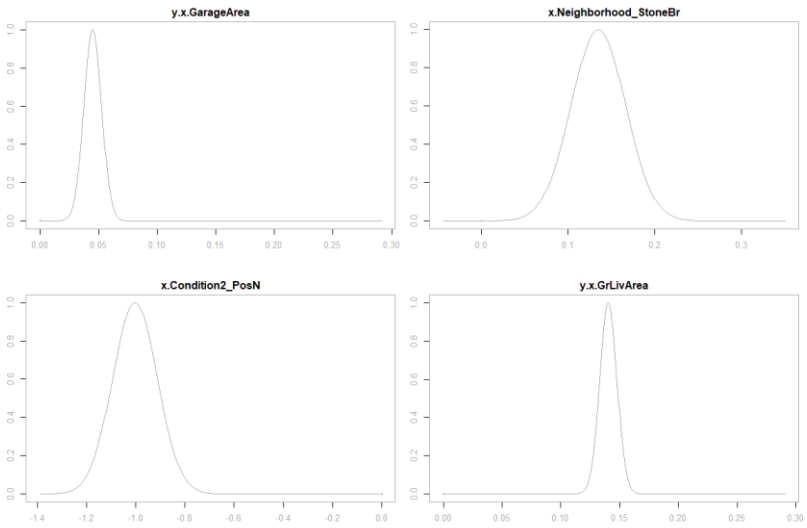
```
#### MODELLO LINEARE E COEFFICIENTI ####
#Modello lineare
x.lin <- as.matrix(ames.finale)
linear <- lm(y~x.lin)
summary(linear)
X <- x.lin

confint(linear)

#COEFFICIENTI
ames.coef = coef(ames.BICfin)
# Retreat bounds of credible intervals
out = confint(ames.coef)[, 1:2]

# Combine results and construct summary table
coef.BIC = cbind(ames.coef$postmean, ames.coef$postsd, out)
names = c("post mean", "post sd", colnames(out))
colnames(coef.BIC) = names
coef.BIC

#Plottaggio dei beta

par(mfrow = c(3, 2), col.lab = "darkgrey", col.axis = "darkgrey", col = "darkgrey")
plot(ames.coef, subset = 2:5, ask = F)
par(mfrow = c(3, 2), col.lab = "darkgrey", col.axis = "darkgrey", col = "darkgrey")
plot(ames.coef, subset = 18:21, ask = F)
confint(ames.coef, parm = 2:21)
```

Plot of the distribution of the “worst” 4 beta coefficients





Plot of the distribution of the “best” 4 beta coefficients




```
library(rjags)
#Modello con Normal-Normal InverseGamma
n <- nrow(x)
p <- ncol(x)
model_string.AMES.NNIG <- "model{

  # Likelihood
  for(i in 1:n){
    Y[i]   ~ dnorm(mu[i],inv.var)
    mu[i] <- alpha + inprod(X[i,],beta[])
  }

  # Prior for beta
  for(j in 1:p){
    beta[j] ~ dnorm(0,inv.var.b)
  }

  # Prior for the inverse variance
  inv.var   ~ dgamma(0.01, 0.01)
  inv.var.b ~ dgamma(0.01, 0.01)
  alpha     ~ dnorm(0, 0.01)

}"


model_AMES.NNIG <- jags.model(textConnection(model_string.AMES.NNIG),
                        data = list(Y=y,n=n,p=p,X=x), n.chains = 2)

update(model_AMES.NNIG, 10000)

AMES.NNIG <- coda.samples(model_AMES.NNIG,
                      variable.names=c("beta","alpha"),
                      n.iter=20000)

summary(AMES.NNIG)
```

Chapter 6

For the sensitivity test the JAGS model used is the same reported in the chapter 5. Only some specific parameters have been changed.

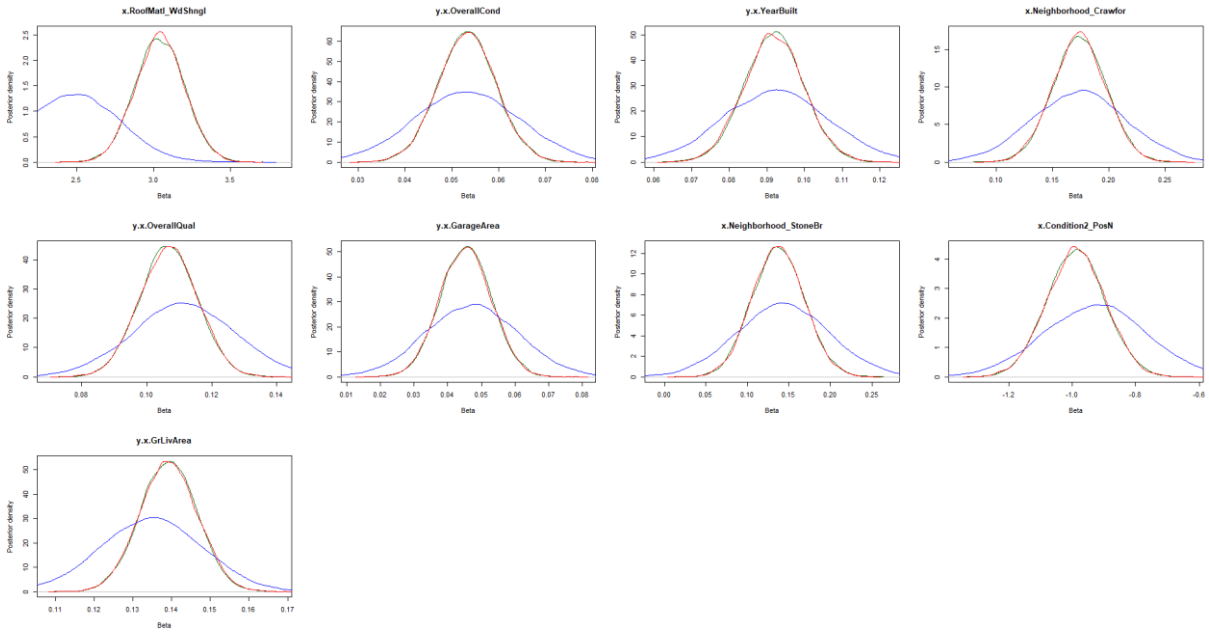
Summary of the second NNIG model.

	Mean	SD		2.5%	25%	50%	75%	97.5%
alpha	9.75960	0.28682	alpha	9.198744	9.56461	9.76128	9.95407	10.32114
beta[1]	0.09861	0.04234	beta[1]	0.015015	0.07014	0.09877	0.12710	0.18116
beta[2]	0.02762	0.01139	beta[2]	0.005201	0.01997	0.02766	0.03529	0.04999
beta[3]	0.10583	0.03916	beta[3]	0.029241	0.07934	0.10598	0.13250	0.18247
beta[4]	0.16678	0.04864	beta[4]	0.071990	0.13374	0.16678	0.19958	0.26261
beta[5]	0.09097	0.03491	beta[5]	0.022595	0.06752	0.09078	0.11452	0.15987
beta[6]	0.07255	0.01427	beta[6]	0.044619	0.06287	0.07266	0.08219	0.10021
beta[7]	2.36668	0.35956	beta[7]	1.667750	2.12438	2.36497	2.60948	3.07379
beta[8]	2.20149	0.32524	beta[8]	1.565016	1.97933	2.20092	2.42135	2.83731
beta[9]	2.26293	0.29327	beta[9]	1.687636	2.06450	2.26133	2.46165	2.83716
beta[10]	-0.03917	0.01015	beta[10]	-0.059015	-0.04603	-0.03918	-0.03231	-0.01937
beta[11]	2.34965	0.28309	beta[11]	1.799326	2.15760	2.34831	2.54172	2.90530
beta[12]	2.48545	0.29334	beta[12]	1.915508	2.28611	2.48471	2.68549	3.05838
beta[13]	0.05314	0.01130	beta[13]	0.030969	0.04552	0.05311	0.06074	0.07534
beta[14]	0.09240	0.01427	beta[14]	0.064634	0.08281	0.09237	0.10199	0.12058
beta[15]	0.17326	0.04246	beta[15]	0.089651	0.14470	0.17339	0.20204	0.25646
beta[16]	0.11138	0.01597	beta[16]	0.080381	0.10061	0.11135	0.12216	0.14293
beta[17]	0.04720	0.01375	beta[17]	0.020178	0.03793	0.04727	0.05646	0.07404
beta[18]	0.14393	0.05672	beta[18]	0.032902	0.10567	0.14348	0.18192	0.25523
beta[19]	-0.92753	0.16269	beta[19]	-1.245655	-1.03721	-0.92703	-0.81894	-0.60774
beta[20]	0.13501	0.01320	beta[20]	0.109113	0.12614	0.13501	0.14393	0.16068

Summary of the third NNIG model.

	Mean	SD		2.5%	25%	50%	75%	97.5%
alpha	1.224e+01	0.10795	alpha	12.022552	12.168364	1.224e+01	12.30835	12.452918
beta[1]	8.353e-02	0.06156	beta[1]	-0.034496	0.042023	8.362e-02	0.12391	0.207668
beta[2]	3.392e-02	0.02407	beta[2]	-0.014227	0.018065	3.398e-02	0.05002	0.080557
beta[3]	6.121e-02	0.05540	beta[3]	-0.045762	0.024105	6.010e-02	0.09815	0.171853
beta[4]	1.382e-01	0.06821	beta[4]	0.008231	0.091394	1.372e-01	0.18298	0.275420
beta[5]	3.839e-02	0.04613	beta[5]	-0.051891	0.007873	3.822e-02	0.06882	0.129993
beta[6]	8.065e-02	0.02185	beta[6]	0.037282	0.066108	8.093e-02	0.09490	0.123960
beta[7]	-4.992e-04	0.11618	beta[7]	-0.229837	-0.073153	-2.029e-04	0.07294	0.232409
beta[8]	2.324e-04	0.11500	beta[8]	-0.229183	-0.070863	-4.005e-04	0.07263	0.234459
beta[9]	2.275e-03	0.11368	beta[9]	-0.222170	-0.068568	2.092e-03	0.07225	0.232047
beta[10]	-4.411e-02	0.01867	beta[10]	-0.080141	-0.056731	-4.413e-02	-0.03193	-0.007771
beta[11]	-7.969e-02	0.09464	beta[11]	-0.275240	-0.138566	-7.636e-02	-0.01717	0.099997
beta[12]	8.017e-02	0.09611	beta[12]	-0.104107	0.016322	7.895e-02	0.14161	0.274088
beta[13]	2.592e-02	0.02476	beta[13]	-0.023479	0.009558	2.585e-02	0.04248	0.074342
beta[14]	1.317e-01	0.02347	beta[14]	0.086303	0.115846	1.317e-01	0.14757	0.178134
beta[15]	1.263e-01	0.07733	beta[15]	-0.023314	0.074315	1.253e-01	0.17689	0.281382
beta[16]	1.097e-01	0.02786	beta[16]	0.054542	0.091152	1.095e-01	0.12818	0.164837
beta[17]	2.259e-02	0.02613	beta[17]	-0.028366	0.004962	2.278e-02	0.03982	0.073287
beta[18]	7.019e-03	0.08809	beta[18]	-0.169125	-0.050185	8.196e-03	0.06489	0.179625
beta[19]	-3.797e-05	0.11517	beta[19]	-0.230317	-0.072082	5.631e-05	0.07181	0.228319
beta[20]	1.173e-01	0.02301	beta[20]	0.072293	0.101926	1.174e-01	0.13252	0.161647

Distribution of the last 9 beta coefficients



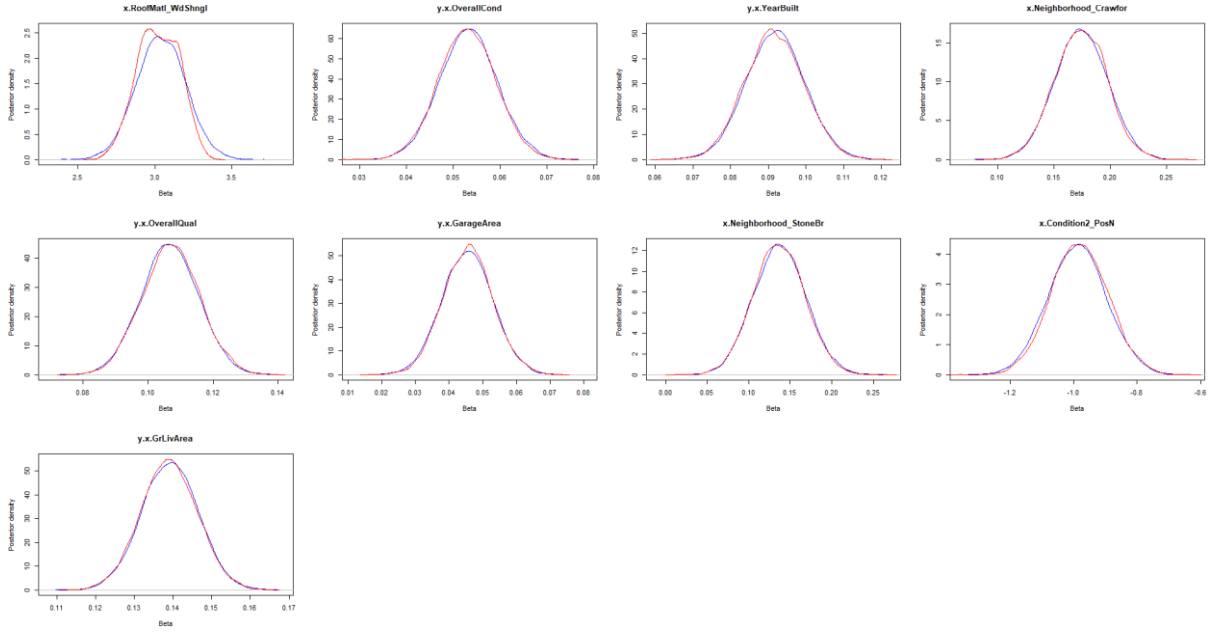
```r
#CONFRONTO SENSITIVITIES
#Prime 12 beta
s1 <- AMES.NNIG[[1]]
s2 <- AMES.NNIG2[[1]]
s3 <- AMES.NNIG3[[1]]
par(mfrow=c(3,4))
for(index in 1:12){
  d1 <- density(s1[,index])
  d2 <- density(s2[,index])
  d3 <- density(s3[,index])
  mx <- max(d1$y,d2$y,d3$y)
  plot(d1,ylim=c(0,mx),xlab="Beta",ylab="Posterior density",main=colnames(x)[index],
       col="darkgreen")
  lines(d2,col="blue")
  lines(d3,col="red")
}
#Ultime 8 beta
for(index in 12:20){
  d1 <- density(s1[,index])
  d2 <- density(s2[,index])
  d3 <- density(s3[,index])
  mx <- max(d1$y,d2$y,d3$y)
  plot(d1,ylim=c(0,mx),xlab="Beta",ylab="Posterior density",main=colnames(x)[index],
       col="darkgreen")
  lines(d2,col="blue")
  lines(d3,col="red")
}
#verde standard, blu diversa media/varianza, rosso diverse iterazioni/burnin
plot(AMES.NNIG )
plot(AMES.NNIG2)
plot(AMES.NNIG3)
```

Chapter 7

```r
  #### JAGS LASSO PRIOR ####

  library(rjags)
  model_string.AMES.LASSO <- "model{

    # Likelihood
    for(i in 1:n){
      Y[i]   ~ dnorm(mu[i],inv.var)
      mu[i] <- alpha + inprod(X[i,],beta[])
    }

    # Prior for beta
    for(j in 1:p){
      beta[j] ~ ddexp(0,inv.var.b)
    }

    # Prior for the inverse variance
    inv.var   ~ dgamma(0.01, 0.01)
    inv.var.b ~ dgamma(0.01, 0.01)
    alpha     ~ dnorm(0, 0.01)

  }"

  DATA = list(Y=y,n=n,p=p,X=x)
  model_AMES.LASSO <- jags.model(textConnection(model_string.AMES.LASSO), data =DATA )

  update(model_AMES.LASSO, 10000)
  AMES.LASSO <- coda.samples(model_AMES.LASSO,variable.names=c("beta","alpha"),n.iter=20000)
  summary(AMES.LASSO)
```

```r
#VISUALIZZAZIONE BETA
#Prime 12 beta
s1 <- AMES.NNIG[[1]]
s2 <- AMES.LASSO[[1]]
par(mfrow=c(3,4))
for(index in 1:12){
  d1 <- density(s1[,index])
  d2 <- density(s2[,index])
  mx <- max(d1$y,d2$y)
  plot(d1,ylim=c(0,mx),xlab="Beta",ylab="Posterior density",main=colnames(x)[index],
       col="blue")
  lines(d2,col="red")
}

#Ultime 8 beta
for(index in 13:20){
  d1 <- density(s1[,index])
  d2 <- density(s2[,index])
  mx <- max(d1$y,d2$y)
  plot(d1,ylim=c(0,mx),xlab="Beta",ylab="Posterior density",main=colnames(x)[index],
       col="blue")
  lines(d2,col="red")
}
#blu NNIG, rosso LASSO
```

Distribution of the last 9 beta coefficients



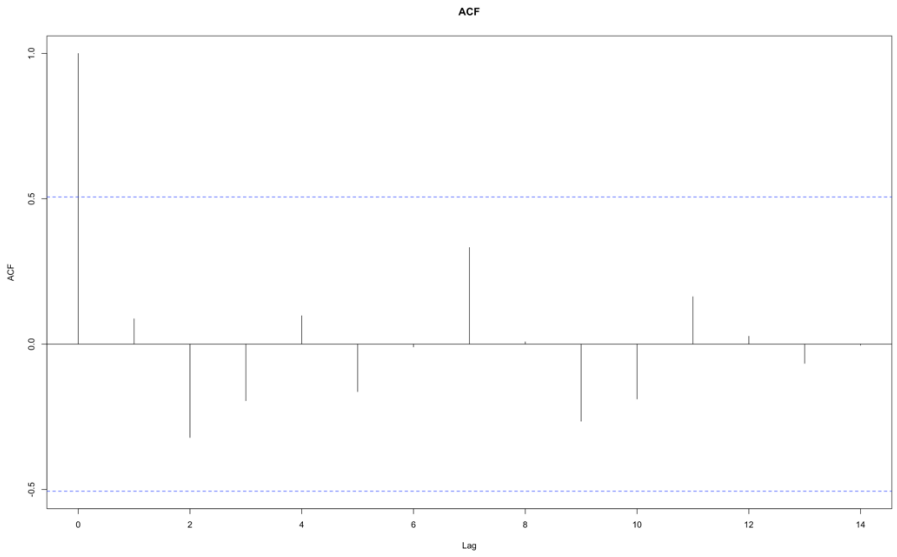
Chapter 8

```
set.seed(2020)
sample <- sample(1:nrow(X), 15)
xreg <- X[-sample,]
xreal <- X[sample,]

nreg <- nrow(xreg)
p <- ncol(xreg)

yreg <- y[-sample] #response delle prime 568 variabili
yreal <- y[sample] #response y reale dei dati
nreal <- nrow(xreal)

#Modello predictive
model_string.AMES.PREDICTIVE <- "model{

  # Likelihood
  for(i in 1:nreg){
    Y[i]   ~ dnorm(mu[i],inv.var)
    mu[i] <- alpha + inprod(xreg[i,],beta[])
  }

  # Prior for beta
  for(j in 1:p){
    beta[j] ~ dnorm(0,inv.var.b)
  }

  # Prior for the inverse variance
  inv.var   ~ dgamma(0.01, 0.01)
  inv.var.b ~ dgamma(0.01, 0.01)
  alpha     ~ dnorm(0, 0.01)

  # Predictive
  for(k in 1:nreal){
    Yp[k]   ~ dnorm(mup[k],inv.var)
    mup[k] <- alpha + inprod(xreal[k,],beta[])
  }

}"



model_AMES.PREDICTIVE <- jags.model(textConnection(model_string.AMES.PREDICTIVE),
                                    data = list(Y=yreg, nreg=nreg, nreal=nreal, p=p, xreal=xreal,
                                                xreg=xreg), n.chains=3)

update(model_AMES.PREDICTIVE, 5000)

AMES.PREDICTIVE <- coda.samples(model_AMES.PREDICTIVE,
                                variable.names=c("Yp"),
                                n.iter=10000)

su <- summary(AMES.PREDICTIVE) #sommario modello predictive
su
quantiles<-su$quantiles
quantile1<-su$quantiles[1:15, 1]
quantile2<-su$quantiles[1:15, 5]
```

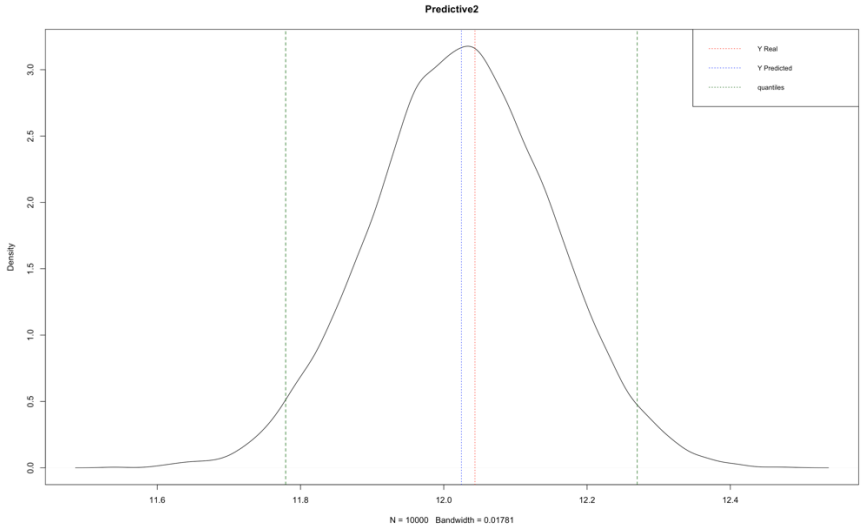
```r
#intervalli di confidenza delle predizioni
(ypredicted <- AMES.PREDICTIVE[,c('Yp[1]')][[1]])
d <- density(ypredicted)
plot(d, main="Predictive")
legend("topright",legend=c("Y Real", "Y Predicted", "quantiles"),
       col=c("red", "blue", "darkgreen"), lty=3, cex=0.8)
abline(v=yreal[1],lty=3,lwd=1, col="red") #Y REAL
abline(v=mean(ypredicted),lty=3,lwd=1, col="blue") #Y PREDICTED


abline(v=quantile1[1],lty=2,lwd=1, col="darkgreen") #INTERVALLO DI CONFIDENZA
abline(v=quantile2[1],lty=2,lwd=1, col="darkgreen") #INTERVALLO DI CONFIDENZA


par(mfrow=c(1,1))
lmpred = su$statistics[1:(nrow(x)-nreg),1]
ypred = as.matrix(lmpred) #ultime 15 response (y) del nostro modello predictive
#autocorrelation factor
acf(ypred,lag.max=50, main="ACF")


#Confronto Real vs Prediction
plot(yreal,col="red",ylim=c(11.5,13),main = "Real vs prediction",ylab="Saleprice",lwd=2)
lines(ypred,type="p", col="blue",lwd=2)
legend("topright",legend=c("Real", "Predicted"),col=c("red", "blue"), lty=3, cex=0.8)


#MSE
yest = ypred-yreal
n = length(yest)
MSE = 1/ (n - 2) * sum((yest ^ 2))
MSE


#PLOT CURVE REALI VS PRED
hist(ypred, prob=T, breaks=10, ylim = range(0, 2), xlim = range(11.2, 13))
curve(dnorm(x,mean=mean(yreal),sd=sd(yreal),log=FALSE),add=T, col='red', lwd=1)
curve(dnorm(x,mean=mean(ypred),sd=sd(ypred),log=FALSE),add=T, col='blue', lwd=1)
legend("topright",legend=c("Real", "Plug-in"),col=c("red", "blue"), lty=1, cex=0.8)
```

For the second prediction, computed using 50 observation, the code is the same.

Autocorrelation factor for the case with 15 observations



Distribution of Yp[2] in the case with 15 observations



Distribution of Yp[17] in the case with 50 observations. This case is reported because is the only one in which the real value is not included in the confidence interval.