MATH 2269
APPLIED BAYESIAN STATISTICS

Assignment 2

Prediction of Sales Prices with Multiple Bayesian Regression Model

komal mehta
S3795392

Table of Contents

1) Preview of Experiment

The Experiment performed here involves the implementation of Bayesian Multiple Regression model for predicting the property sales prices in Melbourne. For this we will using JAGS (Just Another Gibbs Sampler) software which runs under R for this predictive Analysis. The Regression model help us to find the relation of dependent and independent variables with given Bayesian prior expert Knowledge of property real estate agent and Bayesian predictors along with Bayesian estimates.

The data given here consists of sample size of 10000 which is huge sample to process. Data comprises of 6 features out of which Sale Price is the dependent feature and Area, Bedrooms, Bathrooms, Car Parks and Property type are the independent features or predictors. As there are more than one independent predictors, therefore we are using Bayesian multiple Linear Regression model. The equation of the model with their respective coefficients is given by:

$$\mathbf{SalePrice=\beta_0+ \beta_1 Area+ \beta_2 Bedrooms+ \beta_3 Bathrooms + \beta_4 CarParks+ \beta_5 PropertyType +\varepsilon,}$$

Here, $\boldsymbol{\varepsilon \sim Gamma(\alpha, \beta)}$ is the error rate with $\boldsymbol{\beta_0, \beta_1, \beta_2, \beta_3, \beta_4}$ and $\boldsymbol{\beta_5}$ are the coefficients of the independent predictors. For the predictive Analysis, we are going to define JAGS model diagram and model string for defining the model. Next, we will specify the Priors for the independent features which will help us in formation of posterior distribution which will help in determining the significance of model. After specifying the priors its time to go for MCMC diagnostics with different MCMC settings to identify the responsiveness, accuracy and efficiency of the chains formed. Also, after getting best MCMC setting and perfect posteriors model performance is Analysed and reported with some insights coming up with suitable recommendations. Lastly, the approach for efficient Analysis for the prediction of Sale Prices is described.

Lets, start with the descriptive Visualisation of data.

2) Data Descriptive Visualisation

In this section, we will be analysing the relation between dependent and independent features with different plots. Firstly, for Bayesian Regression Analysis we should know what type of distribution dependent variable which is Sale Price in this case it follows. So, the histogram and Kernel density plot of the dependent variable is given below:

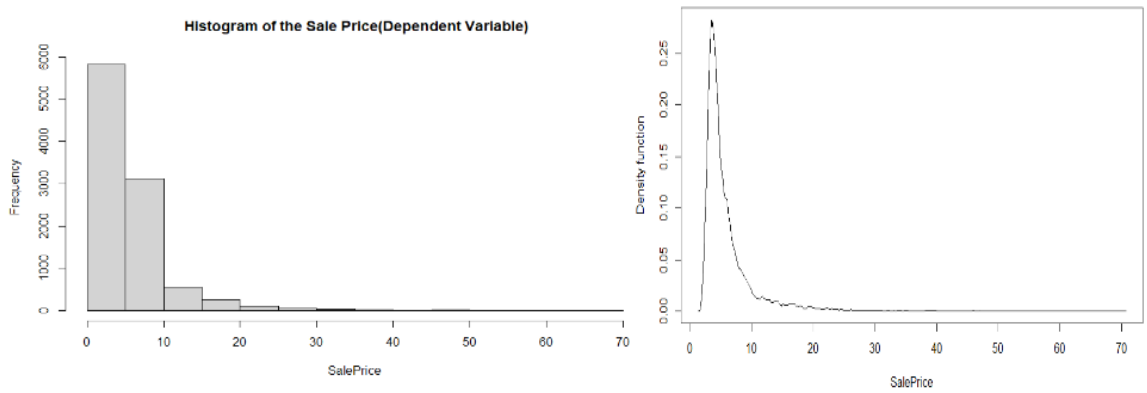


Fig 2.1 Histogram and Kernel density curve (For Code Refer to Appendix C)

From the above plots, it is evident that dependent variable Sale Price is having a long right tail in the distribution so choosing a gamma model for Sale price would be a appropriate option. Overall, model

implemented for this experiment is normal-gamma model as all the independent are following normal or uniform distributions . Next, we are forming correlation matrix to check the correlation between all the independent variables to check the correlation among them which is given as below:

	Area	Bedrooms	Bathrooms	CarParks	PropertyType
Area	1.000	-0.270	-0.087	-0.096	0.32
Bedrooms	-0.270	1.000	0.538	0.435	-0.56
Bathrooms	-0.087	0.538	1.000	0.366	-0.29
CarParks	-0.096	0.435	0.366	1.000	-0.36
PropertyType	0.320	-0.560	-0.290	-0.360	1.00

Fig 2.2 Correlation Matrix of Independnet features (For Code Refer to Appendix C)

From the correlation matrix itself , we can conclude that the correlation of independent feature with itself is the strongest and also we can observed that bedrooms, bathrooms have the strong correlation with other independent variables. After checking the correlation of independent with themselves, its time to go for verifying the distribution independent variables with dependent variables.

The correlation of Sale Price is checked with all the independent variables through scatter plots which is shown as below:

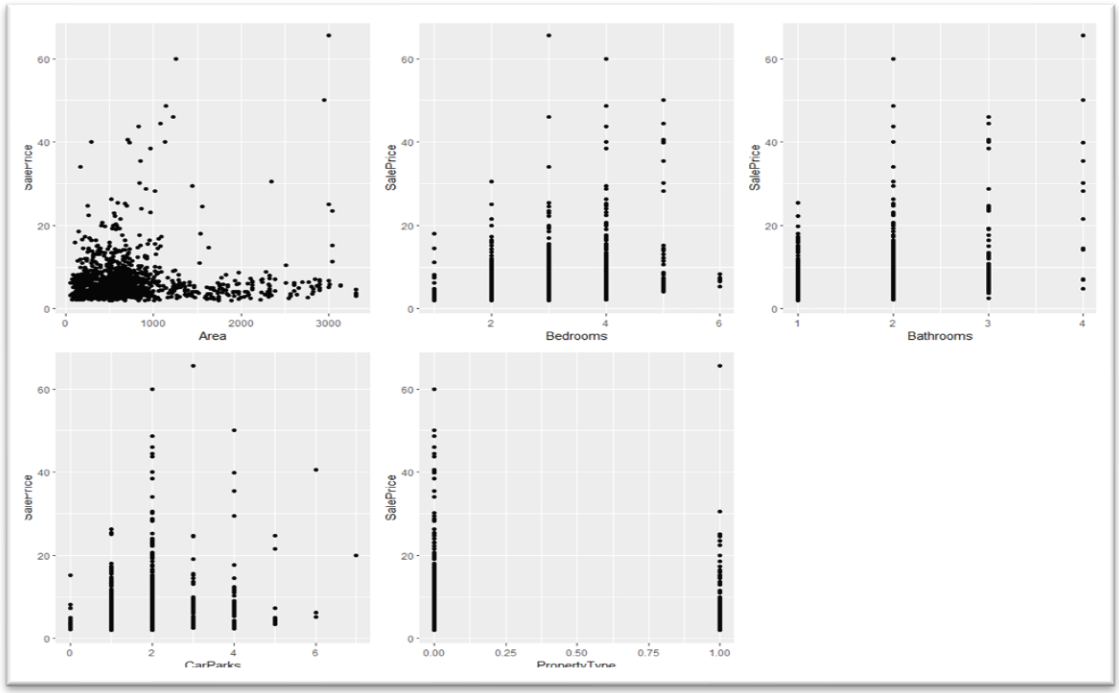


Fig 2.3 Scatter plots of Independent Variables (For Code Refer to Appendix C)

From the above scatter plots, it can be concluded that all the independent variables follow non-linear relationship with Sale Price which defines that there is no strong correlation between Independent variables and dependent variables. The independent variable is having different distributions because of which there is distortion in linearity such as Area having continuous

distributions, Bedrooms, Bathrooms, Carparks following discrete distributions while Property type is having binomial 0 and 1 distributions.

After having a descriptive visualisation of both independent and dependent variables, its time to define and implement model.

3) JAGS model Definition and Implementation

After having a descriptive look of the data, its time to define the model and implement it on the data. Firstly, we are defining the JAGS model diagram as defined below for Bayesian multiple regression model implementation:

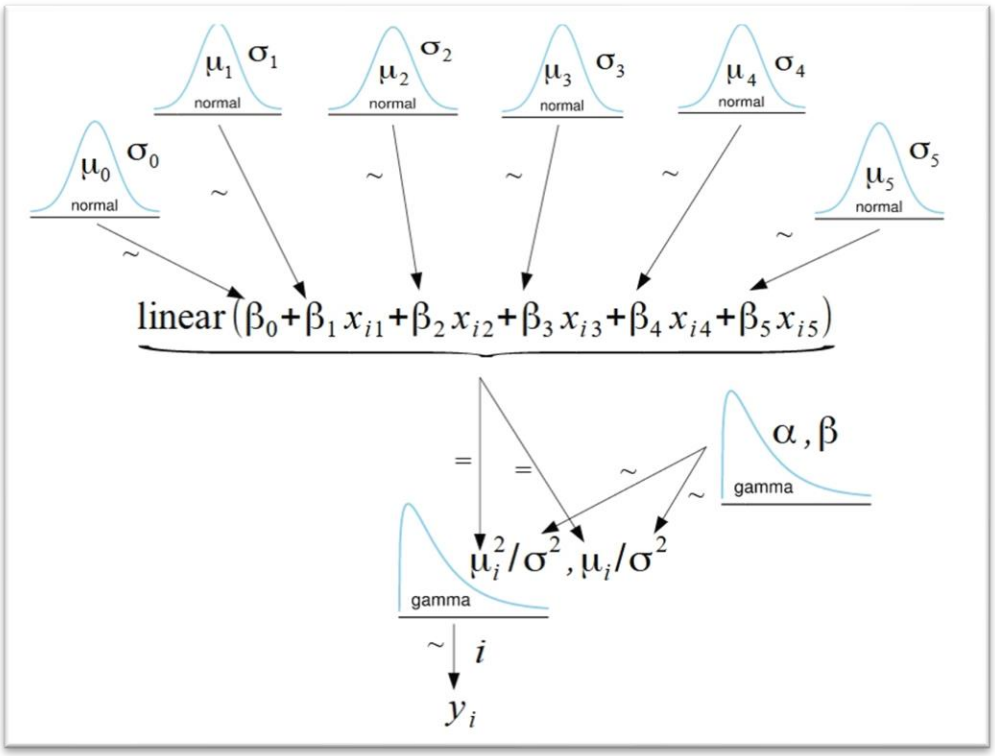


Fig 3.1 Model Diagram

In this model diagram we are using 6 normal prior distribution comprising of **β1 to β5 for** 5 independent variables and **β_0** for Intercept. The prior distributions for these coefficients can be described in JAGS model implementation as given below:

β_0~dnorm (μ_0, σ_0)

β_1~dnorm (μ_1, σ_1)

β_2~dnorm (μ_2, σ_2)

β_3~dnorm (μ_3, σ_3)

β_4~dnorm (μ_4, σ_4)

β_5~dnorm (μ_5, σ_5)

dnorm signifies the normal distributions used for the coefficients of the priors and M_0 are the priors associated mean and S_0 to S_5 is prior variance values represented as 1/variance. These values are priors from the expert knowledge required to construct the posterior for the predicted Sale Price. These priors form a Linear Regression Line equation which is calculated as mean prior represented through this:

$$\mu=\beta_0+ \beta_1x_1+ \beta_2x_2+ \beta_3x_3 + \beta_4x_4+ \beta_5x_5 +\varepsilon,$$

Here, x_1, x_2, x_3, x_4, x_5 represents the predictors such as Area, Bedrooms, Bathrooms, Carparks and Property Type, respectively. They have a uniform distribution falling in the range of $(-\infty,+\infty)$ for which normal distribution are defined. Next, we are using Gamma (α, β) to calculate the variance prior to be used later. Along with these mean and variance priors we are using likelihood which represents data through Gamma (μ^2/σ^2, μ/σ^2) distribution model. The μ and σ is calculated above through given priors. Here, mean will be depicting the estimated predicted value and variance will determine the amount of belief an analyst has. As we have seen from the aforementioned scatter plots that the correlation between Sale Price and Other variables is falling in the range of $(0,+\infty)$ and all are having positive distribution so, normal – gamma model is the appropriate model to used .

Y_i is the final posterior distribution of the Sale Price predicted after Bayesian Regression Model Analysis. In this following Model Regression Analysis, we are taking this model diagram as a reference to calculate the Bayesian estimate which is usually mode for which various distributions are defined. The following distributions will depict the significance of prior information in terms of posterior plots and at the end of Analysis these predictors posterior will help in prediction of Sales Prices.

For the initiation of Bayesian Regression Analysis, firstly standardisation is done then afterwards we are defining model in the form of function. Both of these steps are recollected in one model string which can be found in Appendix D.

After forming the JAGS model diagram and implementing in model String its time to specify the prior information which is assessed through different criteria to check for sensitivity on different posterior distributions.

4) Specification of Priors

In this section, we will be defining the priors for variance(σ^2), the intercept(β_0) and other 5 coefficients (β1, β2, β3, β4, β5) for parameters such as Area, Bedrooms, Bathrooms, Car Parks and Property type, respectively. After specifying these priors, sensitivity is checked on different posteriors through different variances or beliefs value.

For Constructing the posterior for the given priors, we are taking a sample of 1500 from whole data consisting of 10000 records. Also, the MCMC settings for the formation of these posterior is given below:

Number of Chains:3

Number of Adaptations:1000

Number of Burn-in Steps: 100

Number of Thin Steps:65

Number of Saved Steps:1700

These settings are concluded after a number of iterations by changing thin steps from 15,23,33,48 to 65 and later number of saved steps from 1000 to 1200 to 1500 to 1700. Also, before feeding these priors into the regression model the data is standardised with the following priors.

So, the prior distribution specifications are defined below:

A) For Intercept β_0

From the model diagram, the prior distribution is normal curve which is defined below:

$$\beta_0 \sim dnorm(\mu_0, \sigma_0)$$

Here, μ_0 represents mean and σ_0 signifies variance for the distribution. As β_0 signifies the intercept, therefore it does not have association with any other features which states that no expert information could be given for this. Therefore, because of no- expert information on this, the coefficient β_0 is having distribution which is non-informative in nature. The Mean (μ_0) is taken as 0 and Variance (σ_0) is taken as 4. There is no standardisation performed on this prior and posterior constructed out of this prior will be non-informative in nature.
The posterior distribution for β_0 is given below:

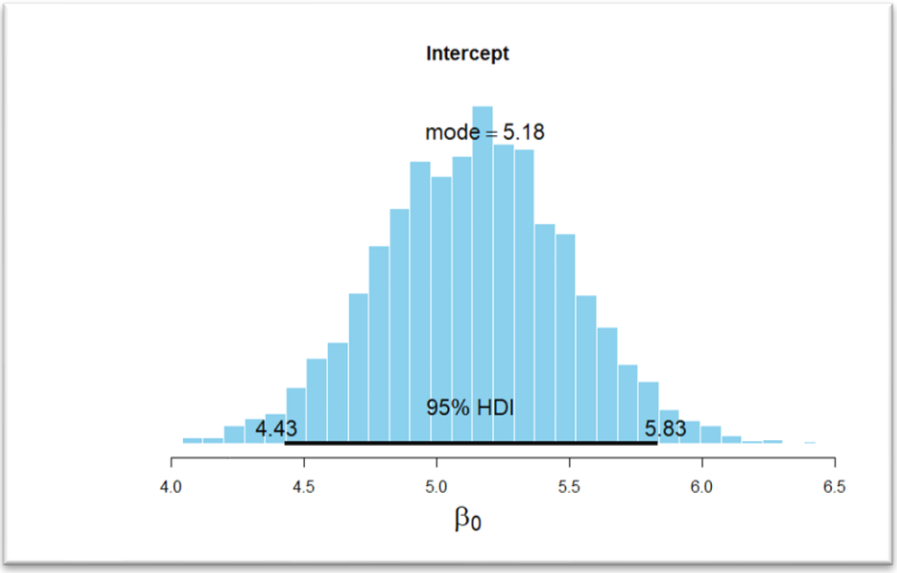


Fig 4.1 Posterior distribution of Intercept (For Code refer to Appendix B)

From the above posterior distribution, it is observed that the Bayesian estimate for this intercept is 5.18 having 95% HDI interval range [4.43,5.83]. For this coefficient we cannot check the sensitivity as it is non- informative in nature because of we do not have prior Mean and variance.

B) For Area Coefficient β_1

From the model diagram, the prior distribution is normal curve which is defined below:

$$\beta_1 \sim dnorm(\mu_1, \sigma_1)$$

Here, μ_1 represents mean for Area Independent feature and σ_1 signifies variance for the Area predictor feature. For β_1 signifying the Area, we are provided with the prior information as every m^2 increase in land size increases the sales price by 90 AUD given with very strong belief.

Because, of its very strong belief the variance is taken as 0.9 which is low variance having more concentrated values. Also, there is scale of 1:10000 therefore the mean prior for this coefficient is taken as 0.0009. The distribution followed for this coefficient is highly informative in nature.

The posterior distribution for β_1 is given below:

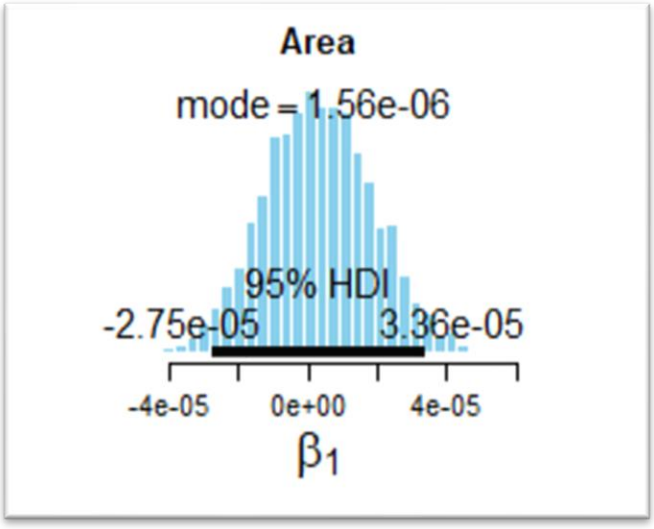


Fig 4.2.1 Posterior distribution of Area (For Code refer to Appendix B)

From the above posterior distribution, it is observed that the Bayesian estimate for this coefficient is 1.56e-06 having 95% HDI interval range [-2.75e-05,3.36e-05]. To access the sensitivity of the posterior distributions the variance is changed from 0.9 to 0.2 with same mean prior 0.0009. With these priors, we will have this following posterior distributions:

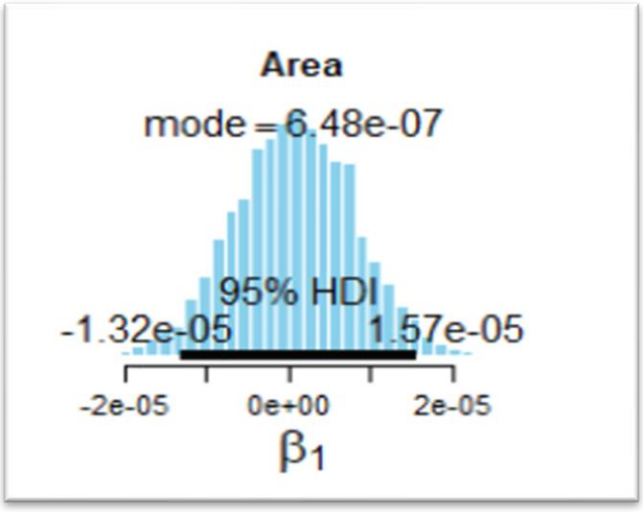


Fig 4.2.2 Posterior distribution of Area with updated variance (For Code refer to Appendix B)

From the above posterior distribution, we can estimate that the Bayesian estimate that is mode is changed from 1.56e-06 to 6.48e-07 which shows if we increases the variance the posteriors distributions are also affected exponentially. Also, the 95% HDI range is changed from [-2.75e-05,3.36e-05] to [-1.32e-05,1.57e-05]. The Bayesian estimate signifies for every unit increase in m^2 unit of Area the Area increases by 6.48e-07.

C) For Bedrooms Coefficient β_2

From the model diagram, the prior distribution is normal curve which is defined below:

$$\beta_2 \sim dnorm(\mu_2, \sigma_2)$$

Here, μ_2 represents mean for Bedrooms Independent feature and σ_2 signifies variance for the Bedrooms predictor feature. For β_2 signifying the Number of Bedrooms, we are provided with the prior information as Every additional bedroom increases the sales price by 100,000AUD given with very weak expert knowledge. Because, of its very weak belief the high variance is taken as 15 having less concentrated, more spread values. Also, there is scale of 1:10000 therefore the mean prior for this coefficient is taken as 1. The distribution followed for this coefficient is highly informative in nature.

The posterior distribution for β_2 is given below:

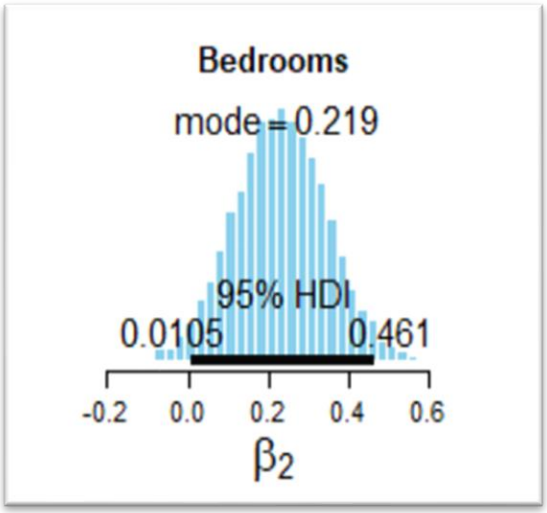


Fig 4.3.1 Posterior distribution of Bedrooms (For Code refer to Appendix B)

From the above posterior distribution, it is observed that the Bayesian estimate for this coefficient is 0.219 having 95% HDI interval range [0.0105,0.461]. To access the sensitivity of the posterior distributions the variance is changed from 15 to 30 with same mean prior 1. With these priors, we will have this following posterior distributions:

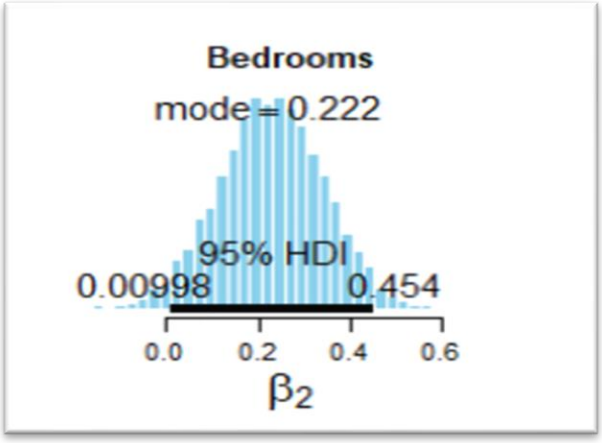


Fig 4.2 Posterior distribution of Area with updated variance (For Code refer to Appendix B)

From the above posterior distribution, we can estimate that the Bayesian estimate that is mode is changed from 0.219 to 0.222 which shows if we increase the variance the posteriors distributions are also affected exponentially. Also, the 95% HDI range is changed from [0.0105,0.461] to [0.00998,0.454]. The Bayesian estimate signifies for every unit increase in number of bedrooms the number of bedrooms increases by 0.222.

D) For Bathrooms Coefficient β_3

From the model diagram, the prior distribution is normal curve which is defined below:

$$B_3 \sim dnorm(\mu_3, \sigma_3)$$

Here, μ_3 represents mean for Bathrooms Independent feature and σ_3 signifies variance for the Bathrooms predictor feature. For β_3 signifying the Number of Bathrooms, we are not provided with no prior expert knowledge. Because, of its distribution followed for this coefficient is highly non - informative in nature. The mean prior taken in this case is zero and variance is taken as 4 to describe the non-informativeness.

The posterior distribution for β_3 is given below:

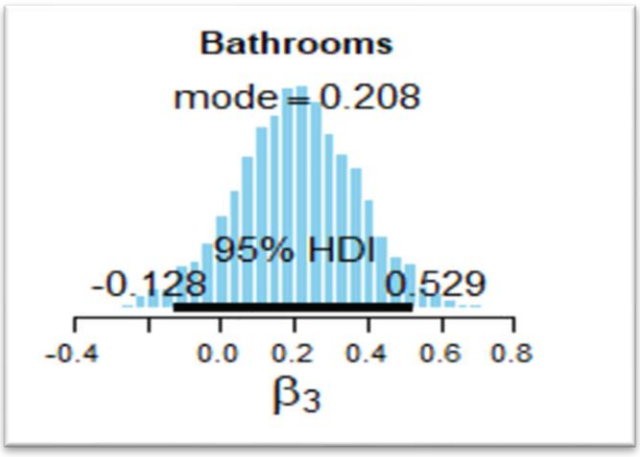


Fig 4.4.1 Posterior distribution of Bathrooms (For Code refer to Appendix B)

From the above posterior distribution, it is observed that the Bayesian estimate for this coefficient is 0.208 having 95% HDI interval range [-0.128,0.529].

The Bayesian estimate signifies for every unit increase in number of Bathrooms the number of bathrooms increases by 0.208.

E) For Car Parks Coefficient β_4

From the model diagram, the prior distribution is normal curve which is defined below:

$$B_4 \sim dnorm(\mu_4, \sigma_4)$$

Here, μ_4 represents mean for Car Parks Independent feature and σ_4 signifies variance for the Car Parks predictor feature. For β_4 signifying the Number of Car Parks, we are provided with the prior information as Every additional car space increases the sales price by 120,000AUD given with strong expert knowledge. Because, of its strong belief the medium variance is taken as 4 having more spreaded values. Also, there is scale of 1:10000 therefore the mean prior for

this coefficient is taken as 1.2. The distribution followed for this coefficient is highly informative in nature.

The posterior distribution for β_4 is given below:

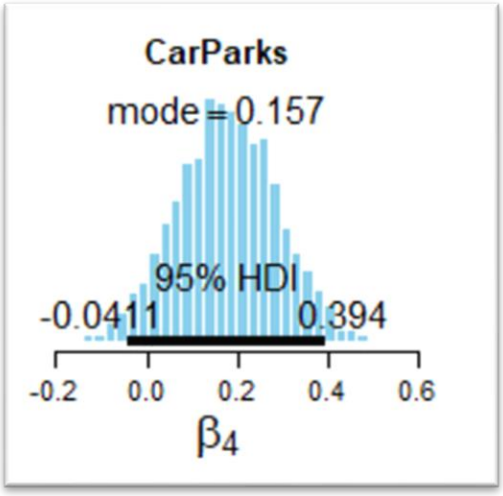


Fig 4.5.1 Posterior distribution of Car Parks (For Code refer to Appendix B)

From the above posterior distribution, it is observed that the Bayesian estimate for this coefficient is 0.157 having 95% HDI interval range [-0.0411,0.394]. To access the sensitivity of the posterior distributions the variance is changed from 4 to 2 with same mean prior 1.2. With these priors, we will have this following posterior distributions:

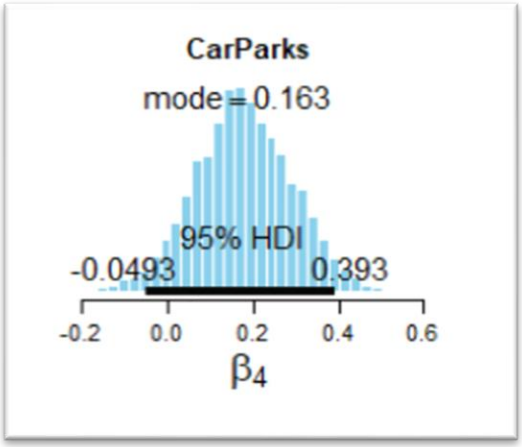


Fig 4.5.2 Posterior distribution of Car Parks with updated variance (For Code refer to Appendix B)

From the above posterior distribution, we can estimate that the Bayesian estimate that is mode is changed from 0.157 to 0.163 which shows if we increase the variance the posteriors distributions are also affected exponentially. Also, the 95% HDI range is changed from [-0.0411,0.394] to [-0.0493,0.393]. The Bayesian estimate signifies for every unit increase in number of Car Parks the number of bedrooms increases by 0.163.

F) For Property Type Coefficient β_5

From the model diagram, the prior distribution is normal curve which is defined below:

$$B_5 \sim dnorm(\mu_5, \sigma_5)$$

Here, μ_5 represents mean for Property type Independent feature and σ_5 signifies variance for the Property type predictor feature. For β_5 signifying the Property type which is having 0 and 1 levels for House and Unit respectively, we are provided with the prior information as if the property is a unit, the sale price will be 150,000 AUD less than that of a house on the average given with very strong expert knowledge. Because, of its very strong belief the high variance is taken as 0.9 having less spreaded, more concentrated values. Also, there is scale of 1:10000 therefore the mean prior for this coefficient is taken as -1.5. The distribution followed for this coefficient is highly informative in nature.

The posterior distribution for β_5 is given below:

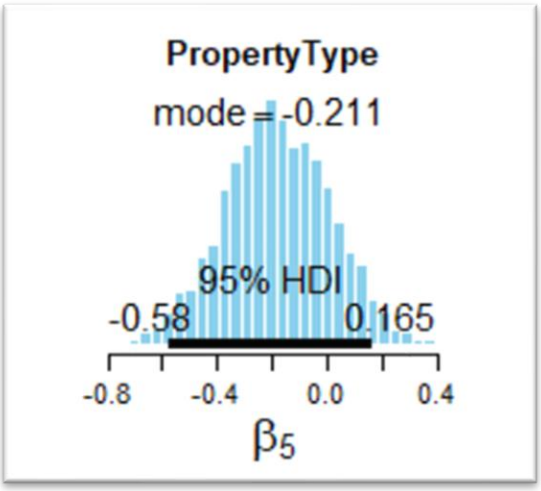


Fig 4.6.1 Posterior distribution of Property type (For Code refer to Appendix B)

From the above posterior distribution, it is observed that the Bayesian estimate for this coefficient is -0.211 having 95% HDI interval range [-0.58,0.165]. To access the sensitivity of the posterior distributions the variance is changed from 0.9 to 0.2 with same mean prior -1.5. With these priors, we will have this following posterior distributions:

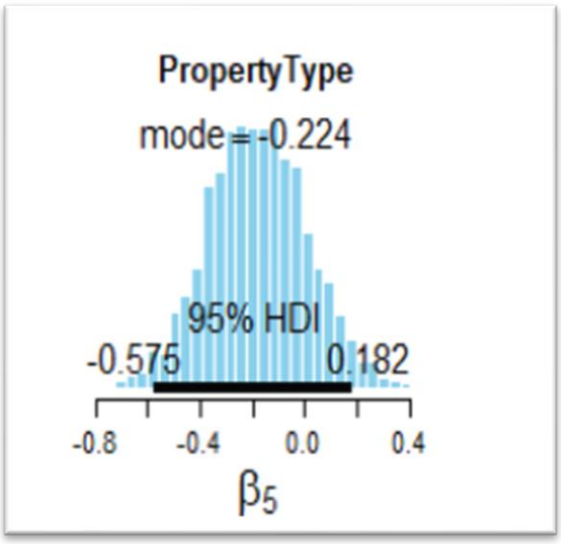


Fig 4.6.2 Posterior distribution of Property type with updated variance(For Code refer to Appendix B)

From the above posterior distribution, we can estimate that the Bayesian estimate that is mode is changed from -0.211 to -0.224 which shows if we increase the variance the posteriors distributions are also affected exponentially. Also, the 95% HDI range is changed from [-0.58,0.165] to [-0.575,0.182]. The Bayesian estimate signifies for every unit increase in property type the property type decreases by 0.224.

G) For variance/Scale(σ^2)

From the aforementioned model diagram, the prior distribution of the variance is given by this equation:

$$\sigma^2 \sim \text{Gamma}(\alpha, \beta)$$

where α is known as shape and β is known as rate of the Gamma distribution. We are given with no prior expert knowledge for the variance so, we can say that its distribution is highly non- informative in nature. Therefore, we have taken mean as 0.1 and variance as 0.1 in this case. Using these priors, the posterior distribution can be seen as below:

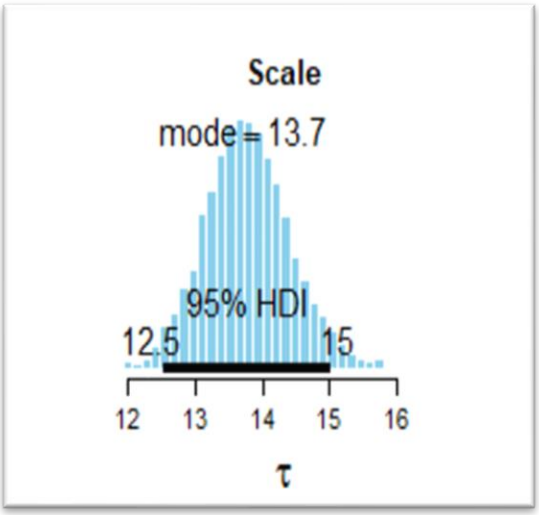


Fig 4.7 Posterior distribution of Scale (For Code refer to Appendix B)

From the above posterior distribution, the Bayesian estimate which is mode is 13.7 and 95% HDI is given as [12.5,15]. The variance is also represented by τ.

5) MCMC Diagnostic Check

After specifications of the priors and constructions of posteriors from these given priors, its time for the constructions of Chains and check for diagnostics and later interpretating the results. After the Chains are constructed through MCMC settings, the representativeness, accuracy, and efficiency of the Chains are analysed. MCMC diagnostics is checked for all the coefficients of all the parameters. For improving the representativeness and accuracy, the MCMC setting like Number of Thin Steps, Length of Chains, Number of Adaptations and Number of Burn- in steps and Number of Saved Steps are modified. Also, efficiency could be checked in the later stage for performance Analysis.

After the observation from the posterior distributions of coefficients β0, β1, β2, β3, β4, β5, the MCMC diagnostics is performed on the chains formed through MCMC JAGS sampler method to estimate these coefficients. Here, we are considering a sample of 1500 for the performing MCMC diagnostic checks as previously used for constructing the posterior distributions as processing the huge data sample 0f 10000 can make the system crash and there is high computation for the process. Initially, the below mentioned MCMC settings are used to check for diagnostics for all the coefficients:

Number of Chains:3

Number of Adaptation Steps:1000

Number of Thin Steps:15

Number of Burn-in Steps:100

Number of Saved Steps:1200

With these initial settings, the diagnostics obtained as below:

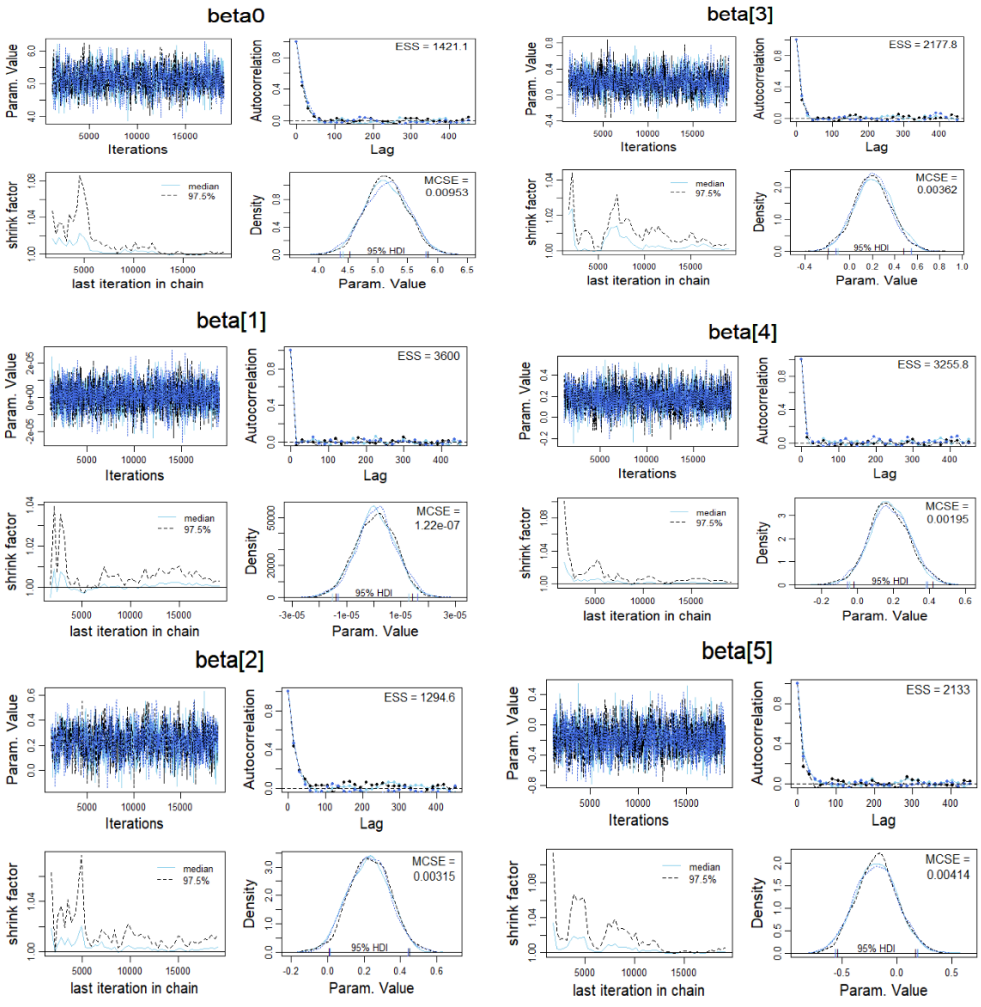


Fig 5.1 Diagnostic check for all parameters (For Code refer to Appendix F)

Analysing the chains representativeness and accuracy of the above diagnostics we can say that the shrink factor of β_0 and β_1, β_2, β_3, β_4, β_5 are below 1.2 which is good for representativeness of the chains.

Next, the chains are blending well with each other and converges at the mean level. The autocorrelation is still there which shows bad representatives of the chains which can be improved by increasing the thinning steps which means increasing the number of iterations, hence there-forth increasing the number of saved steps. Also, the density plots are overlapping with each other which is good.

Secondly, the appropriateness of the Chains can be verified through accuracy criteria as stated ahead. From the diagnostic checks, the ESS which is estimated Sample Size is good for β_1, β_3, β_4, β_5 with about 2000 to 3000 range nearly closer to 10,000 which is expected optimal value for ESS while ESS is really poor for β_0, β_2 with values 1421 and 1294 respectively. As ESS is directly related to Number of Saved Steps so, increasing the saved steps ESS is elevated to a great extent which could be equal to 10000 data points value. As MCSE (Monte Carlo Standard Error) is related to ESS by this equation:

MCSE = SD/ESS

We can say that increasing the number of saved steps increases ESS hence, decreases the MCSE which will be nearly equals to 0 which is considered as good estimate. Also, from the above diagnostic, it is observed that MCSE is good for all the coefficients as it is nearly equals to zero.

So, to improve the representativeness and overall accuracy of the chains formed, the MCMC settings are hyper tuned such as Number of Saved Steps and Thinning Steps. Firstly, we are increasing the number of thinning steps from 15 to 48 and Number of Saved Steps from 1200 to 1700 with 1500 Samples. Increasing the Thinning steps improves the autocorrelation between data points of the chains which is seen as nearly equals to 0 and converging at the mean level for all the coefficients β_0 and β_1, β_2, β_3, β_4, β_5. Also increasing the Number of Saved Steps, the ESS is increased reaching nearly up to 10000. The diagnostic check obtained through these hyper- tuned parameters is given below:

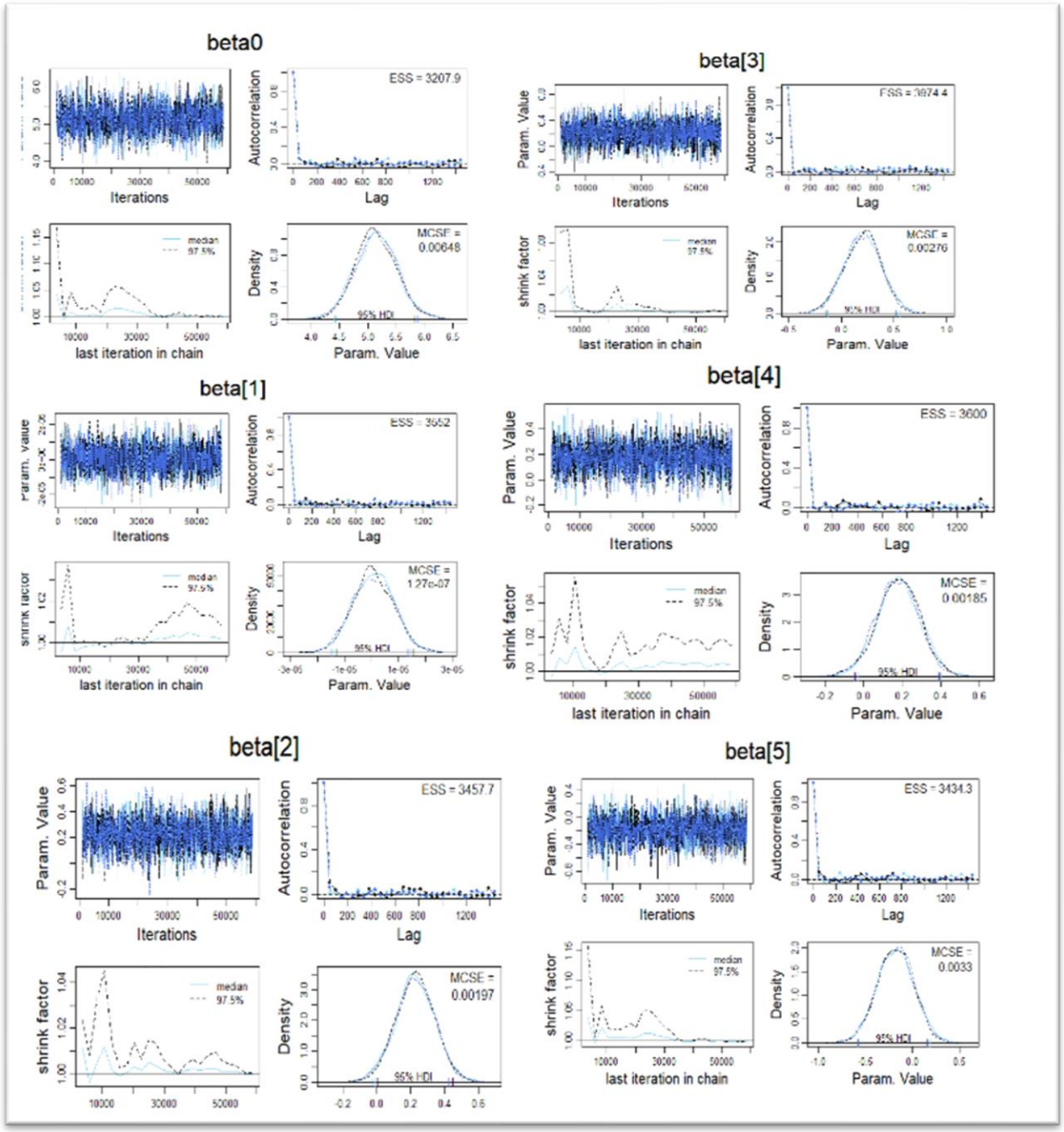


Fig 5.2 Diagnostic check for all the parameters with tuned MCMC settings (For Code refer to Appendix F)

The above diagnostic check after hyper-tuning of the MCMC settings, there is good representativeness and accuracy is obtained with no autocorrelation, Maximum ESS, Low MCSE, density plots overlapped, shrink factor under 1.2. These results depict that the improving the MCMC settings is not affecting the diagnostic negatively. Also, chains for all coefficients are observed to be converging at the mean level.

Also, for the variance diagnostic check is observed with these MCMC settings:

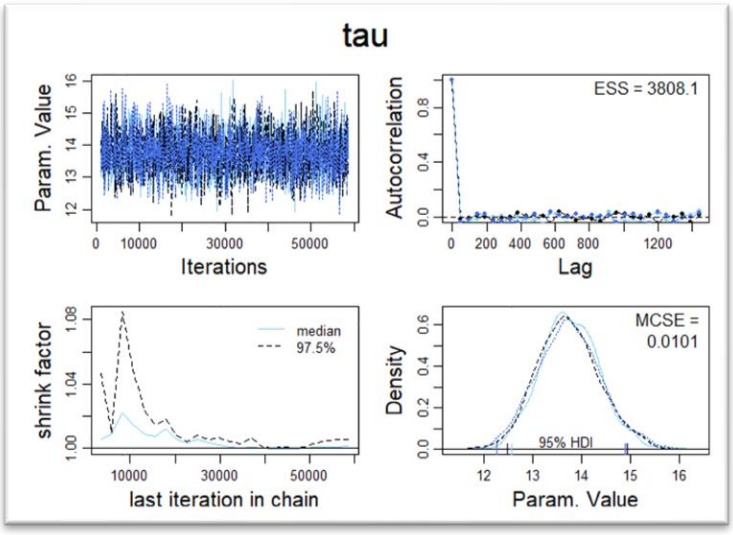


Fig 5.3 Diagnostic check for variance with tuned MCMC settings (For Code refer to Appendix F)

From the above diagnostic check, it is evident that the shrink factor is really good which is under 1.2. Also, the chains are appeared to converge at the mean level. There is no- autocorrelation seen between the data points. Also, the density plots are overlapping with each other. So, it is evident that the chains for variance have good representativeness. Also, the accuracy of the chains is really good as ESS is high nearly equal to 3808 and MCSE equals to 0. Also, the HDI intervals in the density curve are overlapping which is a good indication of fact that chains are highly representative in nature.

After analysing the representativeness and accuracy of the chains, efficiency is also measured for appropriateness of the chains and performance analysis of MCMC JAGS sampler method. Firstly, when diagnostic checks is scheduled through sequential run then its computation power is less. But when the diagnostics is performed through parallel run then its computation is increased by 40% than the sequential run. Also, the parallel computation in MCMC JAGS sampler method can be elevated through accelerating the GPU cores with CUDA (Compute Unified Device Architecture). In this approach, Interface function is written in C to connect between R and C. Also, Fortran codes are written to run the simulation of MCMC faster using R packages like gputools and cudaBayesreg along with clufft.r. r files. Also, jags. parfit function of dclone library can be used for parallel computing through formation of clusters and cloning the data into clusters which increases the efficiency.

6) Results and Outcomes

After retrieving the optimal value of variance different iterations are performed on 10000 data sample with different MCMC settings for the Bayesian Multiple regression model stated above. Also, we will be calculating the best Posterior with good Bayesian estimate and good HDI interval using different predictors values given for Independent features such as Area, Bedrooms, Bathrooms, Carparks and Property Type. Also, the appropriateness of chains are checked for different prediction settings.

6.1 Parameters and Predictors Bayesian Estimates Calculation

In this section, Bayesian estimates for variance(σ^2), the intercept(β_0) and other 5 coefficients (β_1, β_2, β_3, β_4, β_5) for parameters such as Area, Bedrooms, Bathrooms, Car Parks and Property type,

respectively are explored here. Later, the Sales Price is predicted from different values of Independent features. The prior distributions for all of the parameters with their respective mean and variance is given below:

σ^2 ~Gamma(0.001, 0.001)

β_0~dnorm (0,4)

β_1~dnorm (0.0009, 0.2)

β_2~dnorm (1, 30)

β_3~dnorm (0, 4)

β_4~dnorm (1.2, 2)

β_5~dnorm (-1.5, 0.2)

Here, except β_0, β_3 which are non- informative in nature all other coefficients such as β_1, β_2, β_4, β_5 are standardised and assessed for sensitivity with different degree of beliefs. After these tuned priors, we have to update the best settings of MCMC JAGS Sampler method for getting the best appropriateness of the chains in terms of accuracy and representativeness. The MCMC settings changed are mentioned below as:

Number of Adaptation Steps: 1800

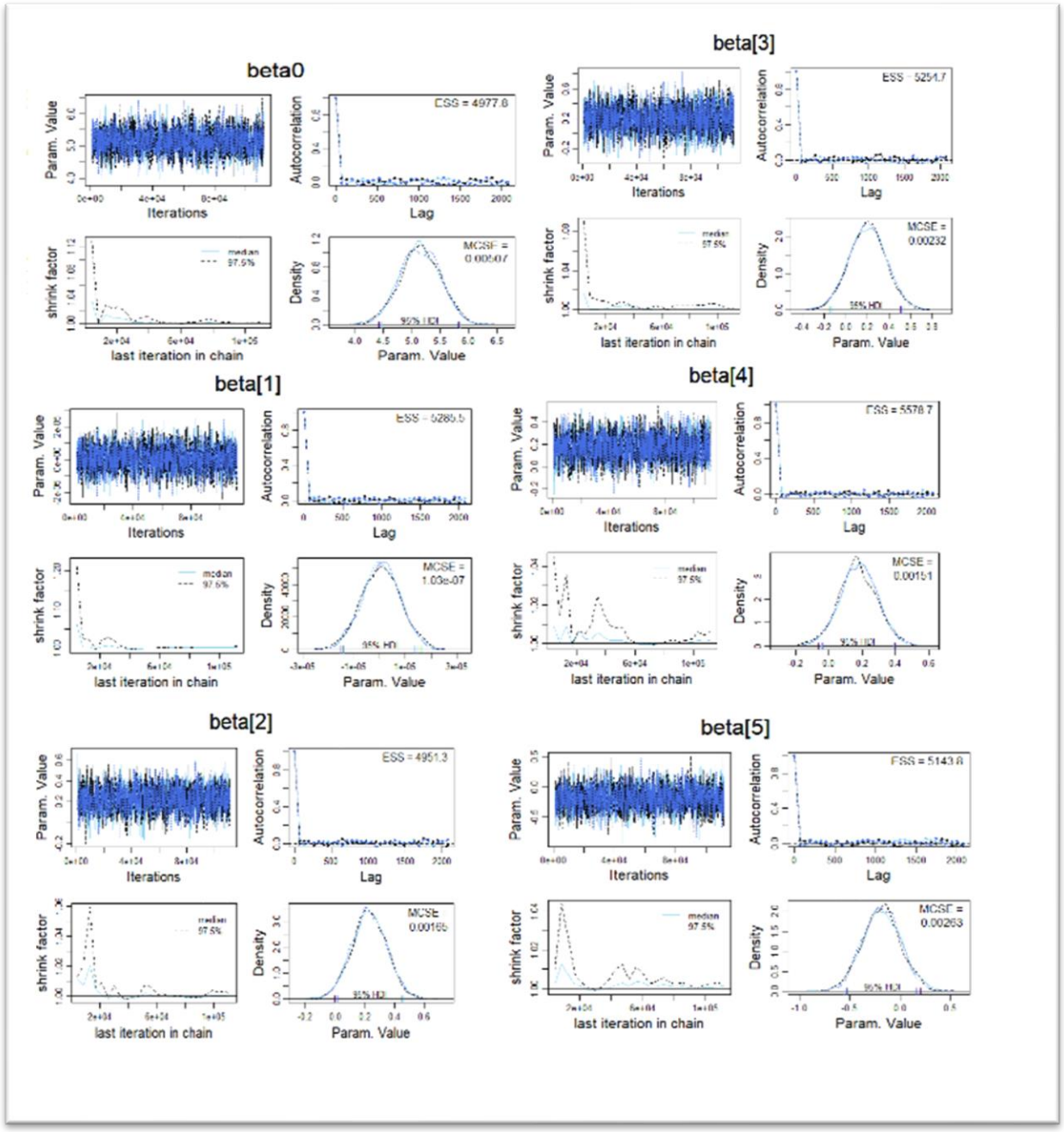
Number of Thin Steps: 65

Number of Saved Steps: 1700

Number of Burn- in steps: 6000

Length of the Chains: 5

The model String is executed through coda samples on the afore-mentioned priors' distributions along with these above mentioned MCMC settings to get this below mentioned diagnostic check:



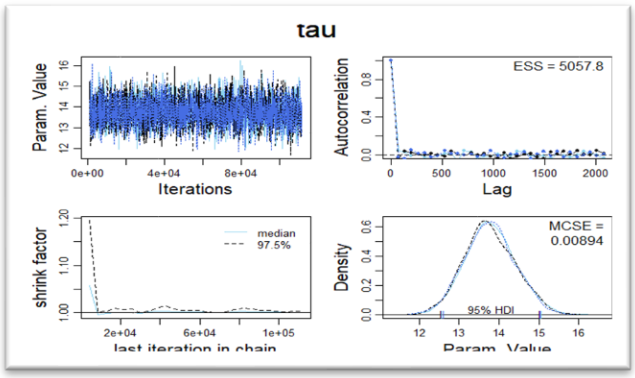


Fig 6.1.1 Diagnostic check for all the parameters with tuned MCMC settings (For Code refer to Appendix F)

From the above output, the representativeness and accuracy are observed really good. Firstly, when we see the shrink factor it is seen as below 1.2 which is good. Also, the chains are converging at the mean level and blending well with each other. Also, the autocorrelation is not present here between the data points which is a good estimate. Also, the density curve is having HDI intervals overlapping with each other which is also good. In terms of accuracy, the chains are depicting high ESS of 5500 to 6000 which is really good while decreasing the MCSE to nearly 0. After these diagnostics check, the posterior distributions of these distributions are interpreted as below:

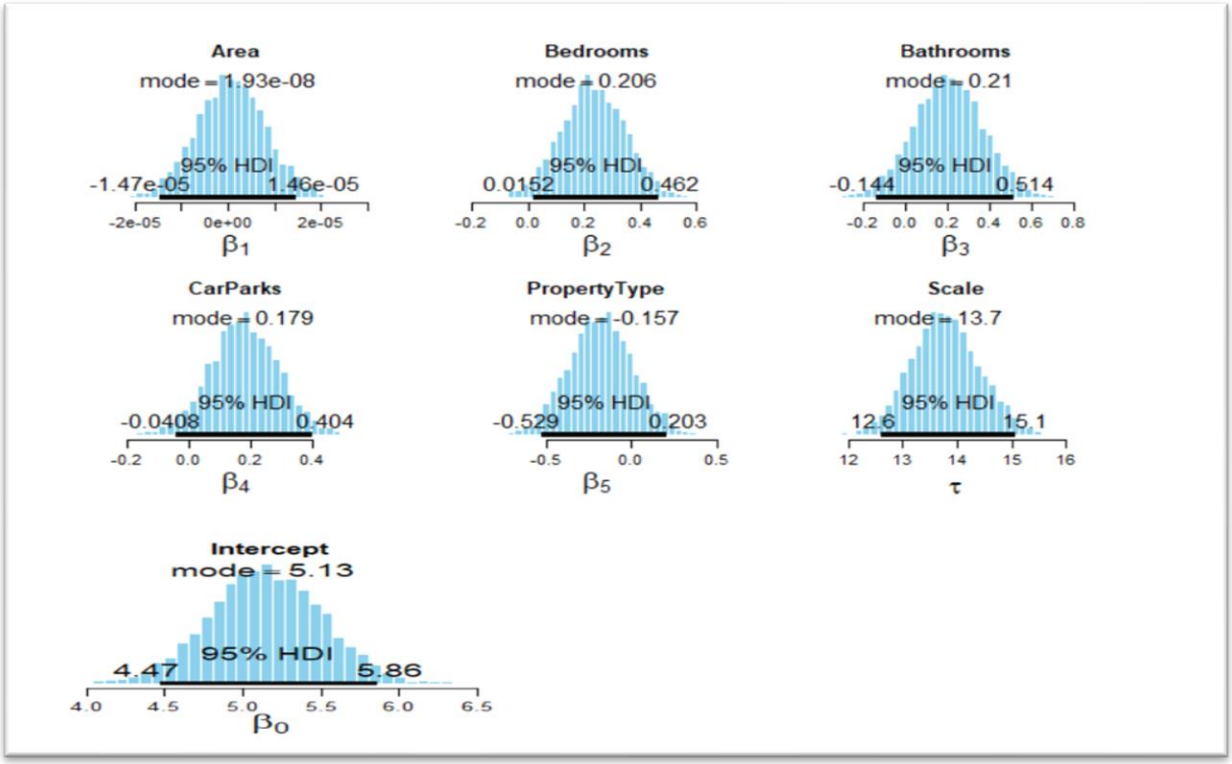


Fig 6.1.2 Posterior distribution for all the parameters with tuned MCMC settings (For Code refer to Appendix F)

From the above posterior distribution of all the parameters with the above MCMC settings, we got some insights from Bayesian estimates and HDIS. Bayesian estimate define the increase or decrease in the value as settings are changed. Let us explore this interpretation of these parameters:

σ^2 -> The distribution of the Sale Price will have variance of1370000 AUD.

B_0 -> If expert knowledge is not given, the sales prices will be 5,13,000 AUD.

B_1 -> if expert knowledge is given, then for every unit increase in the area of property, the sale price increases by 19.3 cents

B_2-> if expert knowledge is given, then for every unit increase in the number of bedrooms, the sale price increases by 20,600 AUD

B_3 -> if expert knowledge is not given, then for every unit increase in the number of bathrooms, the sale price will be 21,000 AUD

β_4 -> if expert knowledge is given, then for every unit increase in the number of Car Parks, the sale price increases by 17,900 AUD

β_5 -> if expert knowledge is given and if the property type given is unit, the sale price will be 15,700 AUD less than that of a house property.

Also, from the posteriors, it can seen that all the independent features are significant as 0 is lying on the outside part closer to HDIs.

After analysing the Bayesian estimates of all the parameters the different predictors are given for Area, bedrooms, Bathrooms, Car Parks and Property type to predict the sales prices of properties with the below given predictors as :

Property No	Area	Bedrooms	Bathrooms	CarParks	PropertyType
1	600	2	2	1	Unit
2	800	3	1	2	House
3	1500	2	1	1	House
4	2500	5	4	4	House
5	250	3	2	1	Unit

Table 6.1: Predictors values for Independent features

The posterior distributions with above predictors is given below:

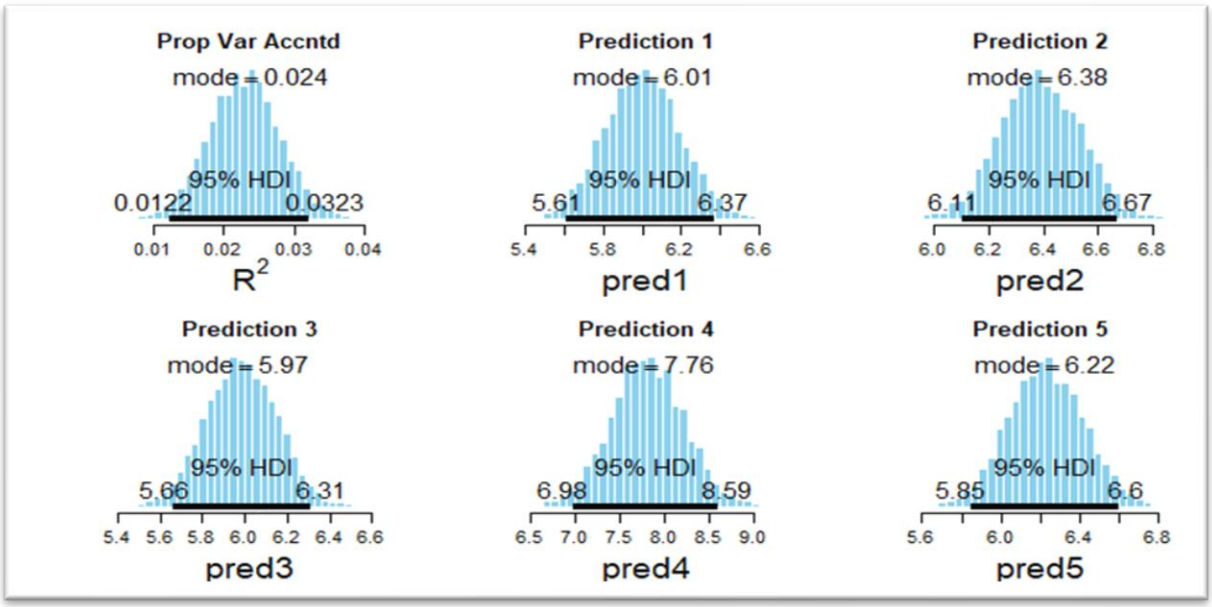


Fig 6.1.3 Posterior distribution for all the predictors with tuned MCMC settings (For Code refer to Appendix F)

From the above posterior outputs, we got some insights from Bayesian estimates and HDI intervals which are described below:

Prediction 1- For Unit Property type and Area = 600, Number of Bedrooms=2, Number of Bathrooms=2 and Number of Car Parks= 1 the Sales Price of the property will be 6,01,000 AUD.

Prediction 2- For House Property type and Area = 800, Number of Bedrooms=3, Number of Bathrooms=1 and Number of Car Parks=2 the Sales Price of the property will be 6,38,000 AUD.

Prediction 3- For House Property type and Area = 1500, Number of Bedrooms=2, Number of Bathrooms=1 and Number of Car parks=1 the Sales Price of the property will be 5,97,000 AUD.

Prediction 4- For House Property type and Area = 2500, Number of Bedrooms=5, Number of Bathrooms=4, Number of Car Parks=4 the Sales Price of the property will be 7,76,000 AUD.

Prediction 5- For Unit Property type and Area = 250, Number of Bedrooms=3, Number of Bathrooms=2 , Number of Car Parks=1 the Sales Price of the property will be 6,22,000 AUD.

After the Analysing of Bayesian estimate for different values of predictors, its time to check the R-squared posterior distribution and interpret goodness of fit. The Bayesian estimate for R- squared value is 0.024 which is really less from optimal value that is 0.085. However, we can neglect this R-squared value as it is of not much significance in Bayesian Linear Regression Analysis.

At the end, we will observe the kernel density plot of Observed vs predictive Values predicted by implemented model.

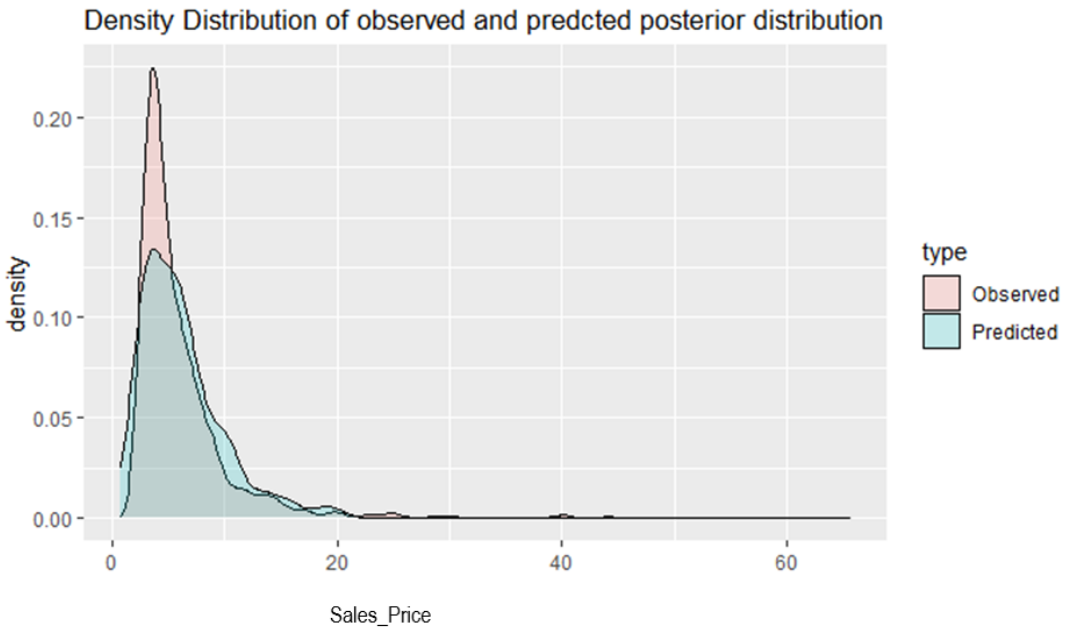


Fig 6.1.4 Density Distribution of Observed vs Predicted Posterior Distribution (For Code Refer to APPENDIX G)

From the above density distribution, we can observe that there is lesser gap between the observed and predicted Sales price values which shows the goodness of fit is yet can be improved. Also, the poor R- squared value and Kernel density curve we can observed that the goodness of fit is really poor. It can Improved through below mentioned methods:

1) Taking into account different parameters values and predictors values

2) Having Large GPU Cores to handle to handle Large Sample of Data
3) Different distributions instead of Gamma distributions such ass Exponential, Bernoulli, Poisson Distributions
4) Appropriate predictors values for the model implementations.

7) Conclusions and Recommendations

From the above Bayesian Multiple Regression model, we got the best Sale Price with appropriate independent features values in the form of this Linear Regression Equation:

Sale Price= 5.13 + 0.00000193Area+ 0.206Bedrooms+ 0.210Bathrooms + 0.179CarParks - 0.157PropertyType +ε

To improve the efficiency of predictive Analysis through Bayesian Regression model following are the future recommendations given as below:

1) Using CUDA to develop interface functions interacting with R and C to accelerate the GPU cores.
2) Importing gputools and cudaBayesreg along with clufft.r. r files in R for parallel computing
3) Importing dclone library to use jags. Parfit function to form cluster and initiates parallel computing.
4) Using high efficiency supercomputers having more GPU cores processors to increase the computation power.

8) REFERENCES

1) Dr, Haydar Demirhan, app for Gamma Distribution Specified by Mean and Standard Deviation, based on the style settings given by Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition: A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier. 3.(Accessed on 01-09-2020)
2) Dr. Haydar Demirhan, MATH2269 Module 4 notes - Markov Chain Monte Carlo - MCMC Methods, School of Science, RMIT University. (Accessed on 01-09-2020)
3) 3) Zhao. Patric .Accelerate R Applications with CUDA.https://developer.nvidia.com/blog/accelerate-r-applications-cuda/ (Accessed on 01-09-2020)
4) 4) Zhao. Patric. Innovation never sleeps. https://developer.nvidia.com/cuda-zone (Accessed on 01-09-2020)
5) https://www.rdocumentation.org/packages/dclone/versions/2.3-0/topics/jags.parfit (Accessed on 01-09-2020)
6) Dr. Haydar Demirhan, MATH2269 Module 5 Just Another Gibbs Sampler notes, School of Science, RMIT University. (Accessed on 01-09-2020)
7) Dr. Haydar Demirhan, MATH2269 Module 5 Bayesian Linear Regression notes, School of Science, RMIT University. (Accessed on 01-09-2020)
8) Dr. Haydar Demirhan, MATH2269, Jags Model Diagrams. https://github.com/rasmusab/distribution_diagrams. (Accessed on 01-09-2020)

9) APPENDIX

A) Summary Info Code

```
#===============PRELIMINARY FUNCTIONS FOR POSTERIOR
INFERENCES====================

smryMCMC_Sale_Price = function(codaSamples ,
                        compVal = NULL,
                        saveName = NULL) {
 summaryInfo = NULL
 mcmcMat = as.matrix(codaSamples, chains = TRUE)
 paramName = colnames(mcmcMat)
 for (pName in paramName) {
  if (pName %in% colnames(compVal)) {
   if (!is.na(compVal[pName])) {
    summaryInfo = rbind(summaryInfo ,
                  summarizePost(
                   paramSampleVec = mcmcMat[, pName] ,
                   compVal = as.numeric(compVal[pName])
                  ))
   }
   else {
    summaryInfo = rbind(summaryInfo , summarizePost(paramSampleVec = mcmcMat[,
pName]))
   }
  } else {
   summaryInfo = rbind(summaryInfo , summarizePost(paramSampleVec = mcmcMat[,
pName]))
  }
 }
 rownames(summaryInfo) = paramName


 if (!is.null(saveName)) {
  write.csv(summaryInfo , file = paste(saveName, "SummaryInfo.csv", sep =
                                """))
 }
 return(summaryInfo)
}
```

B) Code for Plotting Posterior Distribution

```
#========================Defining plot method for construction of posterior
distribution=========================================================

 plotMCMC_Sale_Price = function( codaSamples , data, xName="x" , yName="y" ,

          showCurve=FALSE ,  pairsPlot=FALSE , compVal = NULL,

          saveName=NULL , saveType="jpg" ) {

 #-------------------------------------------------------------------------
```

```
#====================Separating Independent and dependent variables=====================


y = data[, yName]
x = as.matrix(data[, xName])


# ======================Initializing the priors and predicion variables=================


mcmcMat = as.matrix(codaSamples, chains = TRUE)
chainLength = NROW(mcmcMat)
zbeta0 = mcmcMat[, "zbeta0"]
zbeta  = mcmcMat[, grep("^zbeta$|^zbeta\\[", colnames(mcmcMat))]
if (ncol(x) == 1) {
  zbeta = matrix(zbeta , ncol = 1)
}
zVar = mcmcMat[, "zVar"]
beta0 = mcmcMat[, "beta0"]
beta  = mcmcMat[, grep("^beta$|^beta\\[", colnames(mcmcMat))]
if (ncol(x) == 1) {
  beta = matrix(beta , ncol = 1)
}
tau = mcmcMat[, "tau"]
pred1 = mcmcMat[, "pred[1]"] # Prediction 1 for 1st Settings
pred2 = mcmcMat[, "pred[2]"] # Prediction 2 for 2nd settings
pred3 = mcmcMat[, "pred[3]"] # Prediction 3 for 3rd settings
pred4 = mcmcMat[, "pred[4]"] # Prediction 4 for 4th settings
pred5 = mcmcMat[, "pred[5]"] # Prediction 5 for 5th settings


#------------------------------------------------------------------------
```

```
# Compute R^2 for credible parameters:

YcorX = cor(y , x) # correlation of y with each x predictor

Rsq = zbeta %*% matrix(YcorX , ncol = 1)

#------------------------------------------------------------------------

# Marginal histograms:

decideopenGraph_Sale_Price = function( panelCount , saveName , finished=FALSE ,
                        nRow=2 , nCol=3 ) {
 # If finishing a set:
 if ( finished==TRUE ) {
  if ( !is.null(saveName) ) {
   saveGraph( file=paste0(saveName,ceiling((panelCount-1)/(nRow*nCol))),
              type=saveType)
  }
  panelCount = 1 # re-set panelCount
  return(panelCount)
 } else {
  # If this is first panel of a graph:
  if ( ( panelCount %% (nRow*nCol) ) == 1 ) {
   # If previous graph was open, save previous one:
   if ( panelCount>1 & !is.null(saveName) ) {
    saveGraph( file=paste0(saveName,(panelCount%/%(nRow*nCol))),
               type=saveType)
   }
   # Open new graph
   openGraph(width=nCol*7.0/3,height=nRow*2.0)
```

```
    layout( matrix( 1:(nRow*nCol) , nrow=nRow, byrow=TRUE ) )
    par( mar=c(4,4,2.5,0.5) , mgp=c(2.5,0.7,0) )
  }
  # Increment and return panel count:
  panelCount = panelCount+1
  return(panelCount)
 }
}

# Original scale:
panelCount = 1
if (!is.na(compVal["beta0"])){
 panelCount = decideopenGraph_Sale_Price( panelCount ,
saveName=paste0(saveName,"PostMarg") )
 histInfo = plotPost( beta0 , cex.lab = 1.75 , showCurve=showCurve ,
          xlab=bquote(beta[0]) , main="Intercept", compVal = as.numeric(compVal["beta0"] ))
} else {
 histInfo = plotPost( beta0 , cex.lab = 1.75 , showCurve=showCurve ,
          xlab=bquote(beta[0]) , main="Intercept")
}
for ( bIdx in 1:ncol(beta) ) {
 panelCount = decideopenGraph_Sale_Price( panelCount ,
saveName=paste0(saveName,"PostMarg") )
 if (!is.na(compVal[paste0("beta[",bIdx,"]")])) {
  histInfo = plotPost( beta[,bIdx] , cex.lab = 1.75 , showCurve=showCurve ,
           xlab=bquote(beta[.(bIdx)]) , main=xName[bIdx],
           compVal = as.numeric(compVal[paste0("beta[",bIdx,"]")]))
 } else{
  histInfo = plotPost( beta[,bIdx] , cex.lab = 1.75 , showCurve=showCurve ,
           xlab=bquote(beta[.(bIdx)]) , main=xName[bIdx])
 }
}
```

```
  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMarg")
)

  histInfo = plotPost( tau , cex.lab = 1.75 , showCurve=showCurve ,

                xlab=bquote(tau) , main=paste("Scale") )

  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMarg")
)

  histInfo = plotPost( Rsq , cex.lab = 1.75 , showCurve=showCurve ,

                xlab=bquote(R^2) , main=paste("Prop Var Accntd") )

  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMarg") )

  histInfo = plotPost( pred1 , cex.lab = 1.75 , showCurve=showCurve ,

                xlab="pred1" , main="Prediction 1" ) # Added by Demirhan

  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMarg")
)

  histInfo = plotPost( pred2 , cex.lab = 1.75 , showCurve=showCurve ,

                xlab="pred2" , main="Prediction 2" ) # Added by Demirhan

  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMarg") )

  histInfo = plotPost( pred3 , cex.lab = 1.75 , showCurve=showCurve ,

                xlab="pred3" , main="Prediction 3" ) # Added by Demirhan

  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMarg") )

  histInfo = plotPost( pred4 , cex.lab = 1.75 , showCurve=showCurve ,

                xlab="pred4" , main="Prediction 4" ) # Added by Demirhan

  panelCount = decideopenGraph_Sale_Price( panelCount , finished=TRUE ,
saveName=paste0(saveName,"PostMarg") )

  histInfo = plotPost( pred5 , cex.lab = 1.75 , showCurve=showCurve ,

                xlab="pred5" , main="Prediction 5" ) # Added by Demirhan


  # Standardized scale:

  panelCount = 1

  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMargZ") )

  histInfo = plotPost( zbeta0 , cex.lab = 1.75 , showCurve=showCurve ,
```

```
            xlab=bquote(z*beta[0]) , main="Intercept" )

 for ( bIdx in 1:ncol(beta) ) {

  panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMargZ") )

  histInfo = plotPost( zbeta[,bIdx] , cex.lab = 1.75 , showCurve=showCurve ,

             xlab=bquote(z*beta[.(bIdx)]) , main=xName[bIdx] )

 }

 panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMargZ") )

 histInfo = plotPost( zVar , cex.lab = 1.75 , showCurve=showCurve ,

            xlab=bquote(z*tau) , main=paste("Scale") )

 panelCount = decideopenGraph_Sale_Price( panelCount , saveName=paste0(saveName,"PostMargZ") )

 histInfo = plotPost( Rsq , cex.lab = 1.75 , showCurve=showCurve ,

            xlab=bquote(R^2) , main=paste("Prop Var Accntd") )

 panelCount = decideopenGraph_Sale_Price( panelCount , finished=TRUE , saveName=paste0(saveName,"PostMargZ") )


 #------------------------------------------------------------------------------

}
```

C) Code for Data Descriptive Visualisation

```
#===============PRELIMINARY FUNCTIONS FOR POSTERIOR INFERENCES====================

#-------------- Importing the Assignment2PropertySalePrices in Data_Sale_Price ---

Data_Sale_Price <-
  read.csv(
    "C:/Users/komal/OneDrive/Documents/Update RMIT Study Material/3rd Semester/Applied Baysein Statistics/Assignment2/Assignment2PropertyPrices.csv"
  )
head(Data_Sale_Price)


# Setting the seed sample
set.seed(130500)

Sale_Price_Sample = sample_n(Data_Sale_Price, 1500)
Sale_Price_Sample
```

```
# Scatter plots to examine correlation between
# dependent and Independent variables

p1 <- ggplot(Sale_Price_Sample, aes(x = Area, y = SalePrice)) +
 geom_point() +
 xlab("Area") +
 ylab("SalePrice")

p2 <-
 ggplot(
  Sale_Price_Sample,
  aes(x = Bedrooms, y = SalePrice)
 ) +
 geom_point() +
 xlab("Bedrooms") +
 ylab("SalePrice")


p3 <- ggplot(Sale_Price_Sample, aes(x = Bathrooms, y = SalePrice)) +
 geom_point() +
 xlab("Bathrooms") +
 ylab("SalePrice")


p4 <- ggplot(Sale_Price_Sample, aes(x = CarParks, y = SalePrice)) +
 geom_point() +
 xlab("CarParks") +
 ylab("SalePrice")

p5 <-
 ggplot(Sale_Price_Sample, aes(x = PropertyType, y = SalePrice)) +
 geom_point() +
 xlab("PropertyType") +
 ylab("SalePrice")

figure <- ggarrange(p1, p2, p3, p4, p5, nrow = 2, ncol = 3)
figure

# Histogram of the dependent variable
hist(Data_Sale_Price$SalePrice, main = " Histogram of the Sale Price(Dependent Variable)",
xlab = "SalePrice")

# Kernel density estimation of Dependent Variable SalePrice
plot(kde(Data_Sale_Price$SalePrice), xlab = "SalePrice") # with default settings


# Separating the Dependent and Independent variable from the sample data
```

```
y = Data_Sale_Price[, "SalePrice"]
x = as.matrix(Data_Sale_Price[, c("Area", "Bedrooms", "Bathrooms", "CarParks", "PropertyType")])

# Some more descriptives
cat("\nCORRELATION MATRIX OF PREDICTORS:\n ")
show(round(cor(x), 3))
cat("\n")

xPred = array(NA, dim = c(5, 5))
xPred[1,] = c(600, 2, 2, 1, 1)
xPred[2,] = c(800, 3, 1, 2, 0)
xPred[3,] = c(1500, 2, 1, 1, 0)
xPred[4,] = c(2500, 5, 4, 4, 0)
xPred[5,] = c(250, 3, 2, 1, 1)
```

D) Code for Model String

```
data {
   ysd <- sd(y)
   for ( i in 1:Ntotal ) {
    zy[i] <- y[i] / ysd
   }
   for ( j in 1:Nx ) {
    xsd[j] <-   sd(x[,j])
    for ( i in 1:Ntotal ) {
     zx[i,j] <- x[i,j] / xsd[j]
    }
   }
 }

 # Model Specification for scaled data:

model {
for (i in 1:Ntotal) {
zy[i] ~ dgamma((mu[i] ^ 2) / zVar , mu[i] / zVar)
   mu[i] <- zbeta0 + sum(zbeta[1:Nx] * zx[i, 1:Nx])
 }
 # Priors on standardized scale:
 zbeta0 ~ dnorm(0 , 1 / 2 ^ 2)
 zbeta[1] ~ dnorm(0.0009 / xsd[1] , 1 / (0.2 / xsd[1] ^ 2))
 zbeta[2] ~ dnorm(1 / xsd[2] , 1 / (30 / xsd[2] ^ 2))
 zbeta[3] ~ dnorm(0 , 1 / 2 ^ 2)
 zbeta[4] ~ dnorm(1.2 / xsd[4] , 1 / (2 / xsd[4] ^ 2))
 zbeta[5] ~ dnorm(-1.5 / xsd[5] , 1 / (0.2 / xsd[5] ^ 2))

 zVar ~ dgamma(0.01 , 0.01)

 # Transform to original scale:
```

```
  beta[1:Nx] <- (zbeta[1:Nx] / xsd[1:Nx]) * ysd
  beta0 <- zbeta0 * ysd
  tau <- zVar * (ysd) ^ 2

  # Compute predictions at every step of the MCMC

  for (i in 1:5) {
pred[i] <-
 beta0 + beta[1] * xPred[i, 1] + beta[2] * xPred[i, 2] + beta[3] * xPred[i, 3] + beta[4] * xPred[i,
4] + beta[5] * xPred[i, 5]
  }

}
```

E) Code for Parallel Run

```
# Parallel run of the jags model

STARTTIME <- Sys.time()
runJagsOut <- run.jags(
  method = "parallel" ,
  model = "SalePriceModel.txt" ,
  monitor = c("zbeta0" ,  "zbeta" , "beta0" ,  "beta" ,  "tau", "zVar", "pred")  ,
  data = dataList  ,
  inits = initsList,
  n.chains = nChains ,
  adapt = adaptSteps ,
  burnin = burnInSteps ,
  sample = numSavedSteps ,
  thin = thinSteps ,
  summarise = FALSE ,
  plots = FALSE
)
codaSamples = as.mcmc.list(runJagsOut)
```

F) Code for printing the diagnostics check

```
diagMCMC(codaSamples , parName = "beta0")
diagMCMC(codaSamples , parName = "beta[1]")
diagMCMC(codaSamples , parName = "beta[2]")
diagMCMC(codaSamples , parName = "beta[3]")
diagMCMC(codaSamples , parName = "beta[4]")
diagMCMC(codaSamples , parName = "beta[5]")
diagMCMC(codaSamples , parName = "tau")
diagMCMC(codaSamples , parName = "pred[1]")
diagMCMC(codaSamples , parName = "pred[2]")
diagMCMC(codaSamples , parName = "pred[3]")
diagMCMC(codaSamples , parName = "pred[4]")
diagMCMC(codaSamples , parName = "pred[5]")
```

G) Code for Predictive Check

```
# ============ Predictive check ============

coefficients <-  summaryInfo_Sale_Price[8:13, 3] # Get the model coefficients out
Variance <- summaryInfo_Sale_Price[14, 3] # Get the variance out
# Since we imposed the regression model on the mean of the gamma likelihood,
# we use the model (X*beta) to generate the mean of gamma population for each
# observed x vector.

meanGamma <- as.matrix(cbind(rep(1, nrow(x)),  x)) %*% as.vector(coefficients)

# Generate random data from the posterior distribution. Here I take the
# reparameterisation back to alpha and beta.

randomData <-
  rgamma(n = 231,
         shape = meanGamma ^ 2 / Variance,
         rate = meanGamma / Variance)


# Display the density plot of observed data and posterior distribution:

predicted <- data.frame(Sale_Price = randomData)
observed <- data.frame(Sale_Price = y)
predicted$type <- "Predicted"
observed$type <- "Observed"
dataPred <- rbind(predicted, observed)

ggplot(dataPred, aes(elapsed, fill = type)) +
  geom_density(alpha = 0.2)+
  ggtitle("Density Distribution of observed and predicted posterior distribution")
```