Assignment 2
Applied Bayesian Statistics

SEMESTER 2, 2020

Brian Steven Rathod
S3760875 | RMIT UNIVERSITY

Contents

Introduction

Prediction and analysis of property prices is an important research area in academia, attracting strong interest from the housing industry as well. The Australian housing market has long been notorious for drastic fluctuations and inflation in property prices. This report will go through the analysis performed on a fabricated dataset of house prices in Melbourne.

Three Bayesian linear regression models are fitted on the dataset. The first 3 sections contain the mathematical model used and the prior specification for the model. Prior information is taken into consideration to specify the model parameters. Descriptive and summary statistics have been provided.

Sections 4 to 6 describe the findings of each regression model (which are referred to as a 'Run'). MCMC diagnostics are performed for each 'Run' and posterior distributions interpreted in further detail. A goodness-of-fit test is performed to evaluate the suitability of the model. Recommendations are provided in the final section.

Methodology

Data Collection

Data was downloaded on 10th September 2020 from the RMIT Canvas website, at MATH2269 webpage. The dataset contained 10,000 sale price observations for properties sold in Melbourne, which was identified as the dependent variable. The scale is in 100,000s.

There were five independent variables in the dataset –

- Area: Land size in m^2 of the sold property
- Bedrooms: The number of bedrooms
- Bathrooms: The number of bathrooms
- CarParks: The number of car parks
- PropertyType: The type of the property (0: House, 1: Unit)

Data Analysis

The analysis was primarily performed using Rstudio. Furthermore, JAGS software was invoked from R scripts to run the regression models. The relevant R scripts were adapted from the textbook and were provided by Dr. Demirhan on Canvas.

Results & Discussion

1. Descriptive Statistics

Summary statistics for the dataset were calculated using Rstudio. SalePrice and Area are continuous variables and their statistics are displayed in Table 1. Property Sale prices range from $200,000 to $70,00,000. Average sale price is $600,000. Property areas range from 50 m^2 to 3500 m^2. Average area is 690 m^2.

	Min	1st Quartile	Median	Mean	3rd Quartile	Max
SalePrice.100K.	2.000	3.500	4.500	6.094	6.550	70.000
Area	50.0	353.0	568.0	690.2	752.0	3500.0

Table 1. Summary Statistics

The frequency plots for Bedrooms, Bathrooms, CarParks and PropertyType are shown in Figure 1, owing to the nominal nature of those variables. For PropertyType, 0 is 'house' and 1 is 'unit'.

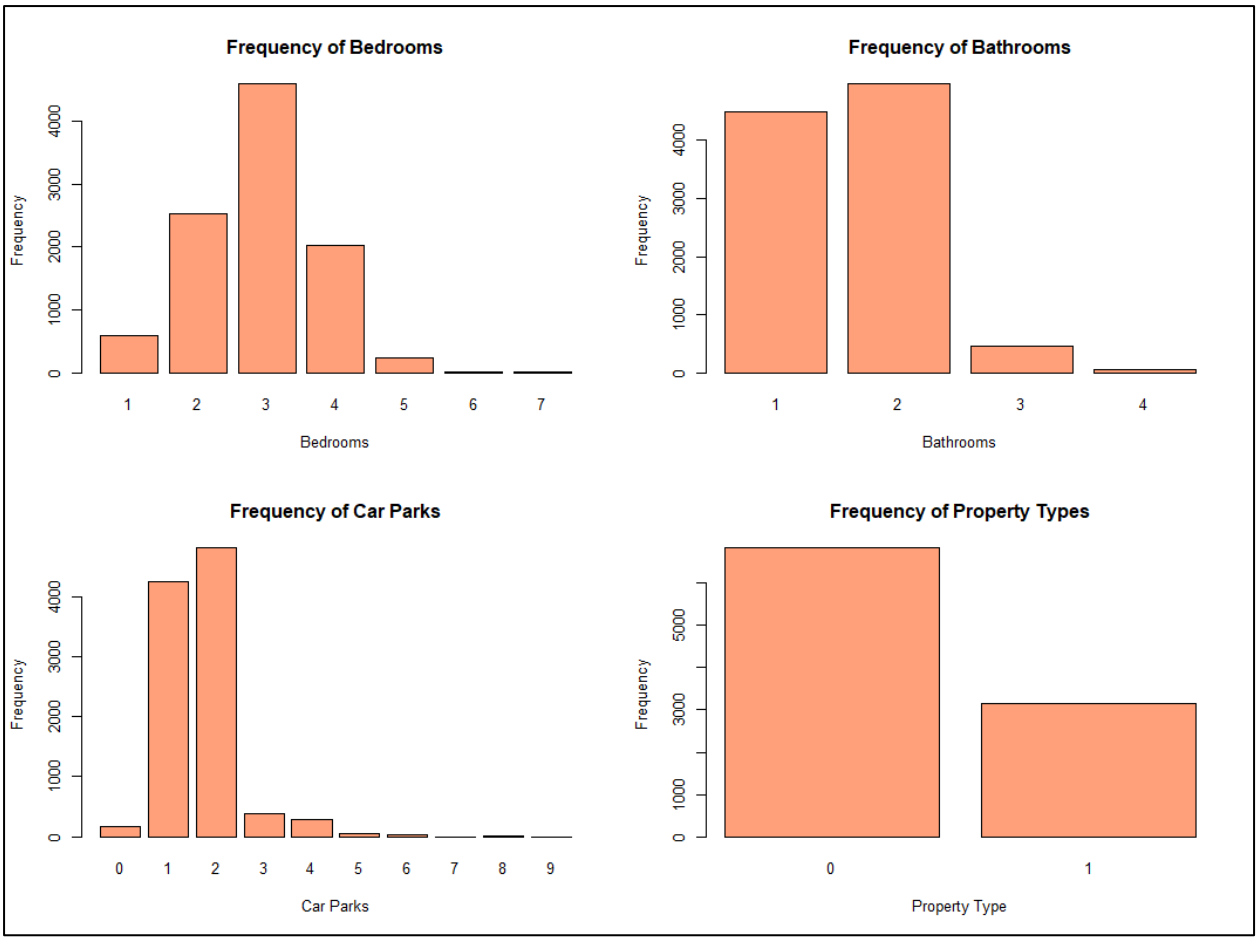

Figure 1. Frequency Plots

Scatter Plots between X and Y (SalePrice) variables

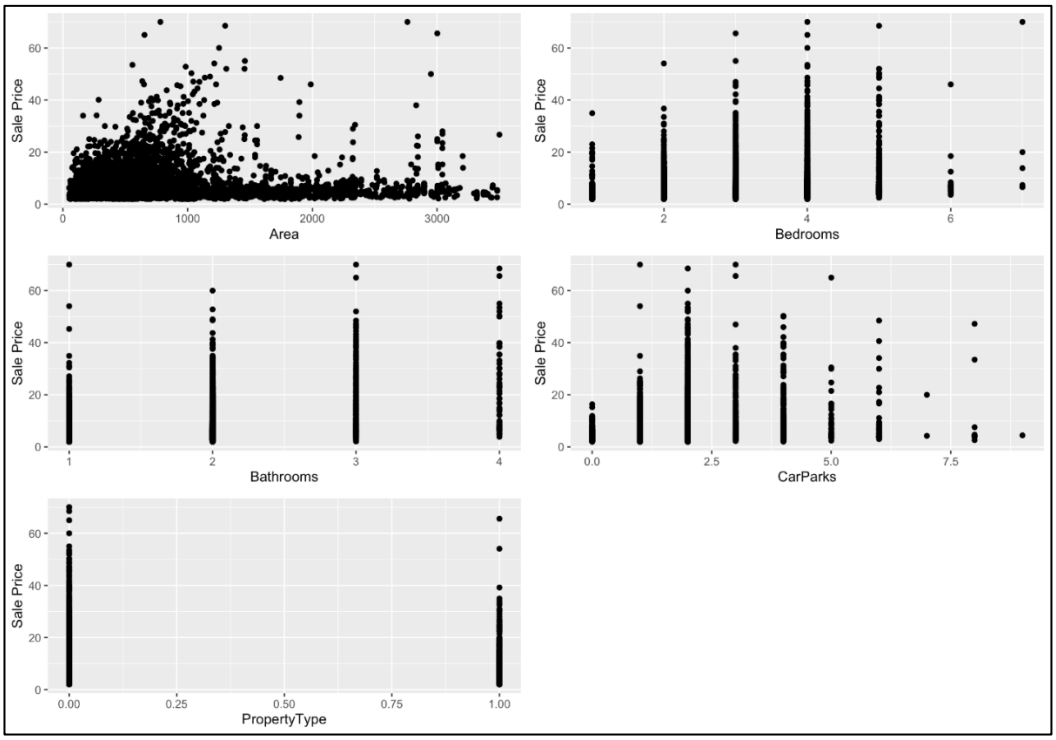

Figure 2. Scatter Plots between X and Y variables

1. Area and Sale Price: No obvious linear relationship can be observed between the two variables. Houses having an area from 0-1000 m2 have a higher representation(dots) on the graphs, compared to other intervals. Higher areas are not strictly related with higher sale prices. The graph however explains the frequency of the X variable.

2. Bedrooms and Sale Price: There is a clear nonlinear relationship between the two variables. 3, 4 and 5-bedroom properties tend to have higher sale prices compared to other no. of bedrooms.

3. Bathrooms and Sale Price: There is a clear nonlinear relationship between the two variables. The scatter plot shows the distribution of 1, 2, 3 and 4 bathrooms by their sale prices.

4. CarParks and Sale Price: There is a clear nonlinear relationship between the two variables. Properties with 3, 4 and 5 carparks tend to have slightly higher sale prices compared to other numbers of bedrooms. However, clear trends could no be observed from the plot.

5. PropertyType and Sale Price: There is a clear nonlinear relationship between the two variables. Frequency of houses and units is displayed in Fig.2. No clear trends could be observed from the plot.

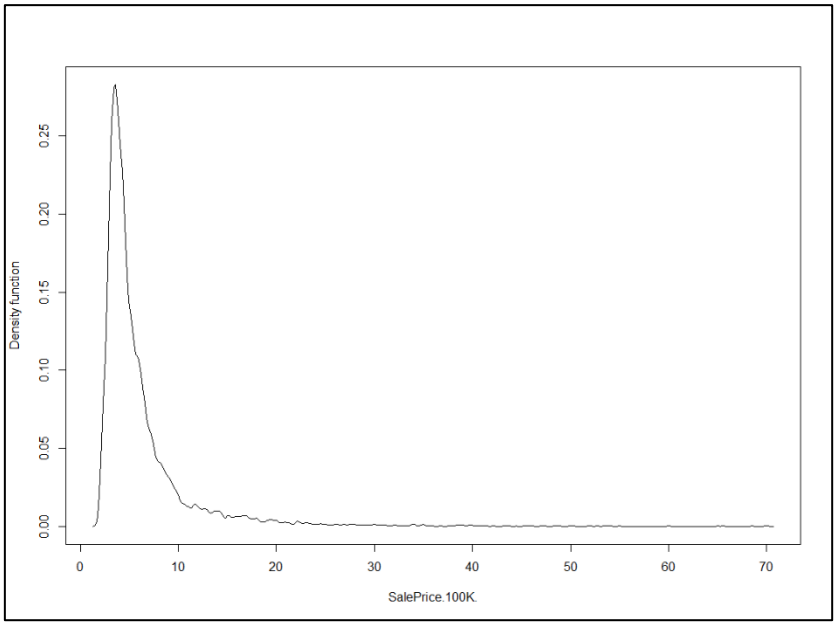

Figure 3. SalePrice Kernel Desnity Estimation

Fig.3 shows the kernel density estimate for the dependent variable. It appears to have a gamma or exponential shape, with a sharp peak at the beginning and a sudden drop and long tail.

2. Mathematical Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \epsilon,$$

$$\epsilon \sim \text{Gamma}\ (\mu^2/\sigma^2,\ \mu/\sigma^2)$$

A multiple linear regression model is fitted on the dependent Y variable, where Beta0 is the intercept and Betas 1-5 represent the slopes for the 5 predictor variables.

ε is modelled with a Gamma distribution, where α is the shape of the distribution and and β is the scale of the distribution. α, β are derived using the formulas: $\alpha = \mu_2/\sigma_2$, $\beta = \mu/\sigma_2$

3. Prior Specification

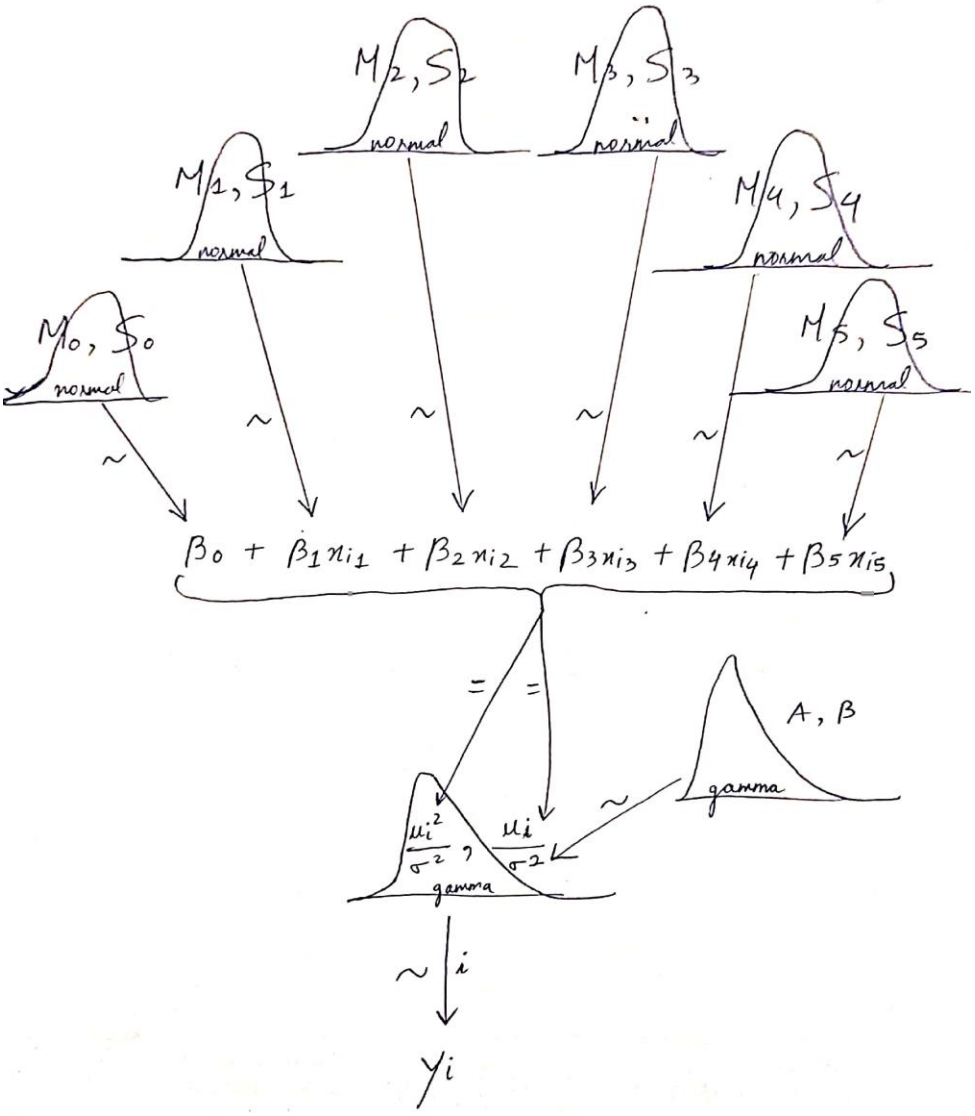

Figure 4. Prior Specification

- The independent variable y(i) has a gamma likelihood with α and β reparameterized as α=μ2/σ2, β=μ/σ2.
- A gamma prior is specified on the σ of the likelihood, as seen in Fig.4.
- A multiple linear regression model is specified on the μ of the likelihood. It has an intercept and Beta values 1 to 5.
- For each Beta parameter in the model, a normal prior is specified with its own mean and variance parameters.
- Specific parameter values for the models are listed in Table.2 below.

Three variations of the Normal-Gamma JAGS regression models were implemented on the data. The key model parameters and differences between the 3 runs have been summarised in Table.2 below. The first section lists the MCMC settings for each run, followed by the prior mean and variance settings.

To summarise, Run 1 was the initial run with a low number of iterations. It had just 3 thinning steps to test the model functionality. Run 2 was the key run for this analysis with the best results obtained. It was an attempt at MCMC optimisation and accurate posterior estimation. Similary, Run 3 was an attempt to test sensitivity of results to changes in MCMC and prior settings.

Note: Standardisation was performed on the data to catalyse MCMC optimisation and efficiency. Prior values were specified on the standardised beta (zbeta) parameters. The zbetas were converted back to their beta parameters during the computation process to calculate the Bayesian estimates.

	Run 1	Run 2	Run 3
MCMC Parameters:			
Adapt Steps	1500	1500	1500
Burn in Steps	5000	5000	5000
No. of Chains	3	3	3
Thinning Steps	3	14	20
Saved steps	4000	6500	4800
No. of iterations	4000	30334	32000
Initial values	Not used	Used	Used
Prior variances			
zbeta[1]	0.1	0.1	0.01
zbeta[2]	4	4	7
zbeta[3]	4	4	4
zbeta[4]	1	1	1
zbeta[5]	0.1	0.1	0.01
Prior means			
zbeta[1]	0.00090	0.00090	0.00090
zbeta[2]	1	1	1
zbeta[3]	0	0	0
zbeta[4]	1.20	1.20	1.20
zbeta[5]	-1.50	-1.50	-1.50

Table 2. Model Settings by each Run

Table 2 highlights the key differences in models. Runs 1, 2 and 3 differ in thinning steps, saved steps and the number of iterations. Thinning steps are increased for each run. Run 3 also has differences in prior variances compared to Runs 1 & 2.

4. RUN 1

4.1 MCMC Diagnostics

The relevant MCMC diagnostic plots for Run 1 have been included in part A of the Appendix of this report. A concise summary of the parameter diagnostics is provided in Table 3, including Betas 1 to 5, Tau and five prediction estimates. Suitable colour themes are used for each category represented.

Parameter	Trace Plot	Density Plot	Shrink factor	ACF	ESS	MCSE
Beta 0	Overlapping can be improved	Overlapping chains, cover HDIs, tails	<1.2, close to 1.00	Significant autocorrelations at higher lags	367.3, too low	Close to 0
Beta 1	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	10807.6, good	Close to 0
Beta 2	Overlapping can be improved	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Significant autocorrelations at higher lags	450, too low	Close to 0
Beta 3	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails	<1.2, close to 1.00	Significant autocorrelations at higher lags	688, Low	Close to 0
Beta 4	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Significant autocorrelations at higher lags	1426.2, Low	Close to 0
Beta 5	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Significant autocorrelations at higher lags	726.3, Low	Close to 0
Tau	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	6565.7, Decent	Close to 0
Pred 1	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Significant autocorrelations at higher lags	1598.1, Low	Close to 0
Pred 2	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Significant autocorrelations at higher lags	1186, Low	Close to 0
Pred 3	Overlapping can be	Overlapping chains, cover	<1.2, close to 1.00	Significant autocorrelations	496.3, Too low	Close to 0

	improved	HDIs, tails and peak		at higher lags		
Pred 4	Overlapping can be improved	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Significant autocorrelations at higher lags	688, Low	Close to 0
Pred 5	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Significant autocorrelations at higher lags	1170.5, Low	Close to 0

Table 3. MCMC Diagnostics - Run 1

Trace plot, density plot and shrink factor reflect the representativeness of the MCMC chains. Overlapping and mixing of chains is decent for most parameters, however, it can be improved for a few other parameters as seen in Table 3. Shrink factor is close to 1.00 which is a good sign. Overall, the representativeness is good but with room for improvement.

ACF, ESS and MCSE are measures of MCMC accuracy. Significant autocorrelation was observed for most parameters at higher lags, which was something that needed to be factored into future runs. Effective Sample Size (ESS) was quite low for the parameters and needs to be improved upon. Monte Carlo Standard Error (MCSE) was close to 0 for all parameters, which is a very good result. To reiterate, the accuracy of chains can be improved.

The time elapsed to run the chains was approximately 8 hours which was poor for a lower combination of iterations. A possible explanation for this could be the nonexecution of initial values for the model. The efficiency therefore has large room for improvement.

4.2 Posterior Distributions

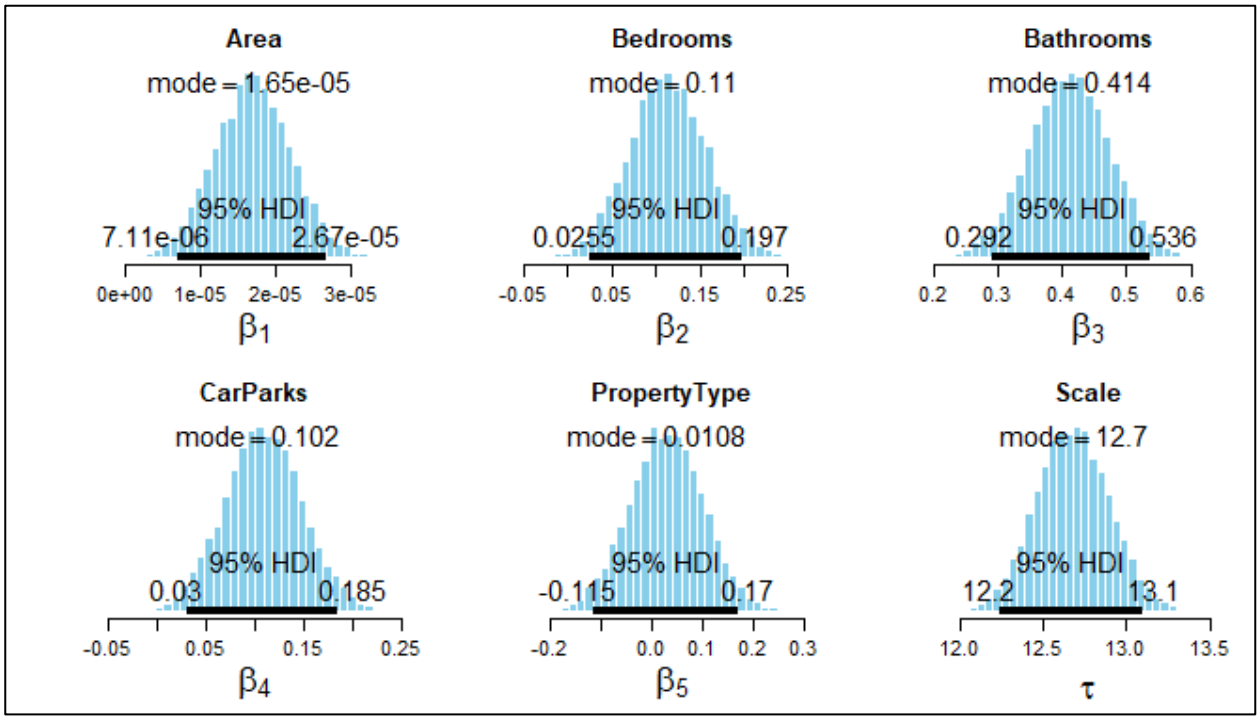

Figure 5. Posterior Distributions - Run 1

- Beta 1 has a posterior mode of 1.65e-05 and the 95% HDI do not include 0, indicating it has a significant impact on the dependent variable.

- Beta 2 has a posterior mode of 0.11 and a 95% HDI (0.02, 0.19), indicating it has a statistically significant impact on the dependent variable.

- Beta 3 has a posterior mode of 0.414 and a 95% HDI (0.29, 0.53), indicating it has a statistically significant impact on the dependent variable.

- Beta 4 has a posterior mode of 0.102 and a 95% HDI (0.03, 0.185), indicating it has a statistically significant impact on the dependent variable.

- Beta 5 has a posterior mode of 0.01 and a 95% HDI covering 0, indicating it might not have a significant impact on the dependent variable.

- Tau has a posterior mode of 12.7 and a 95% HDI (12.2, 13.1), indicating the probable variance for SalePrice.

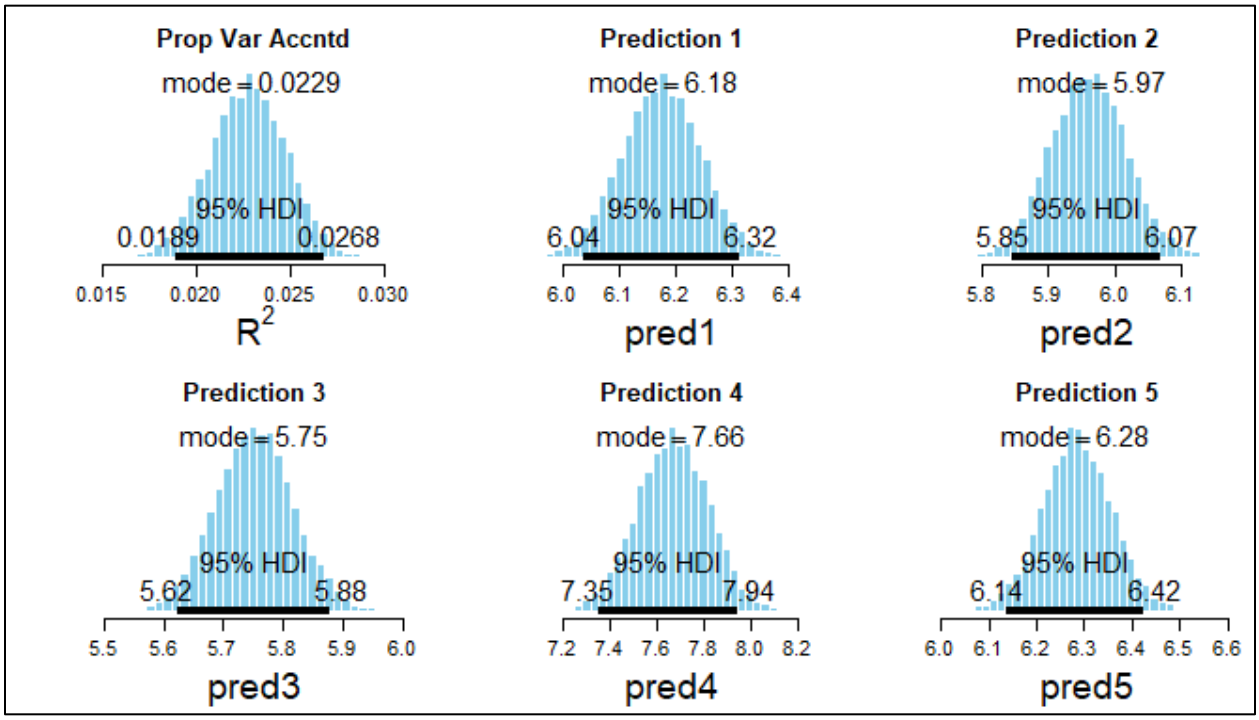

Figure 6. Posterior Distributions - Run 1

R^2 has a posterior mode of 0.02 and a 95% HDI (0.01, 0.02), which translates to roughly 2% of the variance explained by the independent variables. It is very low for a regression model, however, possible reasons for that include:

a) categorical nature of 4 predictor variables out of total 5.
b) high number of observations in the data (Bock, T. 2020)

- Prediction 1 has a posterior mode (predicted price) of 6.18 and a 95% HDI (6.04, 6.32), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 2 has a posterior mode (predicted price) of 5.97 and a 95% HDI (5.85, 6.07), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 3 has a posterior mode (predicted price) of 5.75 and a 95% HDI (5.62, 5.88), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 4 has a posterior mode (predicted price) of 7.66 and a 95% HDI (7.35, 7.94), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 5 has a posterior mode (predicted price) of 6.28 and a 95% HDI (6.14, 6.42), indicating the range of the possible values. The result is statistically valid and significant.

	Mean	Median	Mode	ESS	HDImass	HDIlow	HDIhigh
CHAIN	2.000000e+00	2.000000e+00	9.979797e-01	1.5	0.95	1.00000e+00	3.00000e+00
zbeta0	9.719959e-01	9.718030e-01	9.723675e-01	395.5	0.95	9.20490e-01	1.02639e+00
zbeta[1]	1.864135e-03	1.863125e-03	1.823311e-03	10400.9	0.95	7.83580e-04	2.94666e-03
zbeta[2]	1.987610e-02	1.978190e-02	1.924072e-02	412.8	0.95	4.46471e-03	3.45652e-02
zbeta[3]	4.870011e-02	4.872905e-02	4.896597e-02	671.8	0.95	3.45403e-02	6.34861e-02
zbeta[4]	1.741324e-02	1.737525e-02	1.654621e-02	1424.2	0.95	4.85293e-03	2.98589e-02
zbeta[5]	2.637466e-03	2.610810e-03	9.799504e-04	691.0	0.95	-1.04466e-02	1.54682e-02
beta0	4.979767e+00	4.978780e+00	4.981682e+00	395.5	0.95	4.71589e+00	5.25846e+00
beta[1]	1.690695e-05	1.689780e-05	1.653672e-05	10400.9	0.95	7.10676e-06	2.67251e-05
beta[2]	1.135491e-01	1.130105e-01	1.099193e-01	412.8	0.95	2.55062e-02	1.97466e-01
beta[3]	4.112926e-01	4.115370e-01	4.135380e-01	671.8	0.95	2.91707e-01	5.36166e-01
beta[4]	1.076686e-01	1.074335e-01	1.023076e-01	1424.2	0.95	3.00063e-02	1.84622e-01
beta[5]	2.905790e-02	2.876425e-02	1.079654e-02	691.0	0.95	-1.15093e-01	1.70419e-01
tau	1.267518e+01	1.267390e+01	1.268907e+01	6961.9	0.95	1.22347e+01	1.30956e+01
zVar	4.829088e-01	4.828600e-01	4.834377e-01	6961.9	0.95	4.66126e-01	4.98925e-01
pred[1]	6.176321e+00	6.176835e+00	6.178870e+00	1559.7	0.95	6.03869e+00	6.31647e+00
pred[2]	5.960569e+00	5.960590e+00	5.965886e+00	1231.9	0.95	5.84671e+00	6.07002e+00
pred[3]	5.751186e+00	5.751125e+00	5.751502e+00	486.9	0.95	5.62246e+00	5.87780e+00
pred[4]	7.665624e+00	7.666560e+00	7.661017e+00	701.9	0.95	7.34762e+00	7.94324e+00
pred[5]	6.283952e+00	6.283240e+00	6.279574e+00	1068.4	0.95	6.13829e+00	6.42500e+00

Figure 7. Summary Stats - Run 1

Fig.7 displays the posterior mean, median, mode, ESS and HDI estimates for beta 0-5, tau and predictions 1-5. ESS represents the Effective Sample Size out of total number of iterations. HDImass is the probability density interval of 95%, with adjoining HDI high and low values.

5. RUN2

5.1 MCMC Diagnostics

Run 2 was the most significant model for the purpose of this purpose, owing to the larger thinning steps and number of iterations involved. Prior means and variances are the same as Run 1 but a key difference is the inclusion of (arbitrary) initial values for the chains. The computation time was nearly 14 hours in total. The relevant MCMC plots for each parameter is given in Figures 8 & 9 for Run 2.

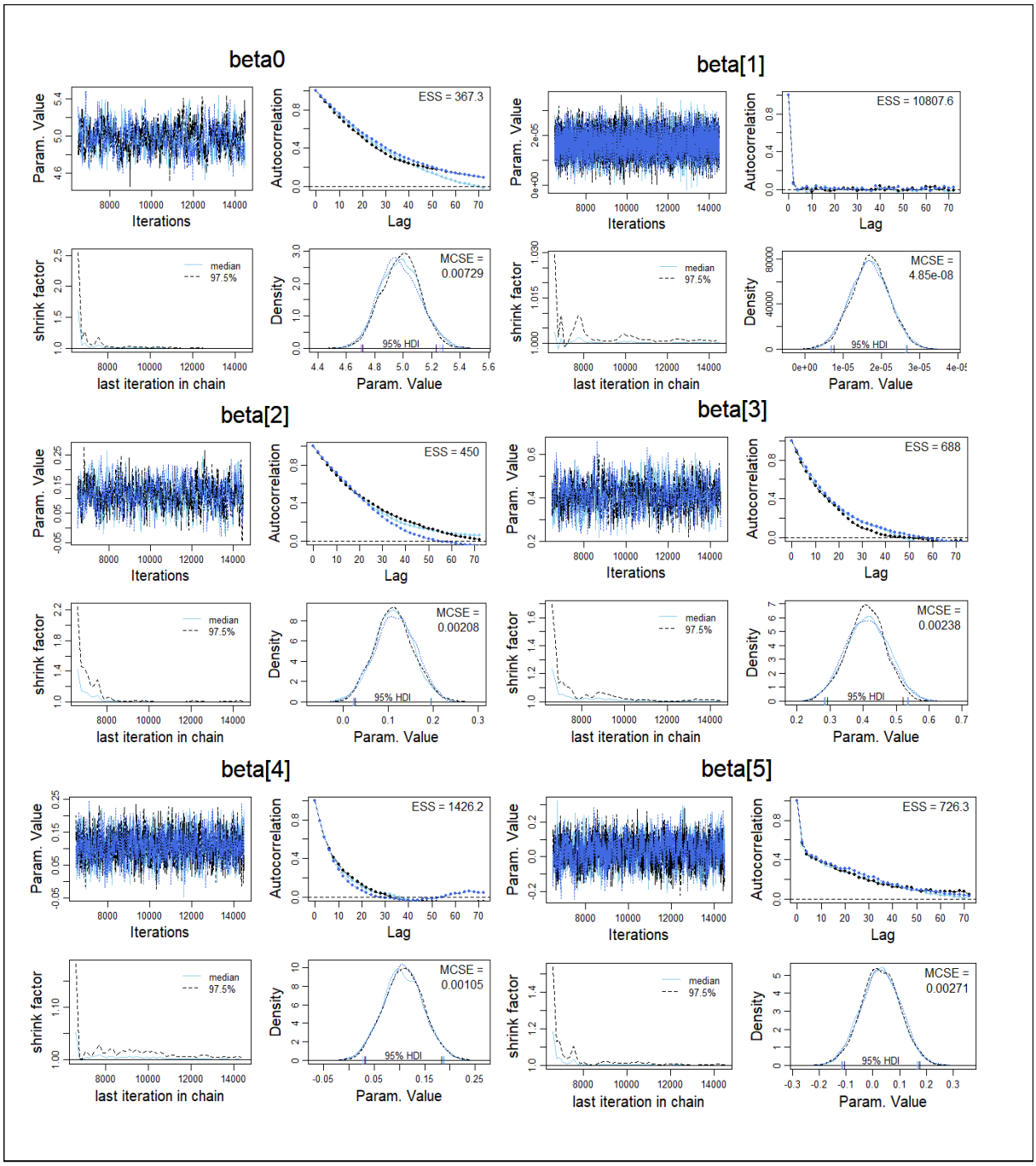

Figure 8. MCMC Diagnostics - Run 2

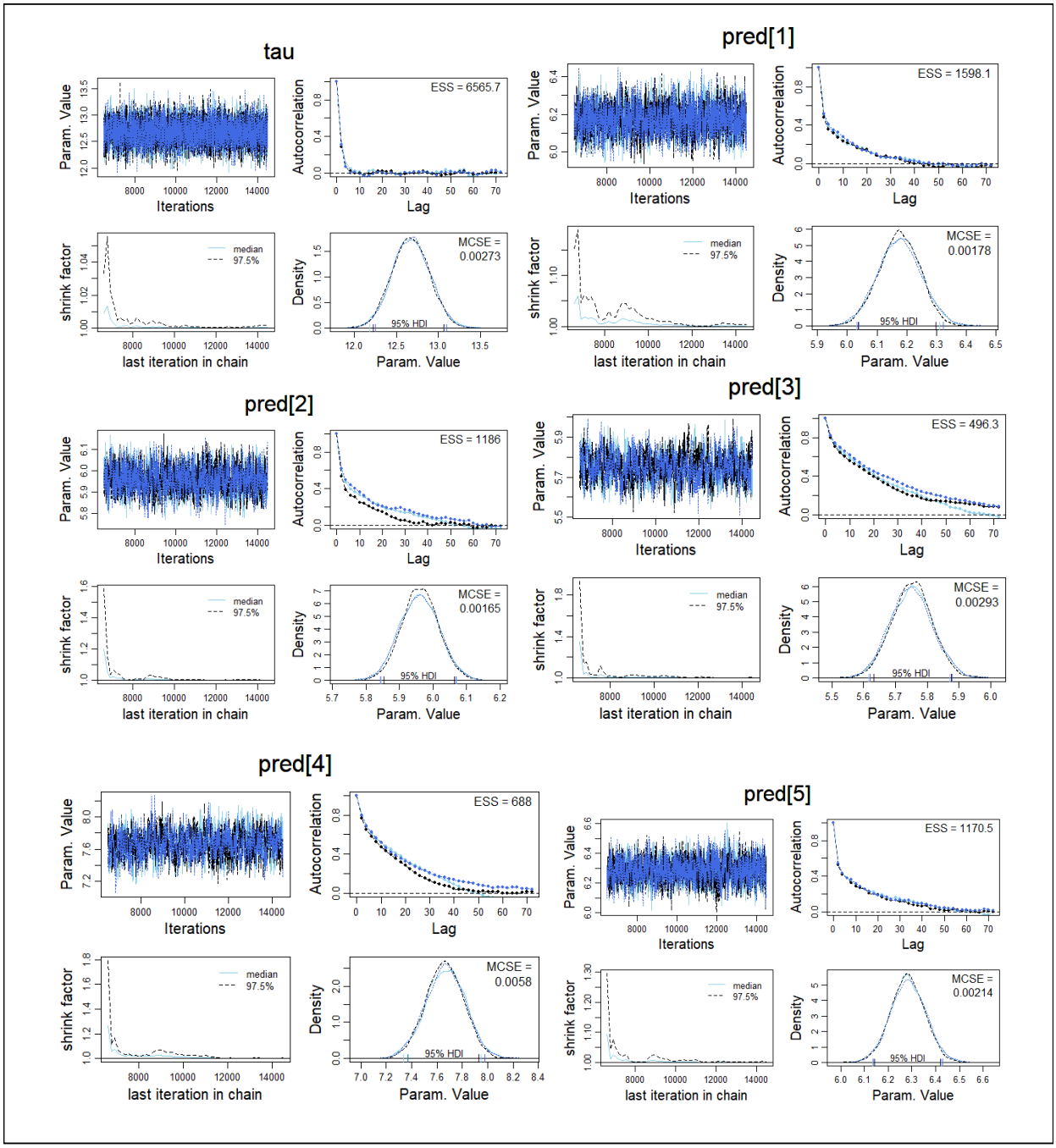

Figure 9. MCMC Diagnostics - Run 2

A summary of the MCMC plots is given in Table 4 below.

Parameter	Trace Plot	Density Plot	Shrink factor	ACF	ESS	MCSE
Beta 0	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Minor autocorrelations at lower lags	4305.8, Low	Close to 0
Beta 1	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	19252.3, Good	Close to 0
Beta 2	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Minor autocorrelations at lower lags	4343.8, Low	Close to 0
Beta 3	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	7636, Decent	Close to 0
Beta 4	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	13112.9, Good	Close to 0
Beta 5	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	Minor autocorrelations at lower lags	6526.9, Low	Close to 0
Tau	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	19500, Good	Close to 0
Pred 1	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	11808.9, Good	Close to 0
Pred 2	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	10198.2, Good	Close to 0
Pred 3	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	5214.7, Low	Close to 0
Pred 4	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	7385.4, Decent	Close to 0

Pred 5	Overlapping well, no failure of convergence	Overlapping chains, cover HDIs, tails and peak	<1.2, close to 1.00	No autocorrelation	9192.5, Decent	Close to 0

Table 4. MCMC Summary - Run 2

Trace plot, density plot and shrink factor reflect the representativeness of the MCMC chains. Overlapping and mixing of chains is significantly improved for most parameters with the inclusion of more thinning steps. Shrink factor is close to 1.00 which is a very good result. Overall, the representativeness is very good for the 3 chains.

ACF, ESS and MCSE are measures of MCMC accuracy. Increased thinning steps reduced the observed autocorrelation in the data as seen in Table 4. ESS saw a dramatic improvement with decent to good scores for most parameters. MCSE was close to 0 for all parameters, which is a very good result. The accuracy of the chains has markedly improved from Run 1 and is acceptable for the regression model.

The time elapsed to run the chains was approximately 14 hours which was quite high compared to Run 1. The factors which increased the run time included the number of iterations, thinning steps and prior settings. However, the usage of initial steps for the model seemed to offset the increase in time used by MCMC parameters.

5.2 Posterior Distributions

The posterior distributions for Run 2 are very similar to those obtained for Run 1, with insignificant differences. Distributions for beta parameters and predictions are displayed in Figure 10 and 11 respectively.

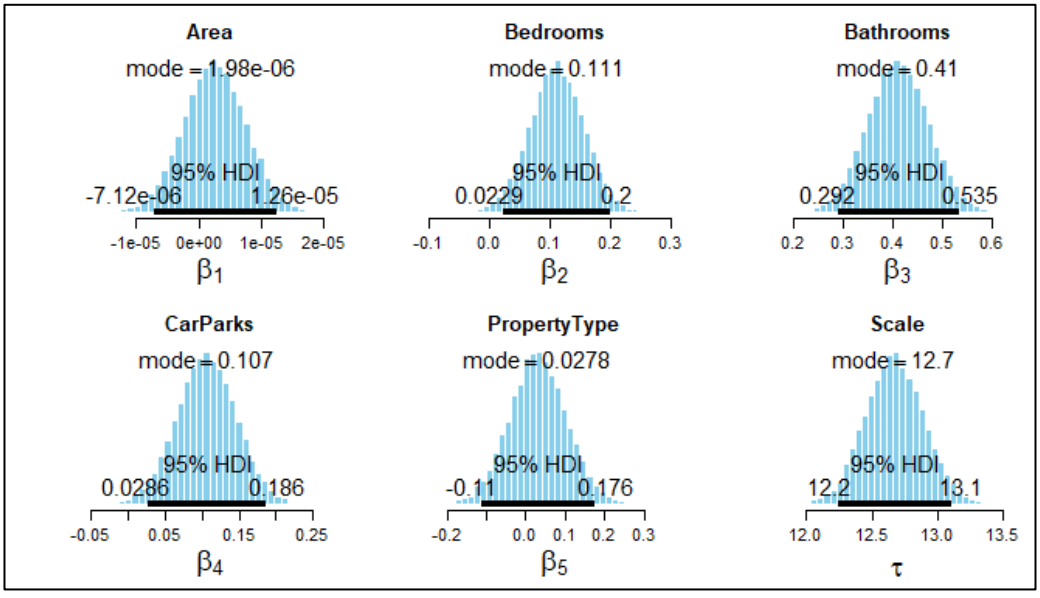

Figure 10. Posterior Distributions - Run 2

- Beta 1 has a posterior mode of 1.98e-06 and a 95% HDI covering 0, indicating it might not have a significant impact on the dependent variable.

- Beta 2 has a posterior mode of 0.11 and a 95% HDI (0.02, 0.2), indicating it has a statistically significant impact on the dependent variable.

- Beta 3 has a posterior mode of 0.41 and a 95% HDI (0.29, 0.53), indicating it has a statistically significant impact on the dependent variable.

- Beta 4 has a posterior mode of 0.107 and a 95% HDI (0.03, 0.186), indicating it has a statistically significant impact on the dependent variable.

- Beta 5 has a posterior mode of 0.02 and a 95% HDI covering 0, indicating it might not have a significant impact on the dependent variable.

- Tau has a posterior mode of 12.7 and a 95% HDI (12.2, 13.1), indicating it has a statistically significant impact on the dependent variable.

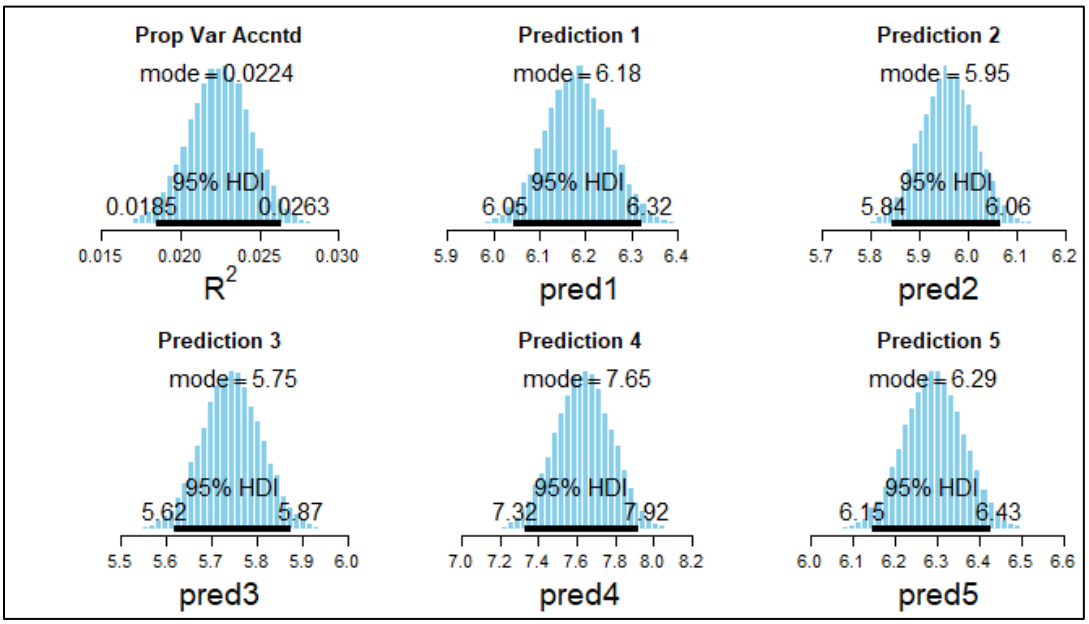

Figure 11. Posterior Distributions - Run 1

R^2 has a posterior mode of 0.02 and a 95% HDI (0.01, 0.02), which translates to roughly 2% of the variance explained by the independent variables. It is very low for a regression model, however, possible reasons for that include:

a) categorical nature of 4 predictor variables out of total 5.
b) high number of observations in the data (Bock, T. 2020)

- Prediction 1 has a posterior mode (predicted price) of 6.18 and a 95% HDI (6.05, 6.32), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 2 has a posterior mode (predicted price) of 5.95 and a 95% HDI (5.84, 6.06), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 3 has a posterior mode (predicted price) of 5.75 and a 95% HDI (5.62, 5.87), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 4 has a posterior mode (predicted price) of 7.65 and a 95% HDI (7.32, 7.92), indicating the range of the possible values. The result is statistically valid and significant.

- Prediction 5 has a posterior mode (predicted price) of 6.29 and a 95% HDI (6.15, 6.43), indicating the range of the possible values. The result is statistically valid and significant.

	Mean	Median	Mode	ESS	HDImass	HDIlow	HDIhigh
CHAIN	2.000000e+00	2.000000e+00	9.979797e-01	1.5	0.95	1.00000e+00	3.00000e+00
zbeta0	9.719959e-01	9.718030e-01	9.723675e-01	395.5	0.95	9.20490e-01	1.02639e+00
zbeta[1]	1.864135e-03	1.863125e-03	1.823311e-03	10400.9	0.95	7.83580e-04	2.94666e-03
zbeta[2]	1.987610e-02	1.978190e-02	1.924072e-02	412.8	0.95	4.46471e-03	3.45652e-02
zbeta[3]	4.870011e-02	4.872905e-02	4.896597e-02	671.8	0.95	3.45403e-02	6.34861e-02
zbeta[4]	1.741324e-02	1.737525e-02	1.654621e-02	1424.2	0.95	4.85293e-03	2.98589e-02
zbeta[5]	2.637466e-03	2.610810e-03	9.799504e-04	691.0	0.95	-1.04466e-02	1.54682e-02
beta0	4.979767e+00	4.978780e+00	4.981682e+00	395.5	0.95	4.71589e+00	5.25846e+00
beta[1]	1.690695e-05	1.689780e-05	1.653672e-05	10400.9	0.95	7.10676e-06	2.67251e-05
beta[2]	1.135491e-01	1.130105e-01	1.099193e-01	412.8	0.95	2.55062e-02	1.97466e-01
beta[3]	4.112926e-01	4.115370e-01	4.135380e-01	671.8	0.95	2.91707e-01	5.36166e-01
beta[4]	1.076686e-01	1.074335e-01	1.023076e-01	1424.2	0.95	3.00063e-02	1.84622e-01
beta[5]	2.905790e-02	2.876425e-02	1.079654e-02	691.0	0.95	-1.15093e-01	1.70419e-01
tau	1.267518e+01	1.267390e+01	1.268907e+01	6961.9	0.95	1.22347e+01	1.30956e+01
zVar	4.829088e-01	4.828600e-01	4.834377e-01	6961.9	0.95	4.66126e-01	4.98925e-01
pred[1]	6.176321e+00	6.176835e+00	6.178870e+00	1559.7	0.95	6.03869e+00	6.31647e+00
pred[2]	5.960569e+00	5.960590e+00	5.965886e+00	1231.9	0.95	5.84671e+00	6.07002e+00
pred[3]	5.751186e+00	5.751125e+00	5.751502e+00	486.9	0.95	5.62246e+00	5.87780e+00
pred[4]	7.665624e+00	7.666560e+00	7.661017e+00	701.9	0.95	7.34762e+00	7.94324e+00
pred[5]	6.283952e+00	6.283240e+00	6.279574e+00	1068.4	0.95	6.13829e+00	6.42500e+00

Figure 12. Summary Statistics- Run 2

Figure 12 displays the posterior mean, median, mode, ESS and HDI estimates for beta 0-5, tau and predictions 1-5. ESS represents the Effective Sample Size out of total number of iterations. HDImass is the probability density interval of 95%, with adjoining HDI high and low values.

6. RUN 3

The third run had the highest thinning steps at 20 and the highest number of iterations at 32000. Arbitrary initial values were specified for the model. The prior variance was set extremely low for Betas 1 and 5 at 0.01 and high for Beta 2 to reflect the degrees of belief. The primary motivation for this run was to:

a) Improve the MCMC accuracy and efficiency
b) Test sensitivity of results to changes in MCMC settings
c) Test sensitivity of results to changes in prior information.

However, changes in prior variances and MCMC settings drastically deteriorated the MCMC diagnostics as seen in Fig.13. The diagnostic plots and error estimates were very poor for the specified settings and the computer struggled to computer posterior distributions for the parameters upon multiple tries. This result was a bit surprising, considering the computation time required for the model was about 14.3 hours. The specified prior variances could possibly be massively interrupting the computation process within JAGS. A clear takeaway from this instance was that the results were very sensitive to settings of MCMC and priors, which should be taken into consideration for future modelling.

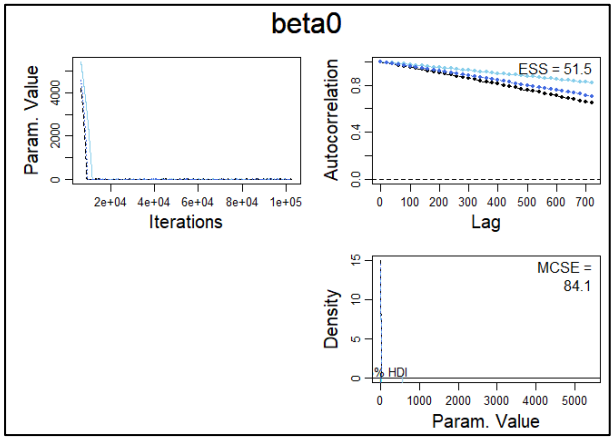

Figure 13. MCMC - Run 3

	Lower95	Median	Upper95	Mean	SD
zbeta0	0.91237	0.98697	1.0555	21.768	117.86
zbeta[1]	-0.00032988	0.000028139	0.00036959	0.000028241	0.0001792
zbeta[2]	-0.0025321	0.017124	0.037821	-1.5497	8.7346
zbeta[3]	0.030175	0.048227	0.06943	-1.0094	6.2154
zbeta[4]	0.00037515	0.016207	0.033656	-0.59831	3.7072
zbeta[5]	-0.017375	-0.000833	0.01665	-0.18404	0.92262
beta0	4.6743	5.0565	5.4076	111.52	603.81
beta[1]	-2.9919e-06	2.5521e-07	3.3521e-06	2.5614e-07	1.6252e-06
beta[2]	-0.014466	0.097827	0.21606	-8.8534	49.9
beta[3]	0.25484	0.40729	0.58637	-8.5249	52.492
beta[4]	0.0023196	0.10021	0.2081	-3.6994	22.922
beta[5]	-0.19143	-0.0091775	0.18344	-2.0276	10.165
tau	12.107	12.683	13.237	454173	2791218
zVar	0.46125	0.48319	0.50431	17303	106342
pred[1]	5.9897	6.1719	6.3544	71.037	367.51
pred[2]	5.8172	5.968	6.1112	69.037	356.47
pred[3]	5.5883	5.766	5.9311	81.59	429.04
pred[4]	7.2011	7.6103	7.998	18.357	56.26
pred[5]	6.0791	6.2711	6.4497	62.184	317.67

Total time taken: 14.3 hours

Figure 14. Summary Statistics - Run 3

Owing to the aforementioned reasons, the model is not appropriate for the analysis. Therefore, detailed MCMC diagnostics for Run 3 will not be discussed in detail.

Predictive Check

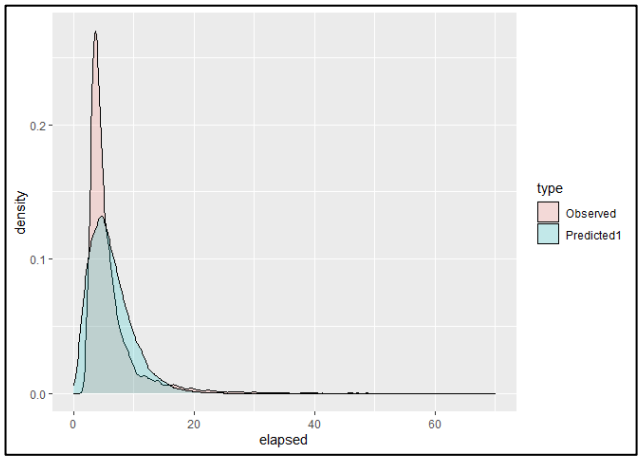

Figure 15. Density Plot - Run 1

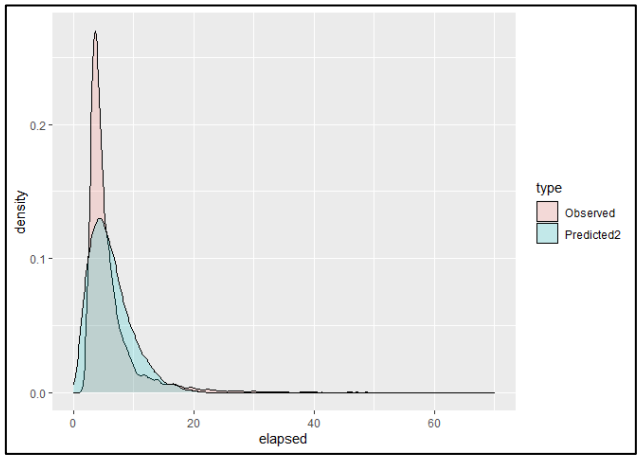

Figure 16. Density Plot - Run 2

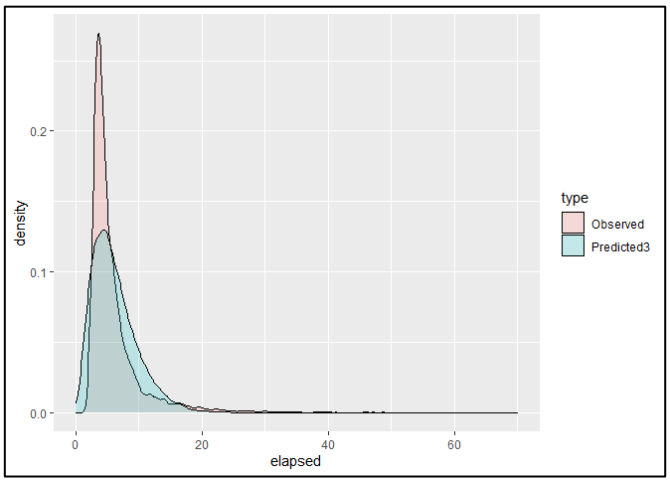

Figure 17. Density Plot - Run 3

The density plots show random data generated from the obtained coefficients of each model. This aids in the identification of goodness of fit for a model. As observed in Figures 15-17, the distibution is quite similar for the 3 models. The specified gamma likelihood covers the shape and tail of the observed data well. However, the peaks of the observed data are not covered by any of the 3 fitted models, which suggests the need for testing and implementation of other likelihoods on the data.

Conclusion

In the fourth section, an initial regression model was fitted on the data. The model had lower thinning steps and fewer iterations. The resultant MCMC diagnostics were poor owing to the high autocorrelations observed in the data. In the fifth section, a second model was fitted with significantly higher thinning steps and iterations to account for the autocorrelations. This drastically improved the MCMC diagnostics; accuracy, representativeness and efficiency were markedly better as a result.

In the sixth section, a third model was fitted to test the sensitivity of results to changes in MCMC settings and prior parameters. Increased thinning steps and very low variance produced very poor MCMC diagnostics and severely decreased the overall efficiency of computation.

Runs 1 and 2 produced very similar predictions for the specified properties [1 to 5]. All three runs had a similar prediction curve.

Property	Run 1 Prediction	Run 2 Prediction
1	6.18	6.18
2	5.97	5.95
3	5.75	5.75
4	7.66	7.65
5	6.28	6.29

Table 5. Predicted Prices

A major limitation lies in the fact that the categorical nature of independent variables was not accounted for in the regression model, which could have provided different results. A polynomial regression could be more suitable for the type of data involved. In the future, categories for each nominal feature could be reencoded as binary before modelling. And different likelihoods could be explored for the data.

References

1. Bock, T. 2020, '8 Tips for Interpreting R-Squared', Displayr Blog, viewed 17 September 2020, < https://www.displayr.com/8-tips-for-interpreting-r-squared/>

2. Demirhan, H. 2020, 'MATH2269_Module3_OnlineNotes' PDF, MATH2269, RMIT University, Melbourne.

3. Demirhan, H. 2020, 'MATH2269_Module4_OnlineNotes' PDF, MATH2269, RMIT University, Melbourne.

4. Demirhan, H. 2020, 'MATH2269_Module5_OnlineNotes' PDF, MATH2269, RMIT University, Melbourne.

5. Demirhan, H. 2020, 'MATH2269_Module6_OnlineNotes' PDF, MATH2269, RMIT University, Melbourne.

6. Demirhan, H. 2020, 'AircraftMultiple_V3.R' Rscript, MATH2269, RMIT University, Melbourne.

7. Demirhan, H. 2020, 'Assignment2PropertyPrices.csv' dataset, MATH2269, RMIT University, Melbourne.

Appendix

(A) Output

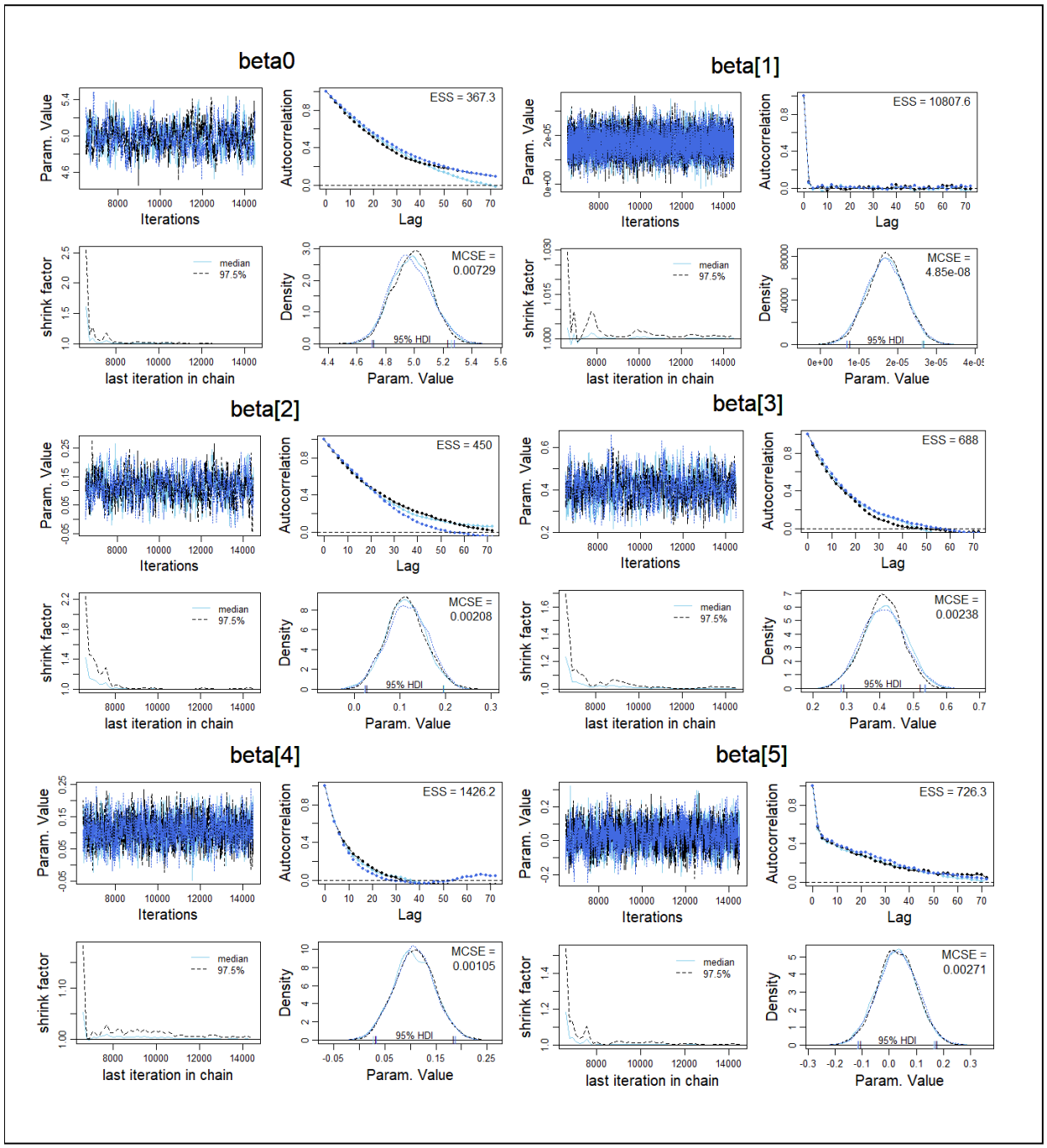

Figure 18. MCMC Diagnostics - Run 1

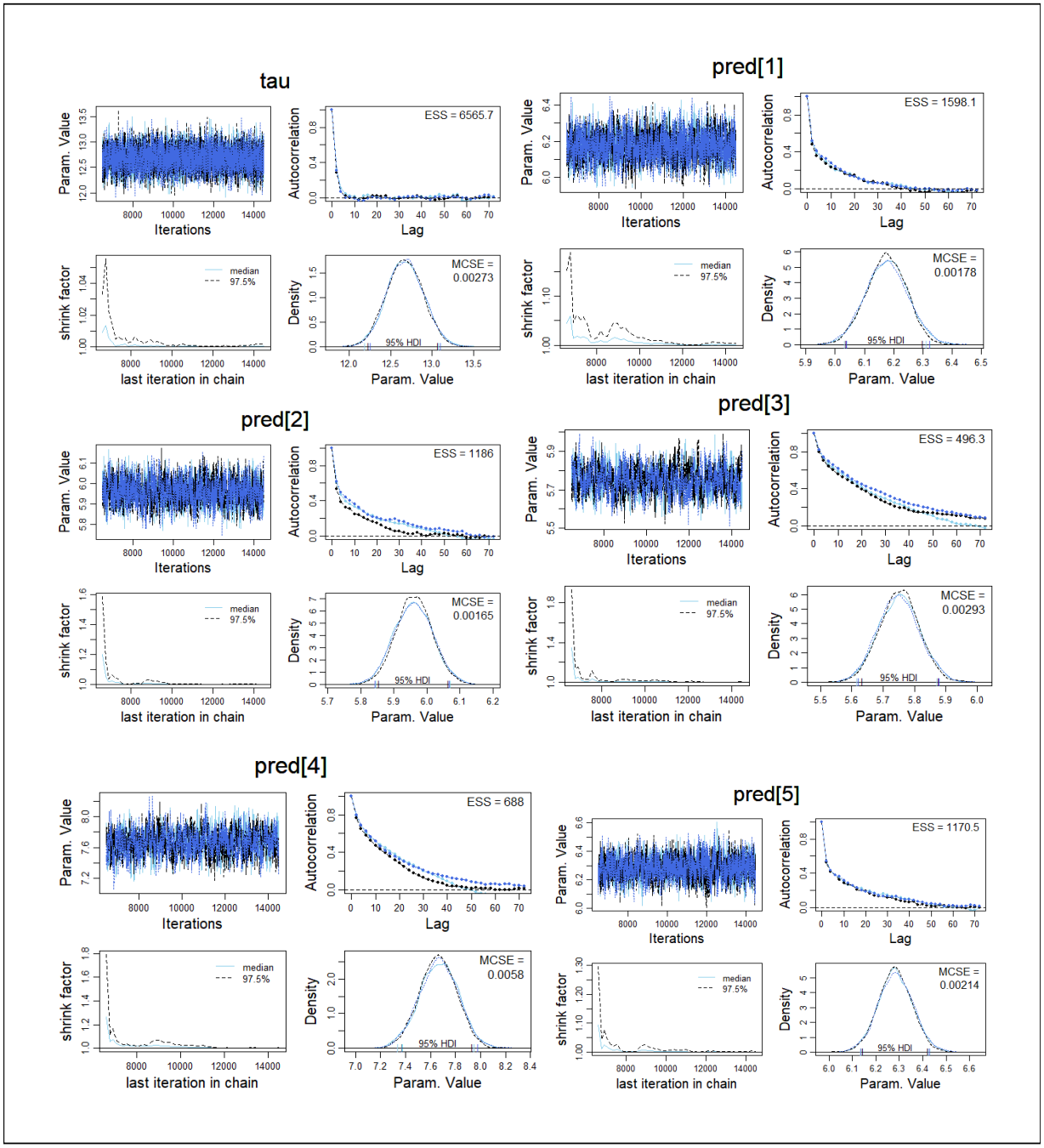

Figure 19. MCMC Diagnostics - Run 1

(B) Rscript

```
graphics.off() # This closes all of R's graphics windows.
rm(list=ls())  # Careful! This clears all of R's memory!
library(ggplot2)
library(ggpubr)
library(ks)
library(rjags)
library(runjags)
setwd("D:/Sem 4/Bayesian/A2")
source("DBDA2E-utilities.R")

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

smryMCMC_HD = 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

  # summaryInfo = rbind( summaryInfo ,
  #                      "tau" = summarizePost( mcmcMat[,"tau"] ) )
  if ( !is.null(saveName) ) {
    write.csv( summaryInfo , file=paste(saveName,"SummaryInfo.csv",sep="") )
  }
  return( summaryInfo )
}

#=============================================================================
===

plotMCMC_HD = function( codaSamples , data , xName="x" , yName="y" ,
                        showCurve=FALSE ,  pairsPlot=FALSE , compVal = NULL,
                        saveName=NULL , saveType="jpg" ) {
  # showCurve is TRUE or FALSE and indicates whether the posterior should
  #   be displayed as a histogram (by default) or by an approximate curve.
  # pairsPlot is TRUE or FALSE and indicates whether scatterplots of pairs
  #   of parameters should be displayed.
```

```
  #-------------------------------------------------------------------------
---
  y = data[,yName]
  x = as.matrix(data[,xName])
  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]"] # Added by Brian
  pred2 = mcmcMat[,"pred[2]"] # Added by Brian
  pred3 = mcmcMat[,"pred[3]"] # Added by Brian
  pred4 = mcmcMat[,"pred[4]"] # Added by Brian
  pred5 = mcmcMat[,"pred[5]"] # Added by Brian

  #-------------------------------------------------------------------------
---
  # Compute R^2 for credible parameters:
  YcorX = cor( y , x ) # correlation of y with each x predictor
  Rsq = zbeta %*% matrix( YcorX , ncol=1 )
  #-------------------------------------------------------------------------
---
  if ( pairsPlot ) {
    # Plot the parameters pairwise, to see correlations:
    openGraph()
    nPtToPlot = 1000
    plotIdx = floor(seq(1,chainLength,by=chainLength/nPtToPlot))
    panel.cor = function(x, y, digits=2, prefix="", cex.cor, ...) {
      usr = par("usr"); on.exit(par(usr))
      par(usr = c(0, 1, 0, 1))
      r = (cor(x, y))
      txt = format(c(r, 0.123456789), digits=digits)[1]
      txt = paste(prefix, txt, sep="")
      if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
      text(0.5, 0.5, txt, cex=1.25 ) # was cex=cex.cor*r
    }
    pairs( cbind( beta0 , beta , tau )[plotIdx,] ,
           labels=c( "beta[0]" ,
                     paste0("beta[",1:ncol(beta),"]\n",xName) ,
                     expression(tau) ) ,
           lower.panel=panel.cor , col="skyblue" )
    if ( !is.null(saveName) ) {
      saveGraph( file=paste(saveName,"PostPairs",sep=""), type=saveType)
    }
  }
  #-------------------------------------------------------------------------
---
  # Marginal histograms:

  decideOpenGraph = 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( 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( 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( panelCount ,
saveName=paste0(saveName,"PostMarg") )
  histInfo = plotPost( tau , cex.lab = 1.75 , showCurve=showCurve ,
                       xlab=bquote(tau) , main=paste("Scale") )
```

```
  panelCount = decideOpenGraph( 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( panelCount ,
saveName=paste0(saveName,"PostMarg") )
  histInfo = plotPost( pred1 , cex.lab = 1.75 , showCurve=showCurve ,
                       xlab="pred1" , main="Prediction 1" ) # Added by Brian
  panelCount = decideOpenGraph( panelCount ,
saveName=paste0(saveName,"PostMarg") )
  histInfo = plotPost( pred2 , cex.lab = 1.75 , showCurve=showCurve ,
                       xlab="pred2" , main="Prediction 2" ) # Added by Brian
  panelCount = decideOpenGraph( panelCount ,
saveName=paste0(saveName,"PostMarg") )
  histInfo = plotPost( pred3 , cex.lab = 1.75 , showCurve=showCurve ,
                       xlab="pred3" , main="Prediction 3" ) # Added by Brian
  panelCount = decideOpenGraph( panelCount ,
saveName=paste0(saveName,"PostMarg") )
  histInfo = plotPost( pred4 , cex.lab = 1.75 , showCurve=showCurve ,
                       xlab="pred4" , main="Prediction 4" ) # Added by Brian
  panelCount = decideOpenGraph( panelCount , finished=TRUE ,
saveName=paste0(saveName,"PostMarg") )
  histInfo = plotPost( pred5 , cex.lab = 1.75 , showCurve=showCurve ,
                       xlab="pred5" , main="Prediction 5" ) # Added by Brian

  # Standardized scale:
  panelCount = 1
  panelCount = decideOpenGraph( 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( 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( panelCount ,
saveName=paste0(saveName,"PostMargZ") )
  histInfo = plotPost( zVar , cex.lab = 1.75 , showCurve=showCurve ,
                       xlab=bquote(z*tau) , main=paste("Scale") )
  panelCount = decideOpenGraph( 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( panelCount , finished=TRUE ,
saveName=paste0(saveName,"PostMargZ") )

  #-------------------------------------------------------------------------
---
}

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

```
myData <- read.csv("Assignment2PropertyPrices.csv")
head(myData)

summary_stats<-t(summary(myData))

write.csv(summary_stats2, "summary_stats.csv")



par(mfrow=c(2,2))

b1<-barplot(table(myData$Bedrooms), col="lightsalmon", main = "Frequency of
Bedrooms", xlab= "Bedrooms",ylab= "Frequency")

b2<-barplot(table(myData$Bathrooms), col="lightsalmon", main = "Frequency of
Bathrooms", xlab= "Bathrooms", ylab= "Frequency")

b3<-barplot(table(myData$CarParks), col="lightsalmon", main = "Frequency of
Car Parks", xlab= "Car Parks", ylab= "Frequency")

b4<-barplot(table(myData$PropertyType), col="lightsalmon", main = "Frequency
of Property Types", xlab= "Property Type", ylab= "Frequency")

par(mfrow=c(1,1))


# Scatter plots
p1 <- ggplot(myData, aes(x=Area, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Area") +
  ylab("Sale Price")

p2 <- ggplot(myData, aes(x=Bedrooms, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Bedrooms") +
  ylab("Sale Price")


p3 <- ggplot(myData, aes(x=Bathrooms, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Bathrooms") +
  ylab("Sale Price")


p4 <- ggplot(myData, aes(x=CarParks, y=SalePrice.100K.)) +
  geom_point() +
  xlab("CarParks") +
  ylab("Sale Price")


p5 <- ggplot(myData, aes(x=PropertyType, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Property Type") +
  ylab("Sale Price")
```

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

# Histogram
hist(myData$SalePrice.100K., main= " Histogram of the dependent variable",
xlab = "SalePrice.100K.")
# Kernel density estimation
plot(kde(myData$SalePrice.100K.), xlab = "SalePrice.100K.") # with default
settings


# THE DATA.
y = myData[,"SalePrice.100K."]
x =
as.matrix(myData[,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)

# Specify the data in a list, for later shipment to JAGS:
dataList <- list(
  x = x ,
  y = y ,
  xPred = xPred ,
  Nx = dim(x)[2] ,
  Ntotal = dim(x)[1]
)

# Initials for Run 2 and Run 3
initsList <- list(
  zbeta0 = 2000,
  zbeta = c(100, 2, 1,1, 0.5),
  Var = 12000000
)

# WE WILL RUN THE MODEL WITH SCALING!

# THE MODEL.
modelString = "
# Standardize the data:
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]
    }
  }
}
# Specify the model 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( 0000.90/xsd[1] , 1/(0.1/xsd[1]^2) ) # 1/ variance for
normal distribution
  zbeta[2] ~ dnorm( 1.00 , 1/(4/xsd[2]^2) ) # 1/ variance for normal
distribution
  zbeta[3] ~ dnorm( 0 , 1/4 ) # 1/ variance for normal distribution
  zbeta[4] ~ dnorm( 1.20/xsd[4] , 1/(0.1/xsd[4]^2) ) # 1/ variance for normal
distribution
  zbeta[5] ~ dnorm( -1.50/xsd[5] , 1/(0.1/xsd[5]^2) ) # 1/ variance for
normal distribution

  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]
  }
  # pred[1] <- beta0 + beta[1] * xPred[1,1] + beta[2] * xPred[1,2] + beta[3]
* xPred[1,3] + beta[4] * xPred[1,4] + beta[5] * xPred[1,5]
  # pred[2] <- beta0 + beta[1] * xPred[2,1] + beta[2] * xPred[2,2] + beta[3]
* xPred[2,3] + beta[4] * xPred[2,4] + beta[5] * xPred[2,5]
  # pred[3] <- beta0 + beta[1] * xPred[3,1] + beta[2] * xPred[3,2] + beta[3]
* xPred[3,3] + beta[4] * xPred[3,4] + beta[5] * xPred[3,5]
  # pred[4] <- beta0 + beta[1] * xPred[4,1] + beta[2] * xPred[4,2] + beta[3]
* xPred[4,3] + beta[4] * xPred[4,4] + beta[5] * xPred[4,5]
  # pred[5] <- beta0 + beta[1] * xPred[5,1] + beta[2] * xPred[5,2] + beta[3]
* xPred[5,3] + beta[4] * xPred[5,4] + beta[5] * xPred[5,5]

}
" # close quote for modelString
# Write out modelString to a text file
writeLines( modelString , con="TEMPmodel.txt" )


parameters = c( "zbeta0" ,  "zbeta" , "beta0" ,  "beta" ,  "tau", "zVar") #
Here beta is a vector!


#Run 1 Settings
adaptSteps = 1500  # Number of steps to "tune" the samplers
```

```
burnInSteps = 5000
nChains = 3
thinSteps = 3 # First run for 3
numSavedSteps = 4000
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )

#Run 2 Settings
adaptSteps = 1500
burnInSteps = 5000
nChains = 3
thinSteps = 14
numSavedSteps = 6500
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )

#Run 3 Settings
adaptSteps = 1500
burnInSteps = 5000
nChains = 3
thinSteps = 20
numSavedSteps = 4800
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )

# Parallel run

runJagsOut <- run.jags( method="parallel" ,
                        model="TEMPmodel.txt" ,
                        monitor=c( "zbeta0" ,  "zbeta" , "beta0" ,  "beta" ,
"tau", "zVar", "pred")  ,
                        data=dataList ,
                        n.chains=nChains ,
                        #inits=initsList , #For Run 2 and 3
                        adapt=adaptSteps ,
                        burnin=burnInSteps ,
                        sample=numSavedSteps ,
                        thin=thinSteps , summarise=FALSE , plots=FALSE )
codaSamples = as.mcmc.list( runJagsOut )

save.image(file="rEnvironment.RData")
load(file="rEnvironment_4.RData")

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]" )
diagMCMC( codaSamples , parName="zbeta0" )
diagMCMC( codaSamples , parName="zbeta[1]" )

compVal <- data.frame("beta0" = NA, "beta[1]" = NA, "beta[2]" = NA, "beta[3]"
= NA, "beta[4]" =  NA, "beta[5]" =  NA, "tau" = NA , check.names=FALSE)
```

```
summaryInfo <- smryMCMC_HD( codaSamples = codaSamples , compVal = compVal )
print(summaryInfo)


plotMCMC_HD( codaSamples = codaSamples , data = myData,
xName=c("Area","Bedrooms","Bathrooms","CarParks","PropertyType") ,
             yName="SalePrice.100K.", compVal = compVal)

# ============ Predictive check ============
coefficients <- summaryInfo[8:13,3] # Get the model coefficients out
Variance <- summaryInfo[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= 10000,shape=meanGamma^2/Variance, rate =
meanGamma/Variance)

# Display the density plot of observed data and posterior distribution:
predicted <- data.frame(elapsed = randomData)
observed <- data.frame(elapsed = y)
predicted$type <- "Predicted1"   #"Predicted2","Predicted3"
observed$type <- "Observed"
dataPred <- rbind(predicted, observed)

ggplot(dataPred, aes(elapsed, fill = type)) + geom_density(alpha = 0.2)
```