Math 2269: Applied Bayesian Statistics (2050)
ASSIGNMENT 2

Tarana Tabassum
S3802509
Date of Submission: 22 September 2020

Abstract

Prediction of property prices is an important issue and active research area nowadays. This report conducts Bayesian analyses to predict property sale prices in Melbourne using a fabricated dataset with real data. In this report, taking a quick look into the descriptive statistics, we choose an appropriate mathematical model implementing the suggested prior information on RJags to find out the suitable posterior distribution. Later, the model diagnostics are analysed; posterior distribution for regression parameters and Tau are explained. Lastly, a prediction model is developed through which the Melbourne property sale prices are estimated along with coefficients of regression model & predicted sale price.

Table of Contents

1. Introduction

Prediction of property prices is an important issue and active research area nowadays. Data scientists are coming up with different approaches for the prediction of property prices in Australia using regression methods based on either statistical methods or deep/machine learning approaches. We are working in a progressive way to conduct Bayesian analyses to predict property sale prices in Melbourne using a fabricated dataset with real data after a number of other analyses for the population distributions of the variables included in the dataset. This step is undertaken to comply with the rules around the use and distribution of the real data.

2. Objective

This report's objective is to model the sale prices of properties in Melbourne using the predictors given in the dataset and expert knowledge from a real estate agent.

3. Methodology

In this report, for each predictor, expert information and the degree of belief in the prior information is given as follows:

- **Area**: Every m^2 increase in land size increases the sales price by 90 AUD. This is a **very strong** expert knowledge.
- **Bedrooms**: Every additional bedroom increases the sales price by 100,000AUD. This is a **weak** expert knowledge.
- **Bathrooms**: There is **no expert knowledge** on the number of bathrooms.
- **Car Parks**: Every additional car space increases the sales price by 120,000AUD. This is a **strong** expert knowledge.
- **Property Type**: If the property is a unit, the sale price will be 150,000 AUD less than that of a house on the average. This is a **very strong** expert knowledge.

To identify what type of model implementation is required to be applied for this report's purpose, we first take a quick look into the descriptive statistics. We choose an appropriate mathematical model described later, as well as a model diagram to be implemented on R Studio. We then implement the prior information on RJags to find out the suitable posterior distribution. Later, the model diagnostics are analysed; posterior distribution for regression parameters and Tau are explained. Later a prediction model is developed through which the Melbourne property sale prices are estimated along with coefficients of regression model & predicted sale price.

4. Results & Interpretation

4.1 Descriptive Analysis

The variables of the dataset used for this purpose are described below -

- SalePrice100K.: Sale Price in 100K AUD
- 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)
-

We plot scatter plots for dependent and independent variables and find the following output -

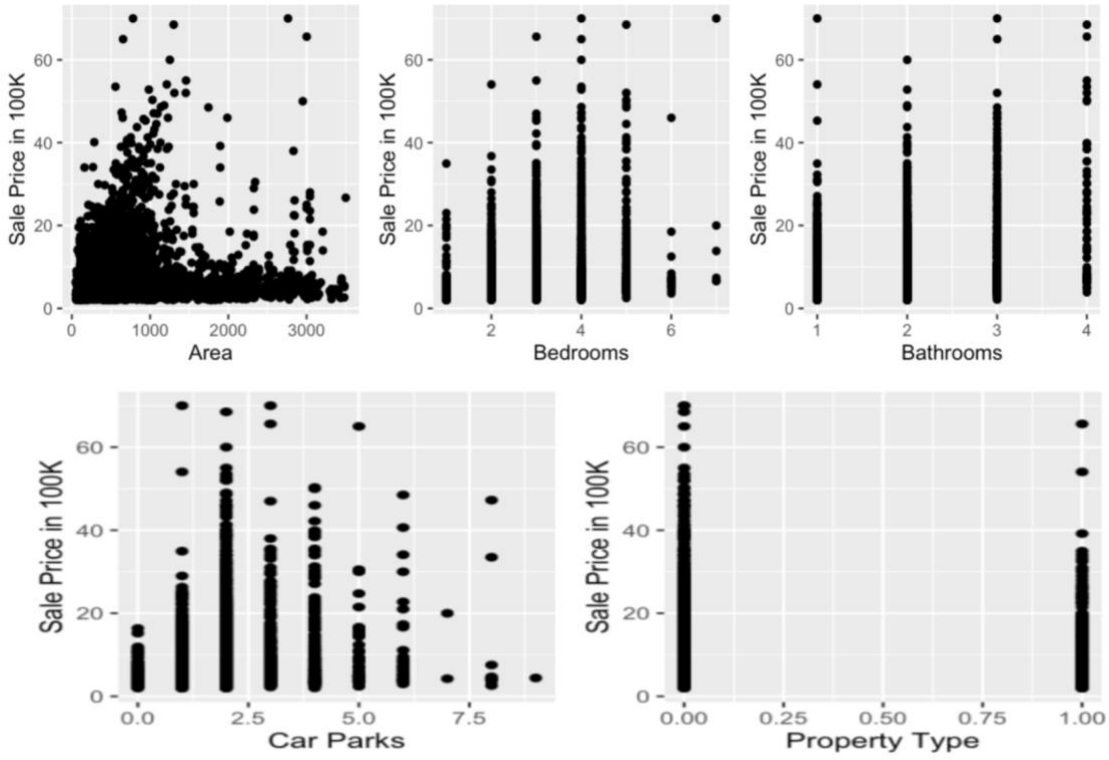

Figure 1: Scatter plots on the variables of the dataset

The scatter plots are projected here which demonstrate some outliers.

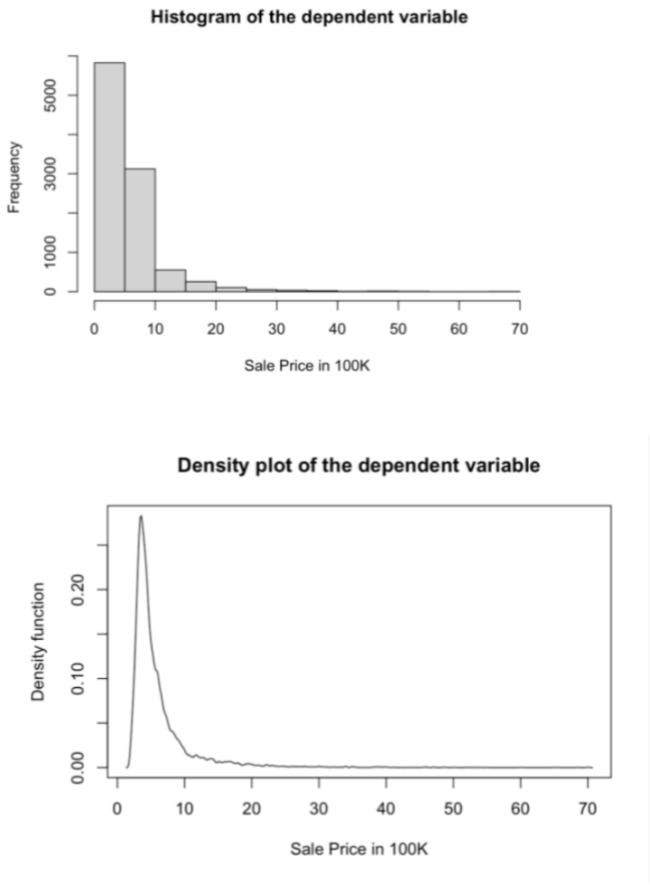

Figure 2: Histogram and density plot based on the dependent variable in dataset

It can be observed that the dependent variable (SalePrice 100K) has started from zero, without any negative value. Also, the density plot is similar to the gamma model. This suggests that gamma likelihood can be a good fit for this dependent variable.

4.2 Mathematical Model

Here, gamma distribution is selected for likelihood based on the descriptive analysis. The suggested mathematical model follows -

$$Y=\beta 0+\beta 1X1+\beta 2X2+\beta 3X3+\beta 4X4+\beta 5X5+\varepsilon,\ \varepsilon \sim Gamma(\mu^2/\sigma^2, \mu/\sigma^2)$$

Here,
Y = dependent variable (Sale Price)
μ: mean
σ^2: the variance

4.3 JAGS Model

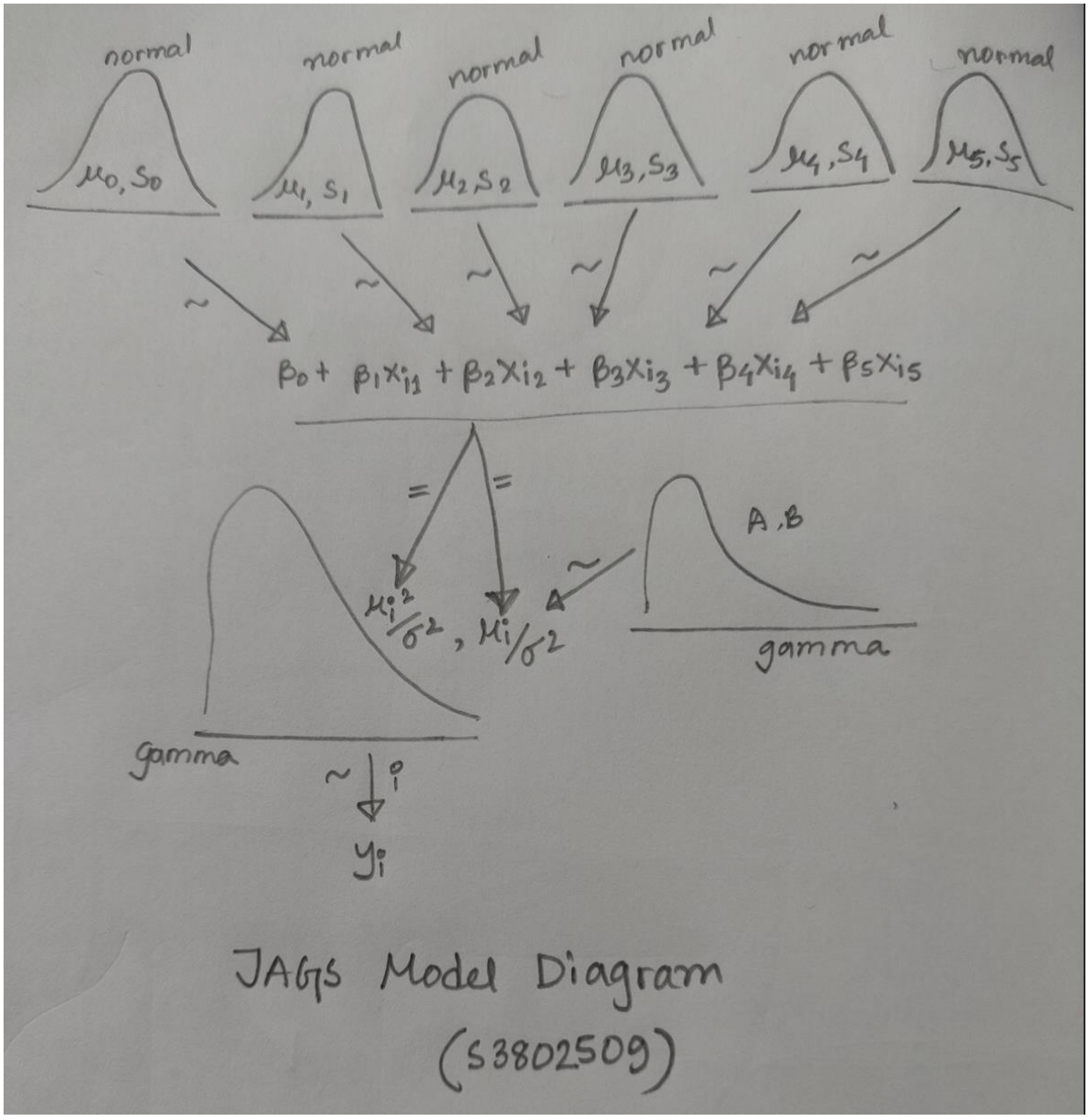

Figure 3: JAGS model diagram

In the above diagram, we see that the six regression normally distributed parameters. Each observation is in gamma distribution where parameters are reparametrized in terms of mean and variance. Gamma prior is on variance.

4.4 Specification of Prior Distribution

The prior information used here in this reporting is based on the expert knowledge provided before. Following is a screenshot from RStudio -

```
zbeta0 ~ dnorm( 0 , 1/2^2 )
zbeta[1] ~ dnorm( (90/100000)/xsd[1] , 1/(0.05/xsd[1]^2) )
zbeta[2] ~ dnorm( 1/xsd[2] , 1/(6/xsd[2]^2)
zbeta[3] ~ dnorm( 0 , 1/9 )
zbeta[4] ~ dnorm( 1.2/xsd[4] , 1/(1.3/xsd[4]^2) )
zbeta[5] ~ dnorm( 1.5/xsd[5] , 1/(0.05/xsd[5]^2) )
```

Figure 4: Prior Information Screenshot

4.5 Implementation of Model

According to the expected proper posterior distribution of the model diagram provided before, following are the adaptive step, burn-in step, number of chains, number of thinning, number of saved steps entered in RStudio -

```
adaptSteps = 1900
burnInSteps = 6000
nChains = 3
thinSteps = 23
numSavedSteps = 6000
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )
```

Figure 5: Implementing the model in RStudio

4.6 Model Diagnostics

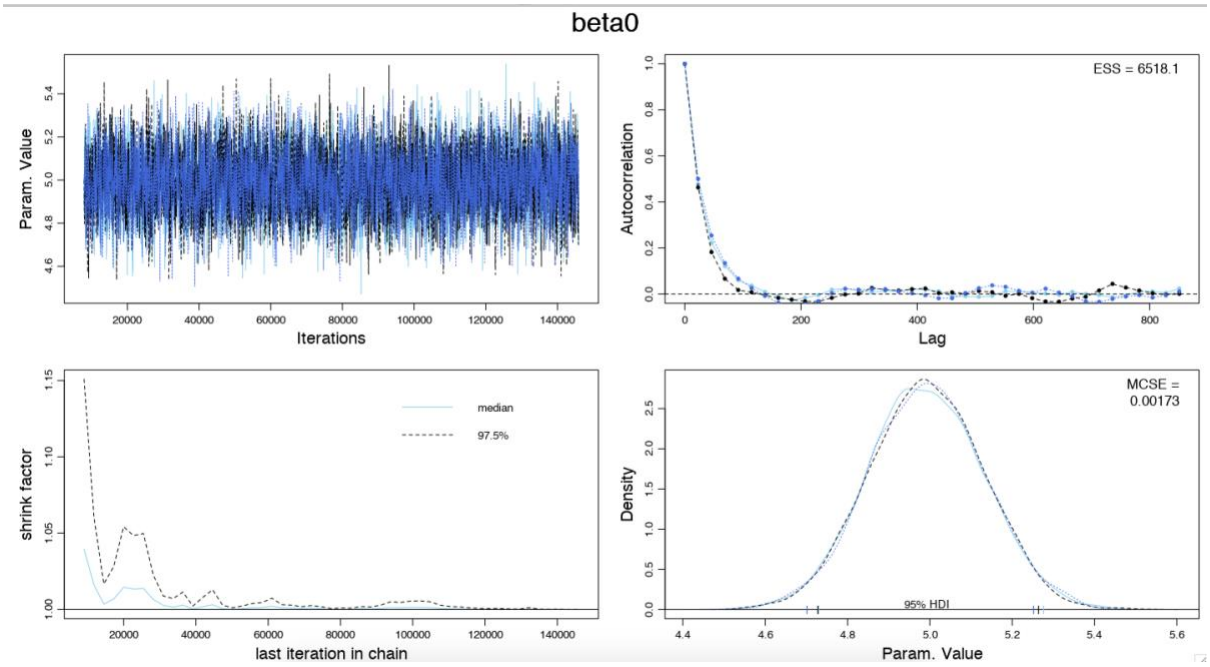

Figure 6: Diagnostics of $\beta 0$

From the figure above, the diagnostics for $\beta 0$ can be observed. Here, good amount of overlapping in trace plots can be observed. A low auto correlation (due to machine incompetence), higher Estimated Sample Size (ESS) can also be seen according to our expected proper posterior. The shrink factor is observed to be less than 1.2, along with overlapping density plot with overlapping HDIs and very low Monte Carlo Standard Error (MCSE) 0.00173.

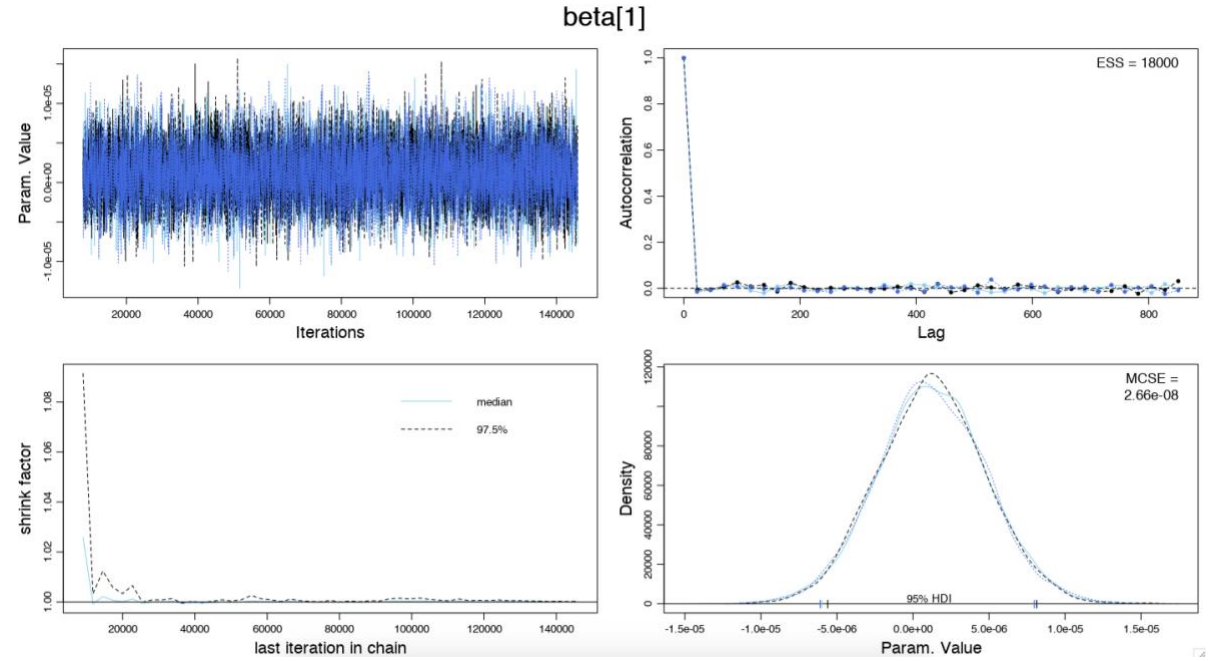

Figure 7: Diagnostics of $\beta 1$

From the figure above, the diagnostics for $\beta 1$ can be observed. Here, good amount of overlapping in trace plots can be observed. No auto correlation, higher Estimated Sample Size (ESS) can also be seen according to our expected proper posterior. The shrink factor is observed to be less than 1.2, along with overlapping density plot with overlapping HDIs and very low Monte Carlo Standard Error (MCSE).

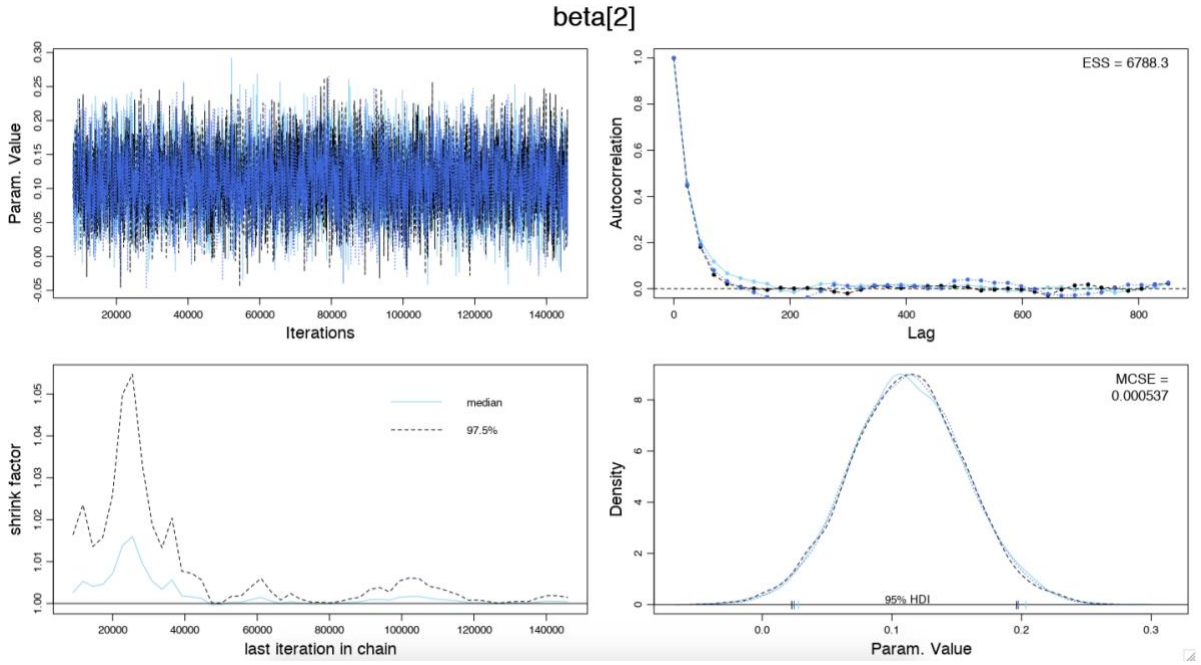

Figure 8: Diagnostics of $\beta 2$

Figure 8 shows the diagnostics of $\beta 2$, it shows overlapping trace plots, low auto correlation (due to machine incompetence), higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE).

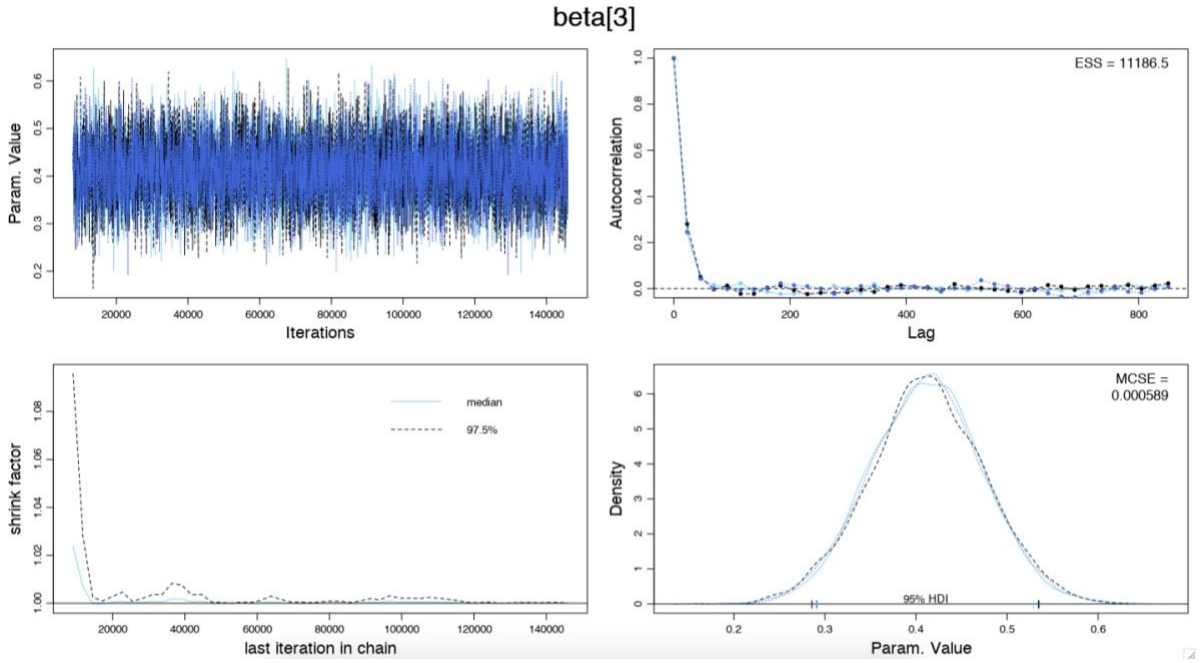

Figure 9: Diagnostics of $\beta 3$

Figure 9 shows the diagnostics of $\beta 3$, it shows overlapping trace plots, no auto correlation, higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE).

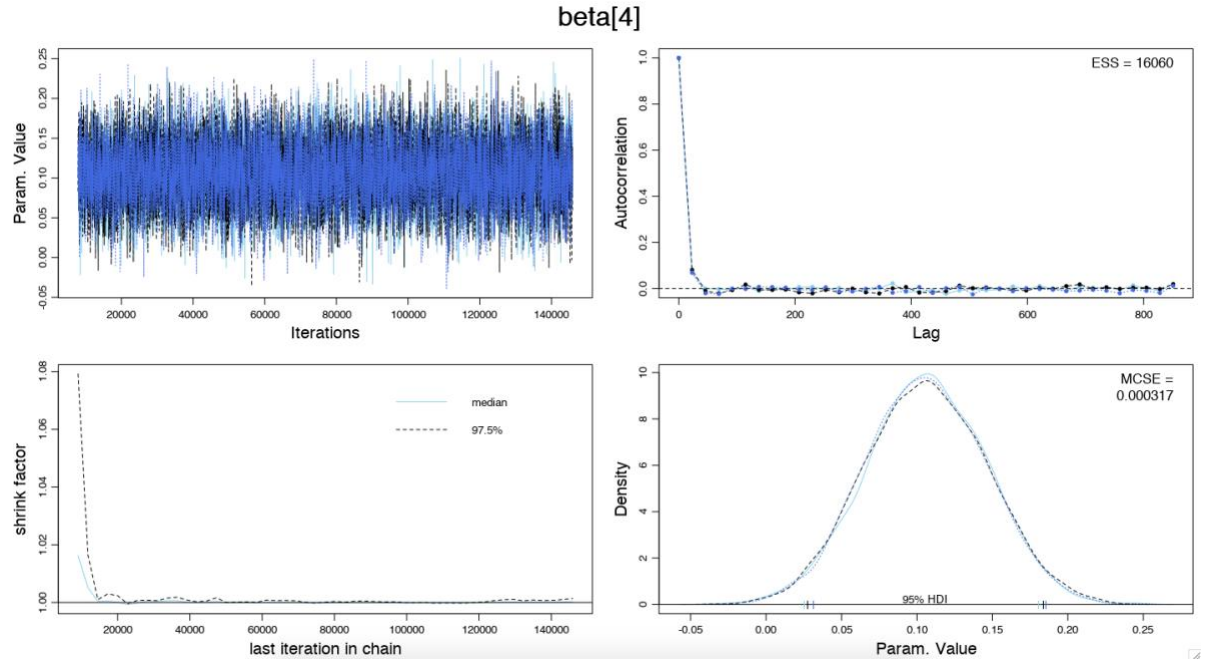

Figure 10: Diagnostics of $\beta 4$

Figure 10 shows the diagnostics of $\beta 4$, it shows overlapping trace plots, no auto correlation, higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE).

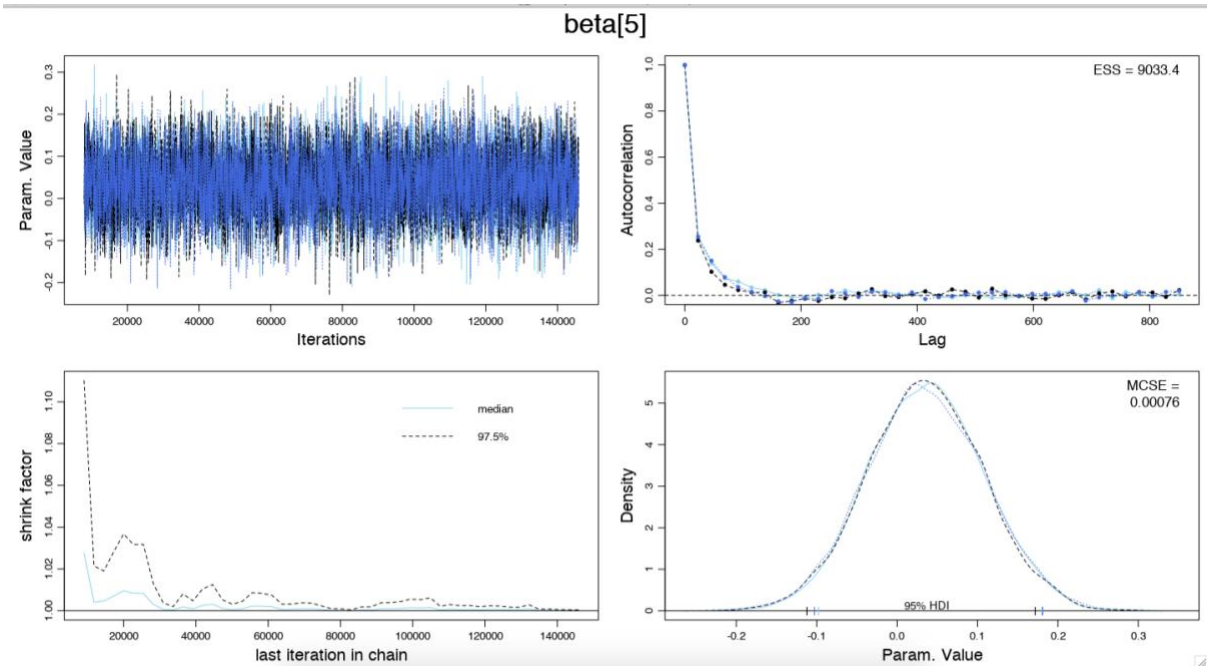

Figure 11: Diagnostics of $\beta 5$

Figure 11 shows the diagnostics of $\beta 5$, it shows overlapping trace plots, low auto correlation due to mainly machine incompetence, higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE).

Diagnostics for τ:

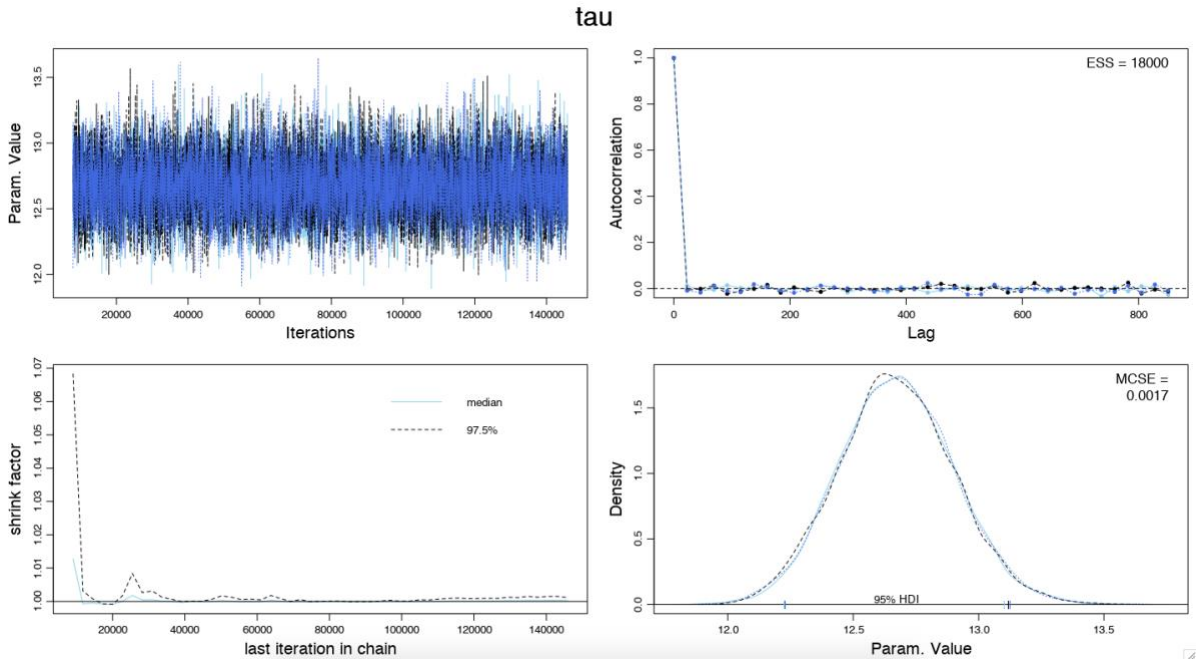

Figure 12: Diagnostics of τ

Figure 12 shows the diagnostics of τ, it shows overlapping trace plots, no auto correlation, higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE).

4.7 Posterior Distribution and Bayesian estimation of regression parameters and τ

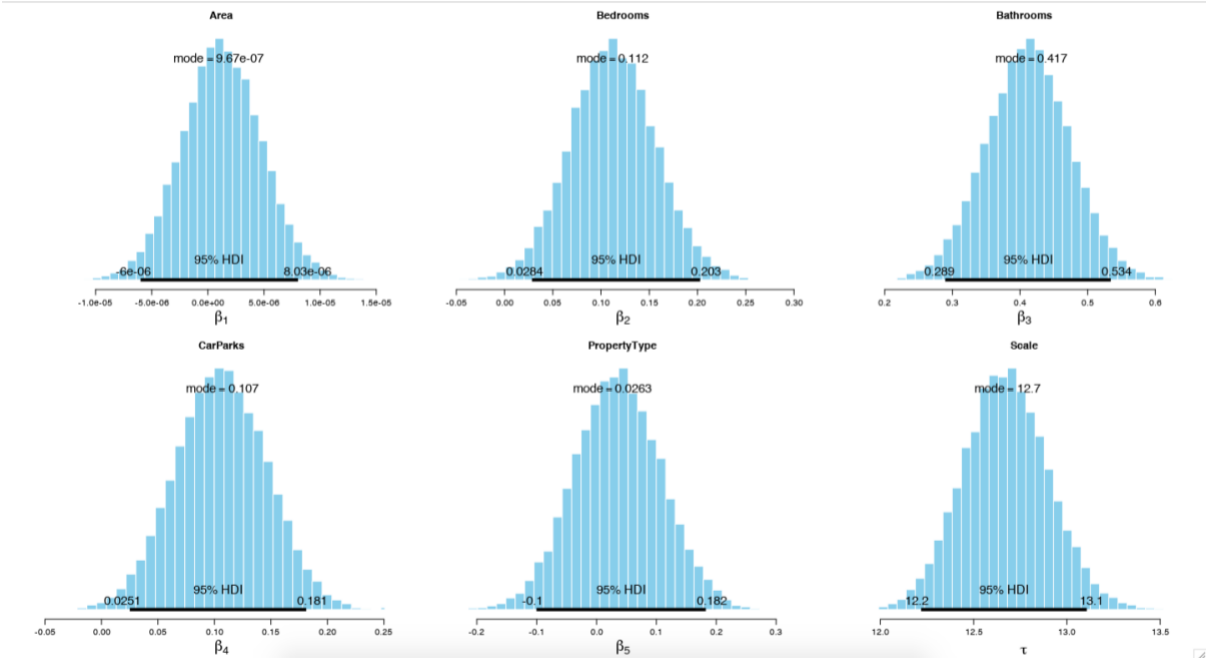

Figure 13: Posterior distribution - regression parameters and τ

Figure 13 shows the posterior distributions for regression parameters and for TAU. These are explained below:

β1: The value of β1 is 9.67e-07 and it lies in the 95% HDI : -6e-06 to 8.03e-06.

β2: The value of β2 is .112 and it lies in the 95% HDI : 0.0284 to 0.203.

β3: The value of β3 is 0.417 and it lies in the 95% HDI : 0.289 to 0.534.

β4: The value of β4 is 0.107 and it lies in the 95% HDI : 0.0251 to 0.181.

β5: The value of β5 is 0.0263 and it lies in the 95% HDI : -0.1 to 0.182.

τ: The value of τ is 12.7 and it lies in the 95% HDI : 12.2 to 13.1.

4.8 Prediction diagnostics

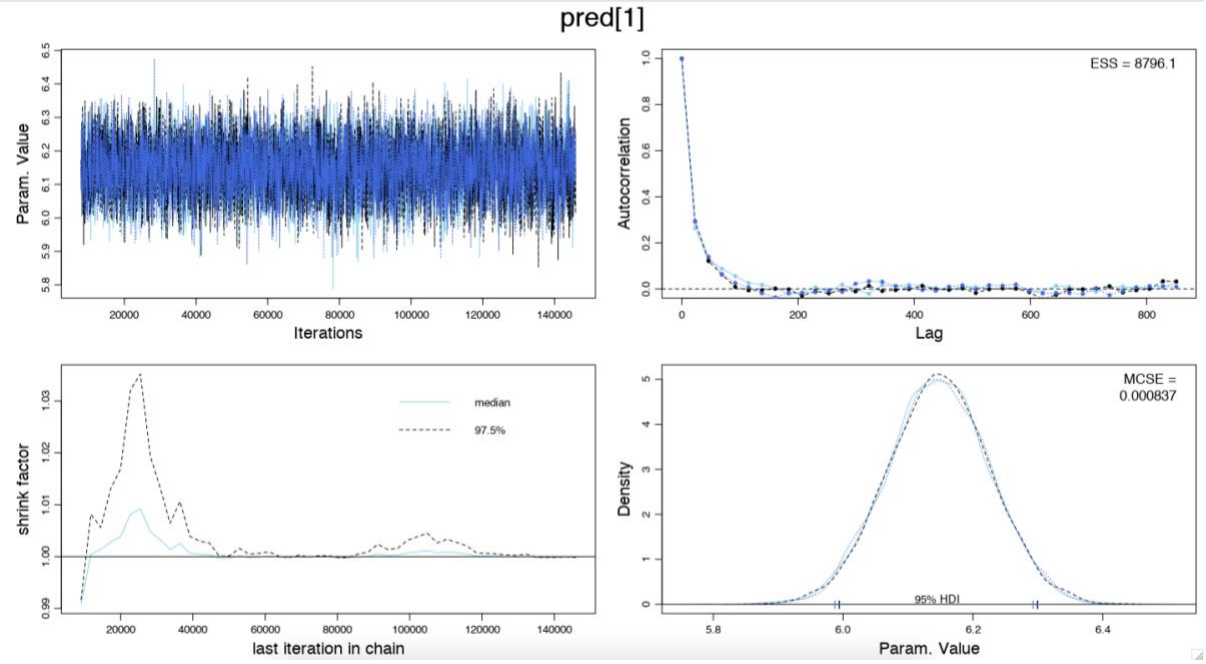

Figure 14: Diagnostics of prediction 1

Figure 14 shows the diagnostics of prediction 1, it shows overlapping trace plots, low auto correlation (due to machine incompetence), higher Estimated Sample Size (ESS), shrink factor less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE).

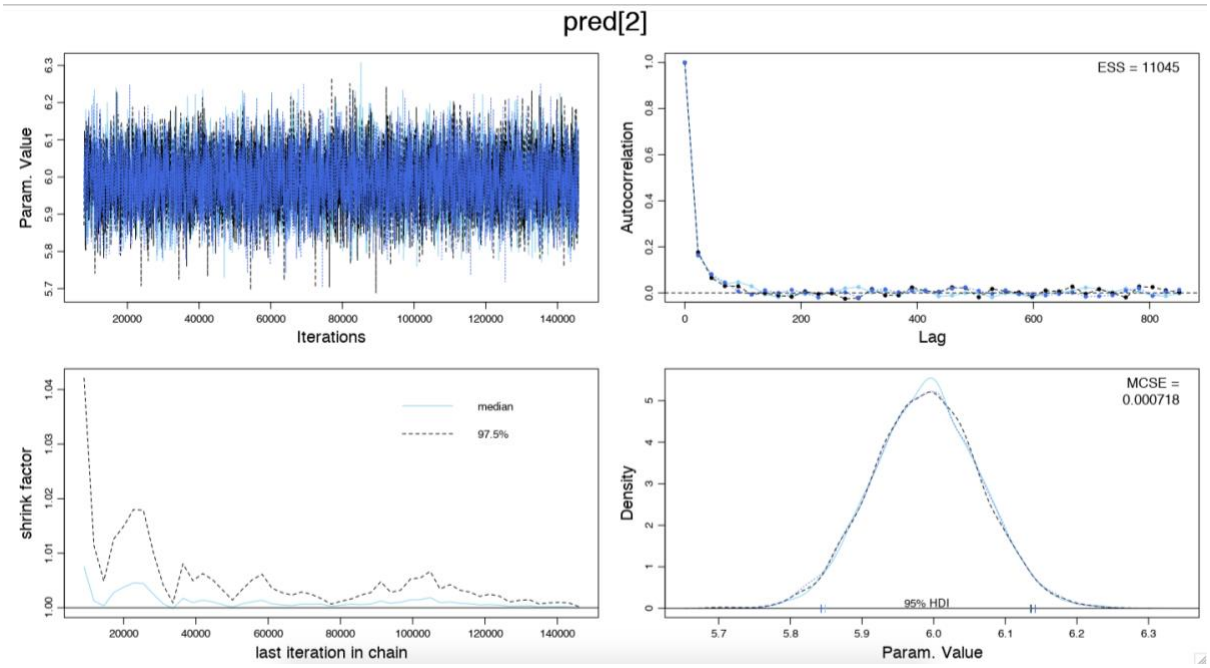

Figure 15: Diagnostics of prediction 2

Figure 15 shows the diagnostics of prediction 2, it shows overlapping trace plots, low auto correlation (due to machine incompetence), higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE) 0.000718.

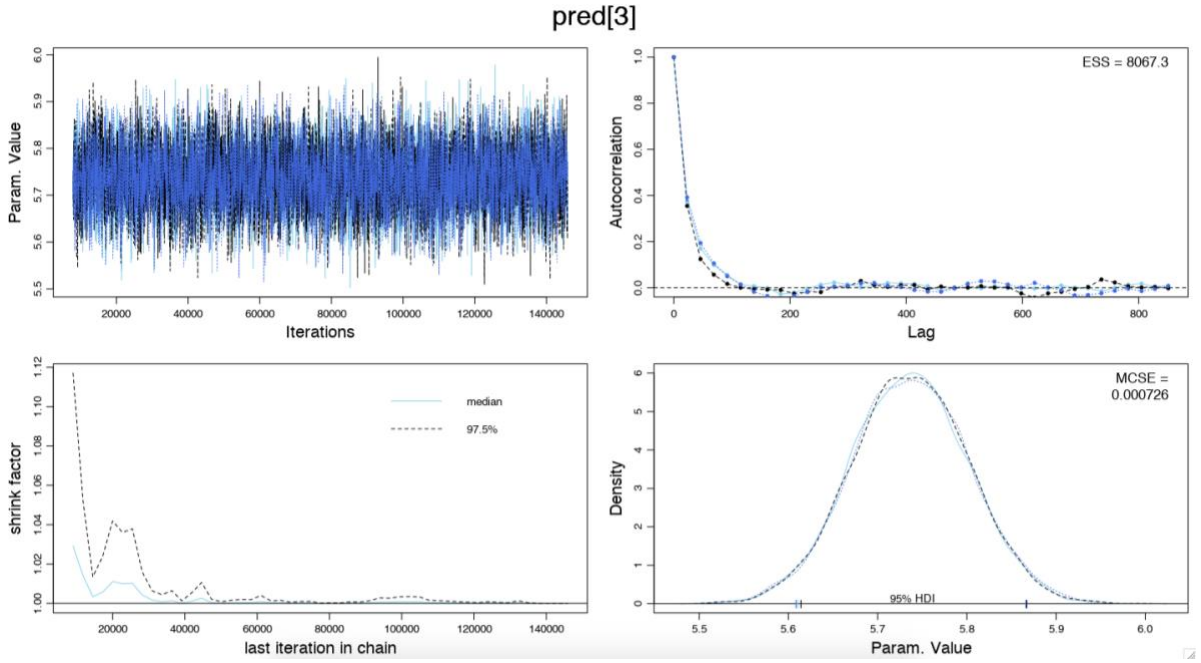

Figure 16: Diagnostics of prediction 3

Figure 16 shows the diagnostics of prediction 3, it shows overlapping trace plots, low auto correlation (due to machine incompetence), higher Estimated Sample Size (ESS), shrink factor less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE) 0.000726.

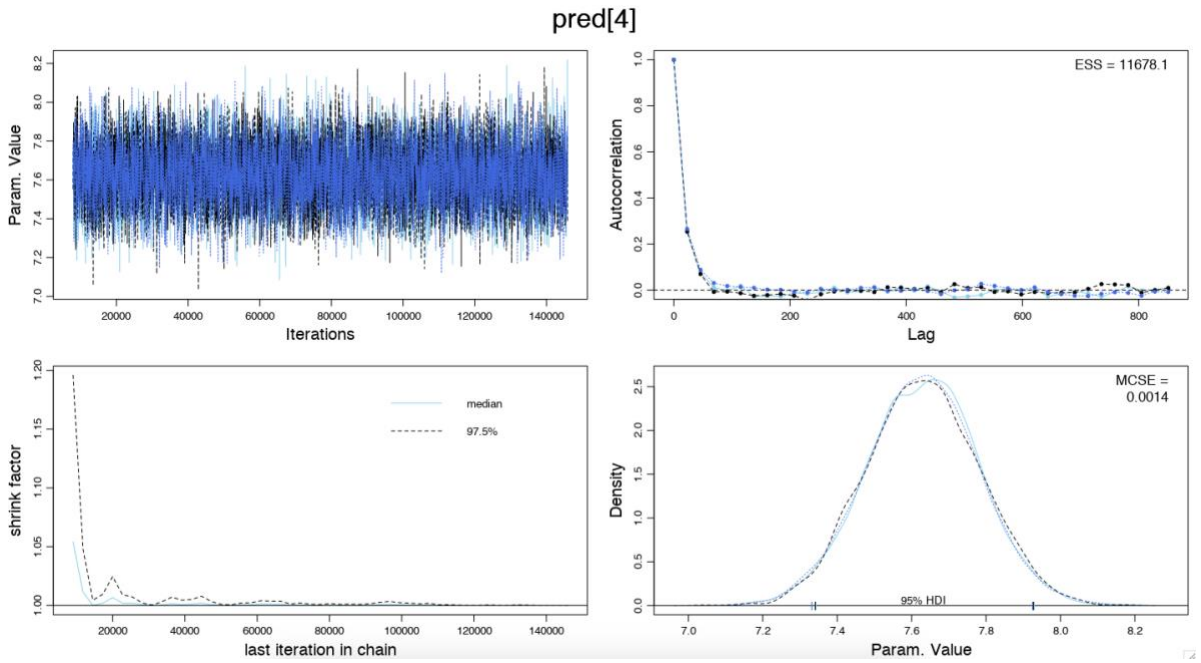

Figure 17: Diagnostics of prediction 4

Figure 17 shows the diagnostics of prediction 4, it shows overlapping trace plots, low auto correlation (due to machine incompetence), higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE) 0.0014.

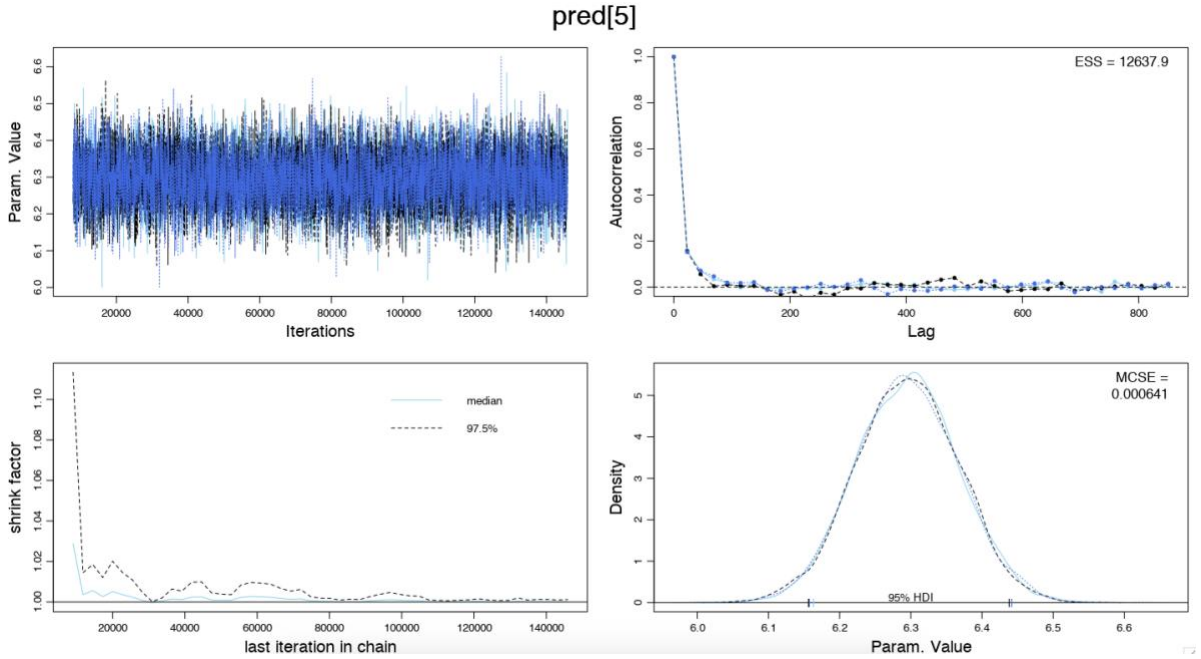

Figure 18: Diagnostics of prediction 5

Figure 18 shows the diagnostics of prediction 5, it shows overlapping trace plots, very low auto correlation (due to machine incompetence), higher Estimated Sample Size (ESS), shrink factor is less than 1.2, overlapping density plot with overlapping HDIs and low Monte Carlo Standard Error (MCSE) 0.000641.

4.9 Predicted Sale Price

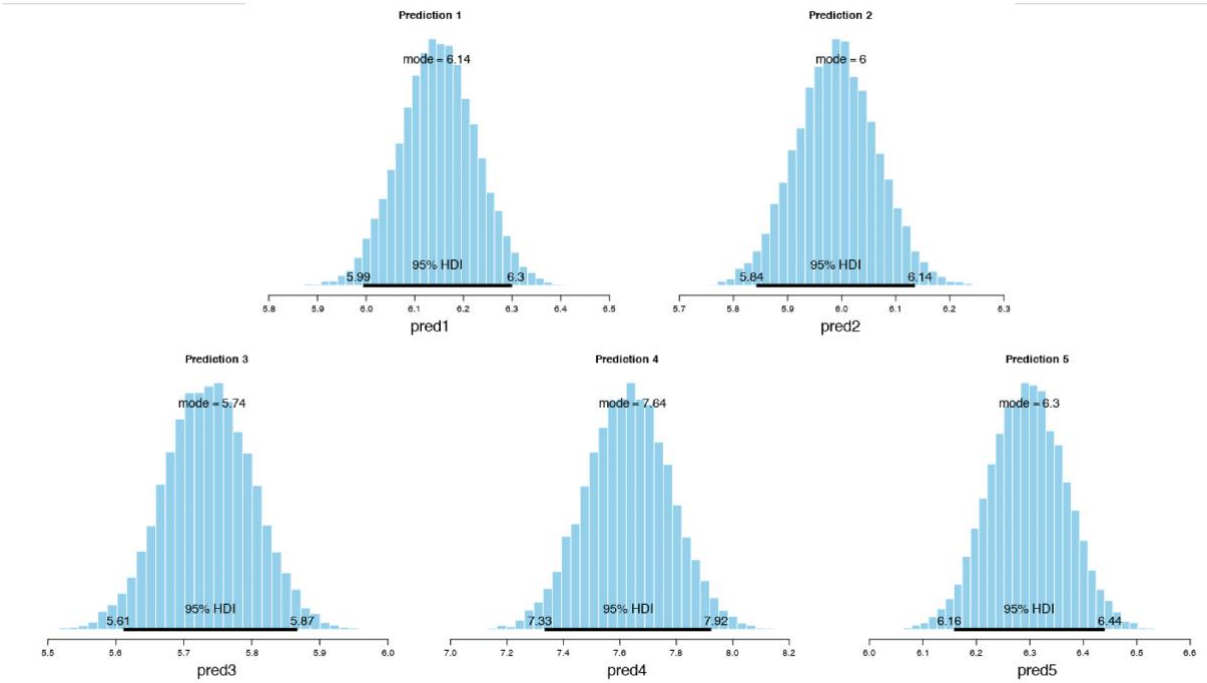

Figure 19: Posterior Distribution for Prediction

Figure 19 is displaying predicted sale prices of the Melbourne properties. These mode values in every posterior distribution of predictions are the predicted sale prices in 100,000 of AUD.

4.10 Summary

	Mean	Median	Mode	ESS	HDImass	HDIlow	HDIhigh	CompVal
CHAIN	2.000000e+00	2.000000e+00	3.002868e+00	1.5	0.95	1.00000e+00	3.00000e+00	NA
zbeta0	9.743466e-01	9.741580e-01	9.735977e-01	6898.4	0.95	9.20204e-01	1.02664e+00	NA
zbeta[1]	1.377442e-04	1.333885e-04	1.066740e-04	17717.1	0.95	-6.61252e-04	8.85649e-04	NA
zbeta[2]	1.968982e-02	1.964320e-02	1.955196e-02	7006.3	0.95	4.96974e-03	3.54891e-02	NA
zbeta[3]	4.877427e-02	4.881595e-02	4.933781e-02	11112.6	0.95	3.42684e-02	6.32496e-02	NA
zbeta[4]	1.717633e-02	1.715865e-02	1.728925e-02	16503.5	0.95	4.05279e-03	2.93097e-02	NA
zbeta[5]	3.235590e-03	3.227715e-03	2.388884e-03	9499.0	0.95	-9.10985e-03	1.65647e-02	NA
beta0	4.991810e+00	4.990845e+00	4.987964e+00	6898.3	0.95	4.71442e+00	5.25974e+00	NA
beta[1]	1.249284e-06	1.209780e-06	9.674820e-07	17717.1	0.95	-5.99729e-06	8.03248e-06	NA
beta[2]	1.124849e-01	1.122185e-01	1.116974e-01	7006.3	0.95	2.83914e-02	2.02744e-01	NA
beta[3]	4.119189e-01	4.122715e-01	4.166784e-01	11112.6	0.95	2.89411e-01	5.34169e-01	NA
beta[4]	1.062037e-01	1.060940e-01	1.069020e-01	16503.5	0.95	2.50590e-02	1.81226e-01	NA
beta[5]	3.564764e-02	3.556085e-02	2.631929e-02	9499.0	0.95	-1.00367e-01	1.82499e-01	NA
tau	1.267375e+01	1.267145e+01	1.268099e+01	18000.0	0.95	1.22185e+01	1.31068e+01	NA
zVar	4.828544e-01	4.827665e-01	4.831296e-01	18000.0	0.95	4.65511e-01	4.99354e-01	NA
pred[1]	6.147571e+00	6.147410e+00	6.144408e+00	8624.1	0.95	5.99381e+00	6.29951e+00	NA
pred[2]	5.990238e+00	5.990470e+00	5.998752e+00	10861.9	0.95	5.84244e+00	6.13545e+00	NA
pred[3]	5.736776e+00	5.736625e+00	5.741731e+00	8325.4	0.95	5.61113e+00	5.86701e+00	NA
pred[4]	7.629848e+00	7.631510e+00	7.644684e+00	11793.3	0.95	7.33342e+00	7.92339e+00	NA
pred[5]	6.295266e+00	6.295490e+00	6.298745e+00	12472.3	0.95	6.15791e+00	6.44003e+00	NA

Figure 20: Summary

Figure 20 depicts the summary statistics of this dataset, from where we find the information –

- Increase in the sale price will be 967,000 AUD when the increase in land size is 1 m2.
- Every bedroom will increase the sale price by 116,000 AUD.
- Every additional bathroom will increase the sale price by 166,000 AUD.
- An additional car park will increase the sale price by 106,000AUD.
- If the property is a unit, the sale price will be 263,000 AUD less than that of a house on average.

	Area	Bedrooms	Bathrooms	CarParks	PropertyType	EstimatedPrice
pred[1]	600	2	2	1	0	6.144408
pred[2]	800	3	1	2	1	5.998752
pred[3]	1500	2	1	1	0	5.741731
pred[4]	2500	5	4	4	0	7.644684
pred[5]	250	3	2	1	1	6.298745

Figure 21: Estimated sale price

Figure 21 depicts the estimated sale price (in 100,00 AUD) of the 5 variables of the dataset.

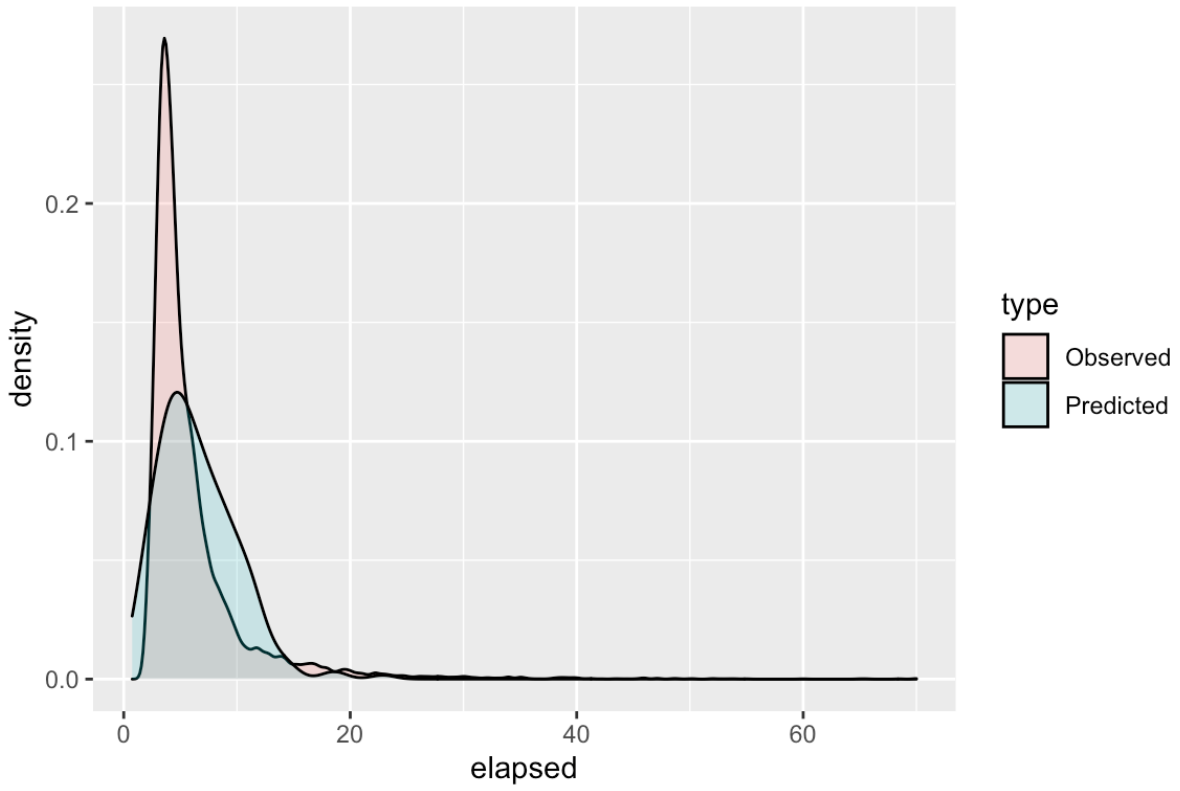

Figure 22: Goodness of fit

From the figure above, we find that the observed model and predicted model are not well fitted, since they did not overlap well enough.

5. Conclusion

Although there was quite a machine incompetence issue, this report presents us with a predicted sale price model for Melbourne's properties, including all the variances mentioned in the dataset.

6. References

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

7. Appendix

RStudio Codes:

```
graphics.off()
rm(list=ls())
library(ggplot2)
library(dplyr)
library(ggpubr)
library(ks)
library(rjags)
library(runjags)
source("DBDA2E-utilities.R")

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" ) {
  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 Demirhan
  pred2 = mcmcMat,"pred[2]"] # Added by Demirhan
  pred3 = mcmcMat,"pred[3]"] # Added by Demirhan
  pred4 = mcmcMat,"pred[4]"] # Added by Demirhan
  pred5 = mcmcMat,"pred[5]"] # Added by Demirhan

  YcorX = cor( y , x )

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

  if ( pairsPlot ) {

    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)}}

decideOpenGraph = function( panelCount , saveName , finished=FALSE , nRow=2 , nCol=3 ) {

 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)
      }
      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)
  }
}

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" )
 panelCount = decideOpenGraph( panelCount ,
finished=TRUE ,
saveName=paste0(saveName,"PostMarg") )
 histInfo = plotPost(
pred2 , cex.lab = 1.75 , showCurve=showCurve ,
                 xlab="pred2" , main="Prediction 2" )

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

 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") )


myData <- read.csv("Assignment2PropertyPrices.csv")
head(myData)
p1 <- ggplot(myData, aes(x=Area, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Aspect ratio") +
  ylab("Cost")
p2 <- ggplot(myData, aes(x=Bedrooms, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Lift-to-drag ratio") +
  ylab("Cost")
p3 <- ggplot(myData, aes(x=Bathrooms, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Weight of plane") +
  ylab("Cost")
p4 <- ggplot(myData, aes(x=CarParks, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Maximal thrust ") +
  ylab("Cost")
p5 <- ggplot(myData, aes(x=PropertyType, y=SalePrice.100K.)) +
  geom_point() +
  xlab("Maximal thrust ") +
  ylab("Cost")
figure <- ggarrange(p1, p2, p3, p4, p5, nrow = 2, ncol = 3)
figure
hist(myData$SalePrice.100K., main= " Histogram of the dependent variable", xlab = "Cost")
plot(kde(myData$SalePrice.100K.), xlab = "Cost")

y = myData[,"SalePrice.100K."]
x = as.matrix(myData[,c("Area","Bedrooms","Bathrooms","CarParks", "PropertyType")])
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, 0)
xPred[2,] = c(800, 3, 1, 2, 1)
xPred[3,] = c(1500, 2, 1, 1, 0)
xPred[4,] = c(2500, 5, 4, 4, 0)
```

```
xPred[5,] = c(250, 3, 2, 1, 1)

dataList <- list(
  x = x ,
  y = y ,
  xPred = xPred ,
  Nx = dim(x)[2] ,
  Ntotal = dim(x)[1]
)
#initsList <- list(
#  zbeta0 = 2000,
#  zbeta = c(100, 1, 1, 0.5),
#  Var = 12000000
#)

modelString = "
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 {
  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] )
  }
  zbeta0 ~ dnorm( 0 , 1/2^2 )  # 1/ variance for normal distribution
  zbeta[1] ~ dnorm( (90/100000)/xsd[1] , 1/(0.05/xsd[1]^2) ) # 1/ variance for normal distribution
  zbeta[2] ~ dnorm( 1/xsd[2] , 1/(6/xsd[2]^2) ) # 1/ variance for normal distribution
  zbeta[3] ~ dnorm( 0 , 1/9 ) # 1/ variance for normal distribution
  zbeta[4] ~ dnorm( 1.2/xsd[4] , 1/(1.3/xsd[4]^2) ) # 1/ variance for normal distribution
  zbeta[5] ~ dnorm( 1.5/xsd[5] , 1/(0.05/xsd[5]^2) ) # 1/ variance for normal distribution

  zVar ~ dgamma( 0.01 , 0.01 )
  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]

}
"
writeLines( modelString , con="TEMPmodel.txt" )


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

adaptSteps = 1900  # Number of steps to "tune" the samplers
burnInSteps = 6000
nChains = 3
thinSteps = 23  # First run for 3
numSavedSteps = 6000
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )

runJagsOut <- run.jags( method="parallel" ,
                  model="TEMPmodel.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 )
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]
Variance <- summaryInfo[14,3]
meanGamma <- as.matrix(cbind(rep(1,nrow(x)), x)) %*% as.vector(coefficients)
randomData <- rgamma(n= 231,shape=meanGamma^2/Variance, rate = meanGamma/Variance)

predicted <- data.frame(elapsed = randomData)
observed <- data.frame(elapsed = y)
predicted$type <- "Predicted"
observed$type <- "Observed"
dataPred <- rbind(predicted, observed)
ggplot(dataPred, aes(elapsed, fill = type)) + geom_density(alpha = 0.2)

predict1<- as.data.frame(xPred)
predict1<-predict1 %>%
  rename(Area = V1, Bedrooms = V2, Bathrooms = V3, CarParks=V4, PropertyType=V5)
predictprice<- summaryInfo[16:20,3]
PredictionPrice<-cbind(predictset,predictprice)
PredictionPrice <- PredictionPrice %>%rename(EstimatedPrice = predictprice)
PredictionPrice
```