TABLE OF CONTENTS

PART A

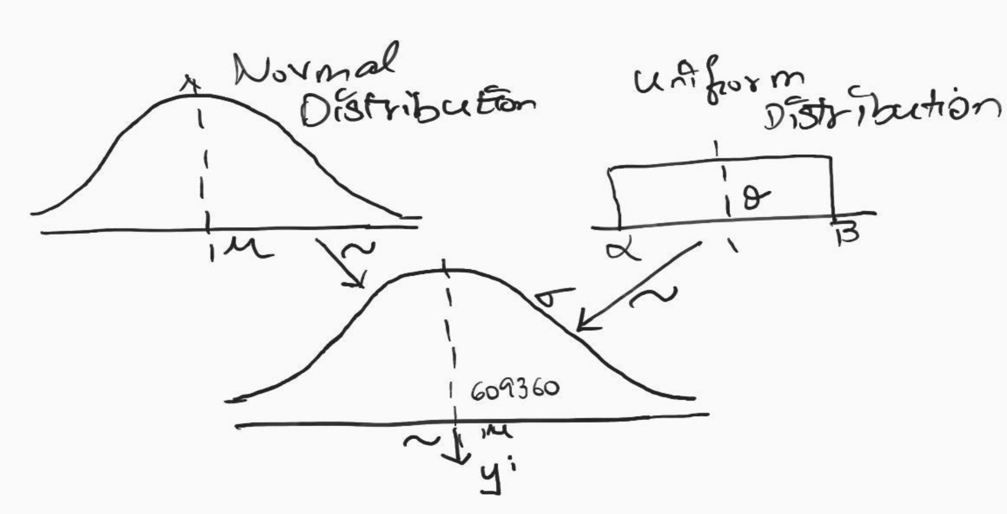
To start with, the model for JAGS is formulated in order to find the distribution of sales price and prior distribution of mu and Sigma. A model diagram can be used to serve this purpose. The model diagram is also used for creating JAGS data and blocks of the model. Coming to the dataset description, we have 10000 observations and 6 variables. The goal is to find the Bayesian estimate of the mean sale price μ and its variance σ. The dataset comprises of the following variables:

1. Salesprice: Sale price in AUD
2. Area: Land size in m2 of the sold property
3. Bedrooms: The number of bedrooms
4. Bathrooms: The number of bathrooms
5. Carparks: The number of car parks
6. Property Type: The type of the property (0: House, 1: Unit)

Let us consider the Salesprice as the dependent variable and the rest of the variables are independent variables.

1. Create a model diagram for JAGS showing the distribution of sale prices and prior distributions of μ and σ2. At this step, please do not forget to consider domains of μ and σ2!

This part involves the creation of JAGS model for all the observations in the given dataset, which are distributed normally with the respective mean and variance.



JAGS MODEL

2. Specify non-informative prior distributions for both of μ and σ^2.

Below are the steps that are incorporated in a Bayesian data analysis:

a) Using Probabilistic sampling to identify the relevant information along with the queries for research.
b) Next the descriptive model is determined with the use of mathematical forms and appropriate parameters to aid the ultimate aim of the analysis.
c) For the re-allocation of credibility, the concept of Bayesian inference the parameter values spread with the implementation of posterior distribution.
d) The procedure of checking the most extreme quality of accuracy with the data and the model close to the posterior presumptions.
e) Meaningful reasonings were gotten from the model to be utilized all the while.

3. Create JAGS **data** and **model** blocks based on the model diagram at the previous step.

Initially load the data to create a JAGS model. The dataset is fed in R for analysis using Read.csv function. The csv file is now converted into a dataframe with all the variables. So, we plan to make the data into a list format in order to combine it with R. The sum of number of the total values of the data given to JAGS is made note of. The implementation of the code is shown below:

```
VData <- read.csv("Assignment2PropertyPrices.csv")
library(readr)
mean(VData$SalePrice)
y = VData$SalePrice #The Y values are the component named Y
Ntotal = length(y) #Compute the total number of flips
dataList = list(   #Put the information into the list.
  y = y ,
  Ntotal = Ntotal
)
mean(y)
sqrt(var(y))
var(y)
```

The primary concern is that the names of the parts must be a similar variable name in the JAGS model specification.

The second step Model specification.

Here the model is determined utilizing the chart. The specification begins with the model as its demonstrated as follows.

```
modelString = "
# open quote for modelString

model {
for ( i in 1:Ntotal ) {
y[i] ~ dnorm( mu , tau ) # Use sample variance as an estimate of the population variance
}
mu ~ dnorm( 0 , 0.5*(10^-12) ) # Here set the prior variance to a value smaller than the sample variance
tau <- pow(sigma, -2)
sigma ~ dunif(52000,5*(10^7))
}
" # close quote for modelString
writeLines( modelString , con="TEMPmodel.txt" )
```

The model implementation is the only factor that differs in-between the various stages. The main criteria is that the model specified opens the meaning of the structure to JAGS and to the components of R model. Moreover, the translation of the string along with the model specification is done by JAGS processor.

4. Compile your model and create Markov chains using the compiled model.

In this part of the analysis the chains are created and initialized. The RJAGS model is made use of using the RJAGS package in R. Moreover, we specify the file name, data, number of chains and the number of steps to take for adapting.

```
# Compile the model
install.packages("coda")
library("coda")

library(R2jags)
library(rjags) # you need to load the package rjags into R here or at early stages
jagsModel = jags.model( file="TEMPmodel.txt" , # the name of the file in which the model
                        # specification is stored
                        data=dataList ,        # the list of data
                        # to let JAGS to create its own initial values for the chains,
                        # simply omit this argument entirely
                        n.chains=4,           # the number of chains to be generated
                        n.adapt=500           # the number of steps to take for adapting
                        # (or tuning) the samplers
)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 10000
   Unobserved stochastic nodes: 2
   Total graph size: 10018

Initializing model
```

The burn-in period is specified and the generation of the chains the posterior inferences are calculated. The below code explains the implementation:

```
codaSamples = coda.samples( jagsModel ,                  # previously created JAGS model object
                            variable.names=c("mu","sigma") , # specify which parameters will have
                            # their values recorded during the
                            # MCMC walk
                            n.iter=8000            # specify the number of iterations for
                            # each chain
)
```

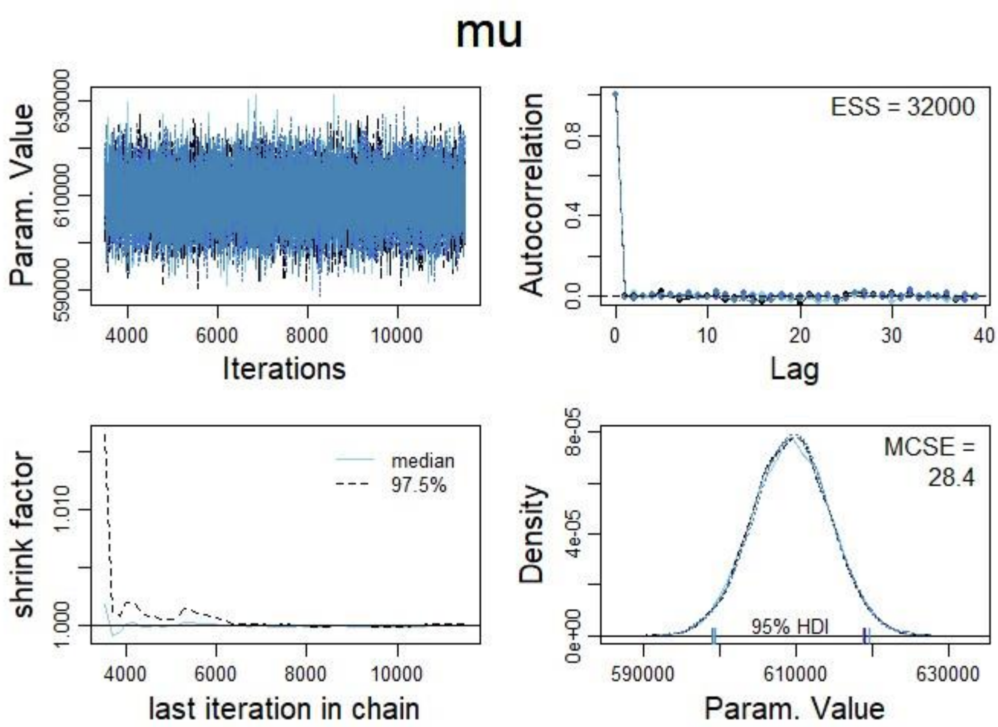
5. Assess the appropriateness of the chains using the MCMC diagnostics.

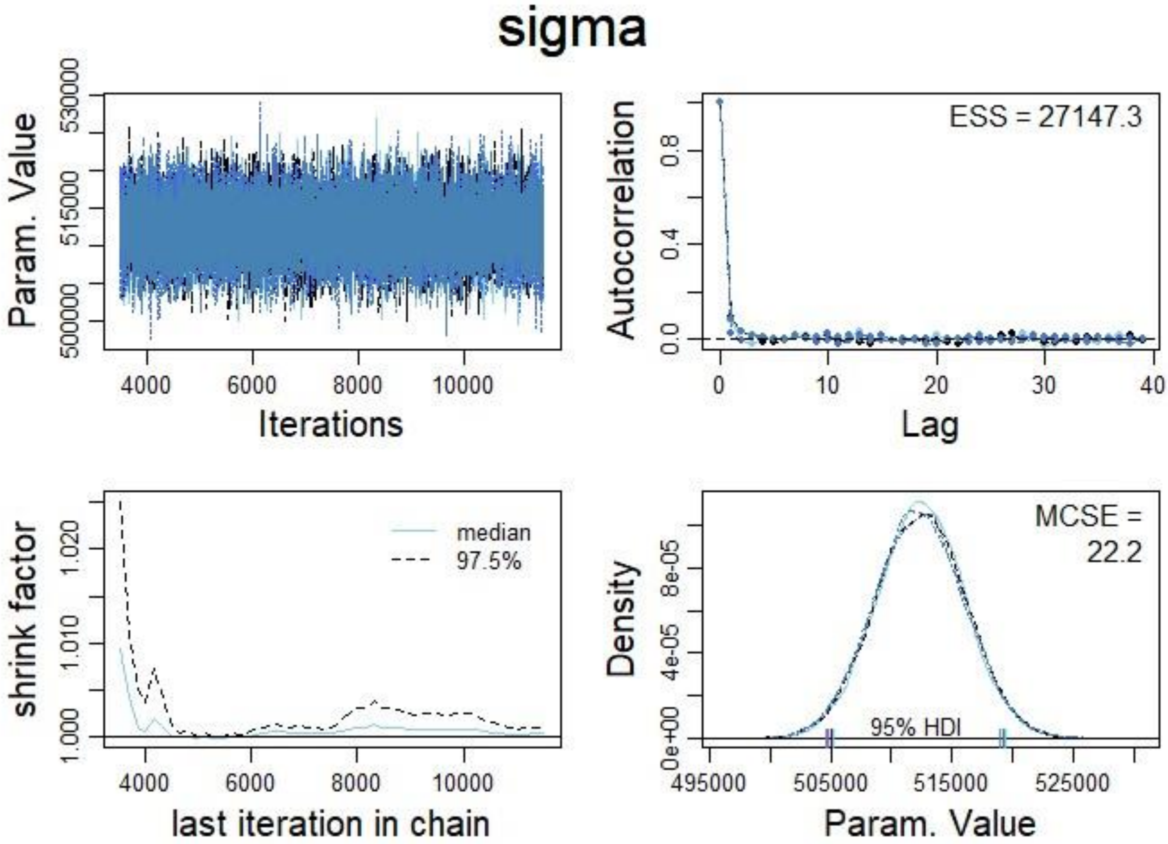
Examining the appropriateness of the Markov chains is done using the MCMC diagnostics in this step. The very important part is that the accurateness of the Markov chains is accessed properly. The exquisite functions in the CODA package help us to create awesome plots and quantify the appropriateness of the Markov chain Monte Carlo chains. The DIAG MCMC from the file given in the Module 5 is made use of for this step. In order to implement the function, we execute the mentioned code.

```
source("DBDA2E-utilities.R")
```

The Diag MCMC function usage is shown below

```
diagMCMC( codaObject=codaSamples , parName="mu" )
diagMCMC( codaObject=codaSamples , parName="sigma" )
```



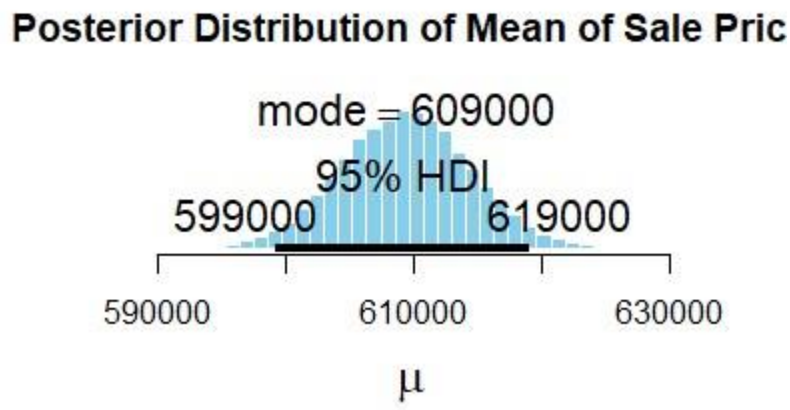


6. Display the posterior distribution of mean sales price μ and its variance σ2 and draw inferences on their Bayesian point and interval estimates.

The posterior distribution plots for the corresponding mean and Salesprice are shown below:

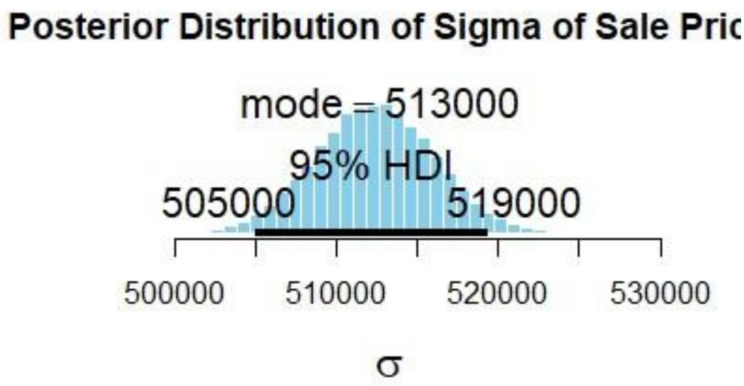
```r
# Display the posterior distribution of mu
plotPost( codaSamples[,"mu"] , # the element of the posterior samples to be plotted
          main="Posterior Distribution of Mean of Sale Price" ,          # main title
          xlab=bquote(mu)       # x-axis label
)

plotPost( codaSamples[,"sigma"] , # the element of the posterior samples to be plotted
          main="Posterior Distribution of Sigma of Sale Price" ,          # main title
          xlab=bquote(sigma)       # x-axis label
)
```

```
     ESS    mean   median      mode hdiMass   hdiLow hdiHigh compVal pGtCompVal ROPElow ROPEhigh pLtROPE
mu 32000 609361 609380.7 609466.4     0.95 599180.9  619107      NA         NA      NA       NA      NA
   pInROPE pGtROPE
mu      NA      NA
```



For Sigma

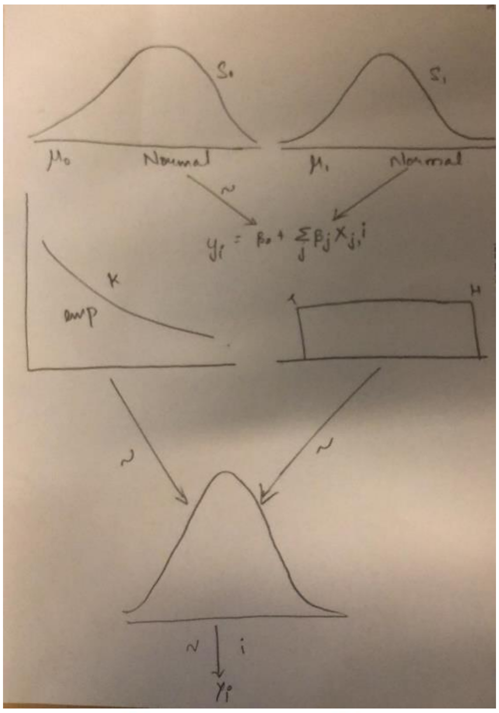
```
         ESS   mean   median     mode hdiMass   hdiLow  hdiHigh compVal pGtCompVal ROPElow ROPEhigh pLtROPE
sigma 27226.3 512379 512385.7 512710.9    0.95 505058.2 519374.5      NA         NA      NA       NA      NA
      pInROPE pGtROPE
sigma      NA      NA
```

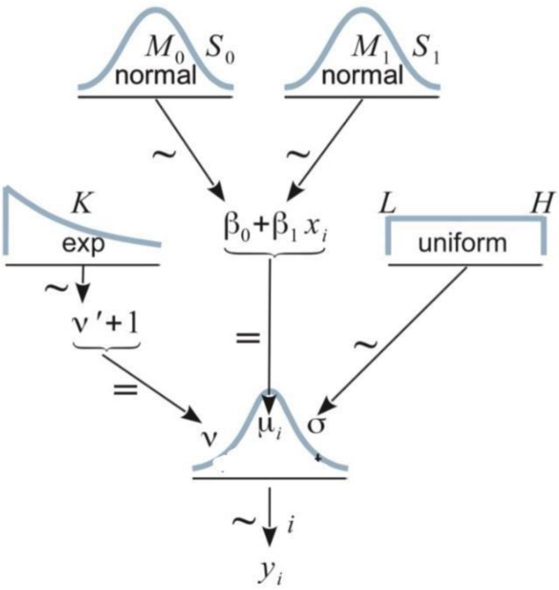


PART-B

- **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.
- **Carparks**: 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.

1. Create a JAGS model diagram showing the multiple linear regression setting in this problem





JAGS MODEL DIAGRAM

1. Specify the prior distributions reflecting the expert information for each predictor.

The prior distribution is depicted below**:**

```
# # Specify the values of indepdenent variable for prediction
# xPred[1] <- 15
# xPred[2] <- 75

# Specify the priors for original beta parameters
# Prior locations to reflect the expert information
mu0 <- ym # Set to overall mean a priori based on the interpretation of constant term in regression
mu[1] <- 0.111 # area
mu[2] <- 0.4 # BEDROOMS
mu[3] <- 0.04 # Bathrooms
mu[4] <- 0.22# Carparks
mu[5] <- 0.49# PropertyType


# Prior variances to reflect the expert information
Var0 <- 350# Set simply to 1
Var[1] <- 0.14 # Area
Var[2] <- 110 # Bedrooms
Var[3] <- 95# Bathrooms
Var[4] <- 0.18 # CARPARKS
Var[5] <- 0.19 # Property Type
```

2. Create JAGS **data** and **model** blocks based on the model diagram and prior distributions at the previous steps:

The code mentioned below is to create a JAGS data and model.

```
  # Compute corresponding prior means and variances for the standardised parameters
  muZ[1:Nx] <-  mu[1:Nx] * xsd[1:Nx] / ysd

  muZ0 <- (mu0 + sum( mu[1:Nx] * xm[1:Nx] / xsd[1:Nx] )*ysd - ym) / ysd

  # Compute corresponding prior variances and variances for the standardised parameters
  VarZ[1:Nx] <- Var[1:Nx] * ( xsd[1:Nx]/ ysd )^2
  VarZ0 <- Var0 / (ysd^2)

}
# Specify the model for standardized data:
model {
  for ( i in 1:Ntotal ) {
    zy[i] ~ dt( zbeta0 + sum( zbeta[1:Nx] * zx[i,1:Nx] ) , 1/zsigma^2 , nu )
  }

  # Priors vague on standardized scale:
  zbeta0 ~ dnorm( muZ0 , 1/VarZ0 )
  for ( j in 1:Nx ) {
    zbeta[j] ~ dnorm( muZ[j] , 1/VarZ[j] )
  }
  zsigma ~ dgamma(0.01,0.01)#dunif( 1.0E-5 , 1.0E+1 )
  nu ~ dexp(1/30.0)

  # Transform to original scale:
  beta[1:Nx] <- ( zbeta[1:Nx] / xsd[1:Nx] )*ysd
  beta0 <- zbeta0*ysd  + ym - sum( zbeta[1:Nx] * xm[1:Nx] / xsd[1:Nx] )*ysd
  sigma <- zsigma*ysd

  # Compute predictions at every step of the MCMC
  pred <- beta0 + beta[1] * xPred[1] + beta[2] * xPred[2] + beta[3] * xPred[3] + beta[4] * xPred[4]
        + beta[5] * xPred[5]

}
```

3. Compile your model and create Markov chains using the compiled model.

```
  # Compute predictions at every step of the MCMC
  pred <- beta0 + beta[1] * xPred[1] + beta[2] * xPred[2] + beta[3] * xPred[3] + beta[4] * xPred[4]
          + beta[5] * xPred[5]

}
" # close quote for modelString
# Write out modelString to a text file
writeLines( modelString , con="TEMPmodel.txt" )
#------------------------------------------------------------------------------
# INTIALIZE THE CHAINS.
# Let JAGS do it...

# Must standardize data first...
#   lmInfo = lm( zy ~ zx )
#   initsList = list(
#     beta0 = lmInfo$coef[1] ,
#     beta = lmInfo$coef[-1] ,
#     sigma = sqrt(mean(lmInfo$resid^2)) ,
#     nu = 5
#   )


#------------------------------------------------------------------------------
# RUN THE CHAINS
parameters = c( "beta0" ,  "beta" ,  "sigma",
                "zbeta0" , "zbeta" , "zsigma", "nu" , "pred" )
```

NEXT SECTION: PREDICTING THE SALES PRICES

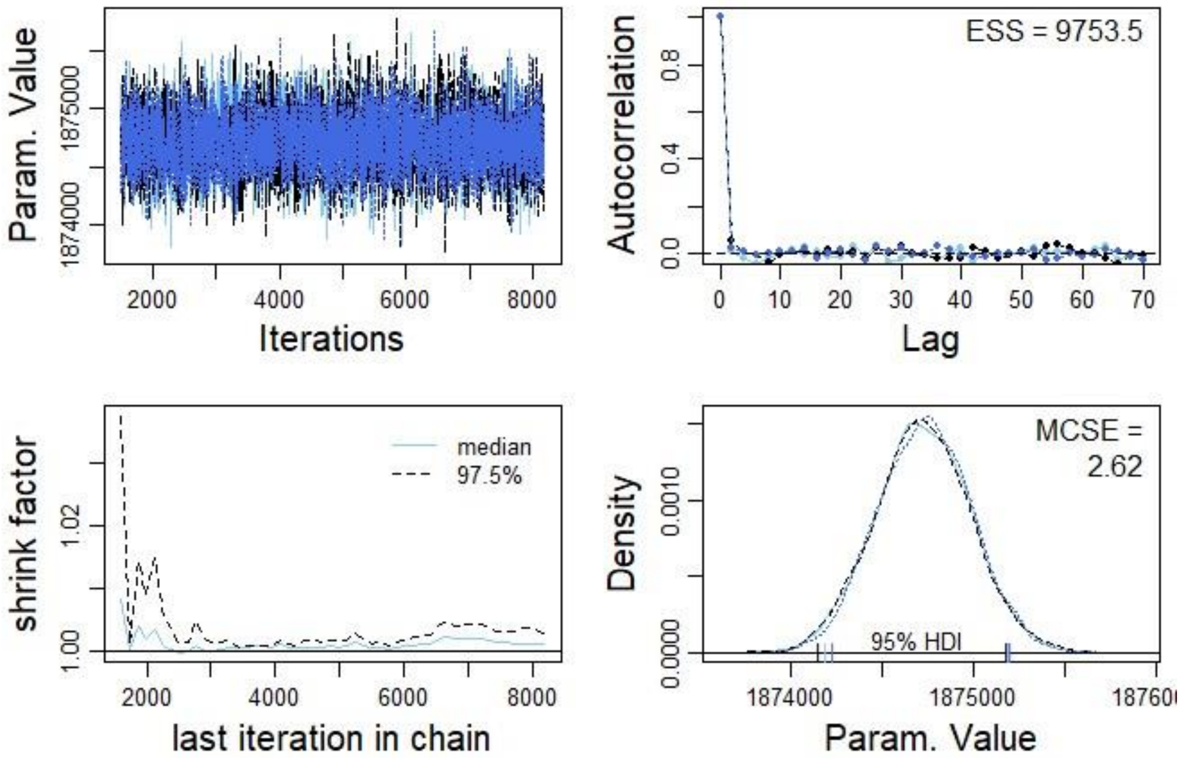
Property No	Area	Bedrooms	Bathrooms	CarParks	PropertyType
1	600	2	2	1	Unit
2	800	3	1	2	House
3	1500	2	1	1	House
4	2500	5	4	4	House
5	250	3	2	1	Unit

Case 1: The condition for Prediction of the Salesprice

Area =600, bedrooms =2, bathrooms =2, car parks =1 and property type = unit.

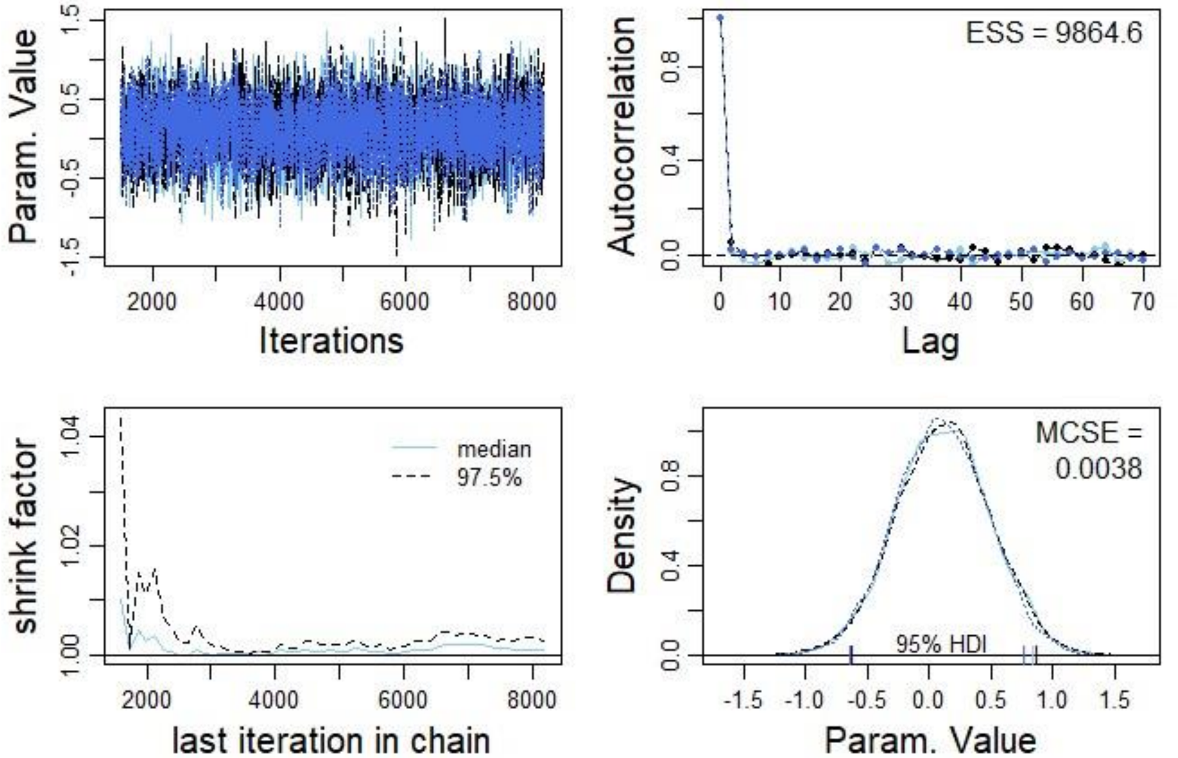
```
CORRELATION MATRIX OF PREDICTORS:
               Area  Bedrooms Bathrooms CarParks PropertyType
Area          1.000   -0.345    -0.104   -0.152        0.286
Bedrooms     -0.345    1.000     0.498    0.500       -0.597
Bathrooms    -0.104    0.498     1.000    0.324       -0.271
CarParks     -0.152    0.500     0.324    1.000       -0.411
PropertyType  0.286   -0.597    -0.271   -0.411        1.000
```

beta0

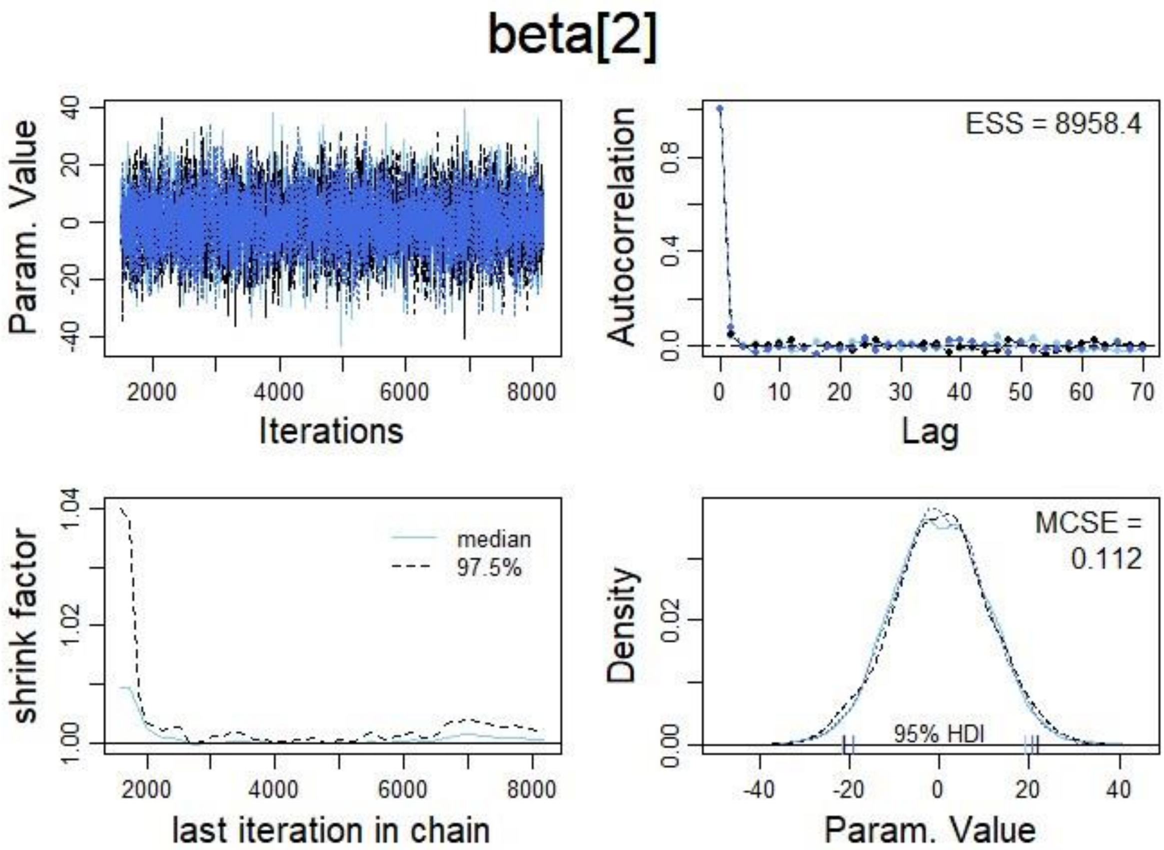


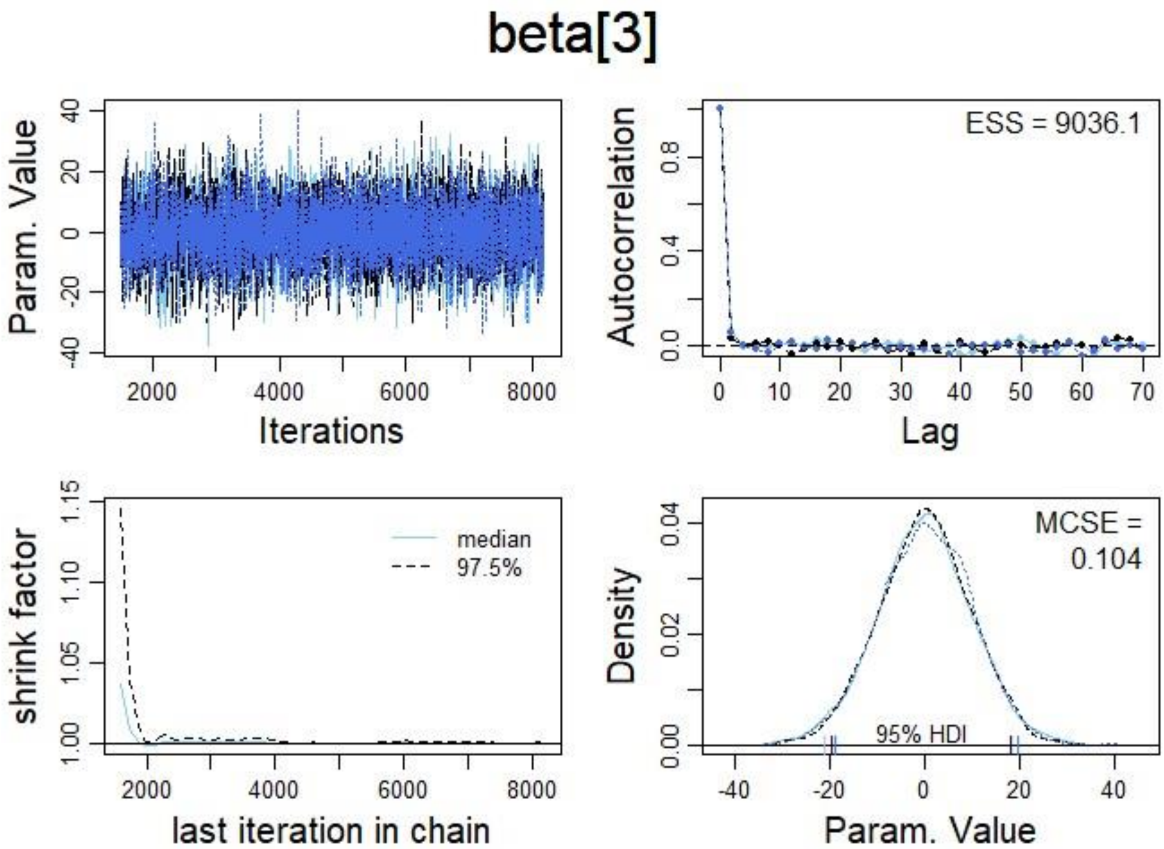
Param. Value
Iterations
Autocorrelation
ESS = 9753.5
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
2.62
95% HDI
Param. Value

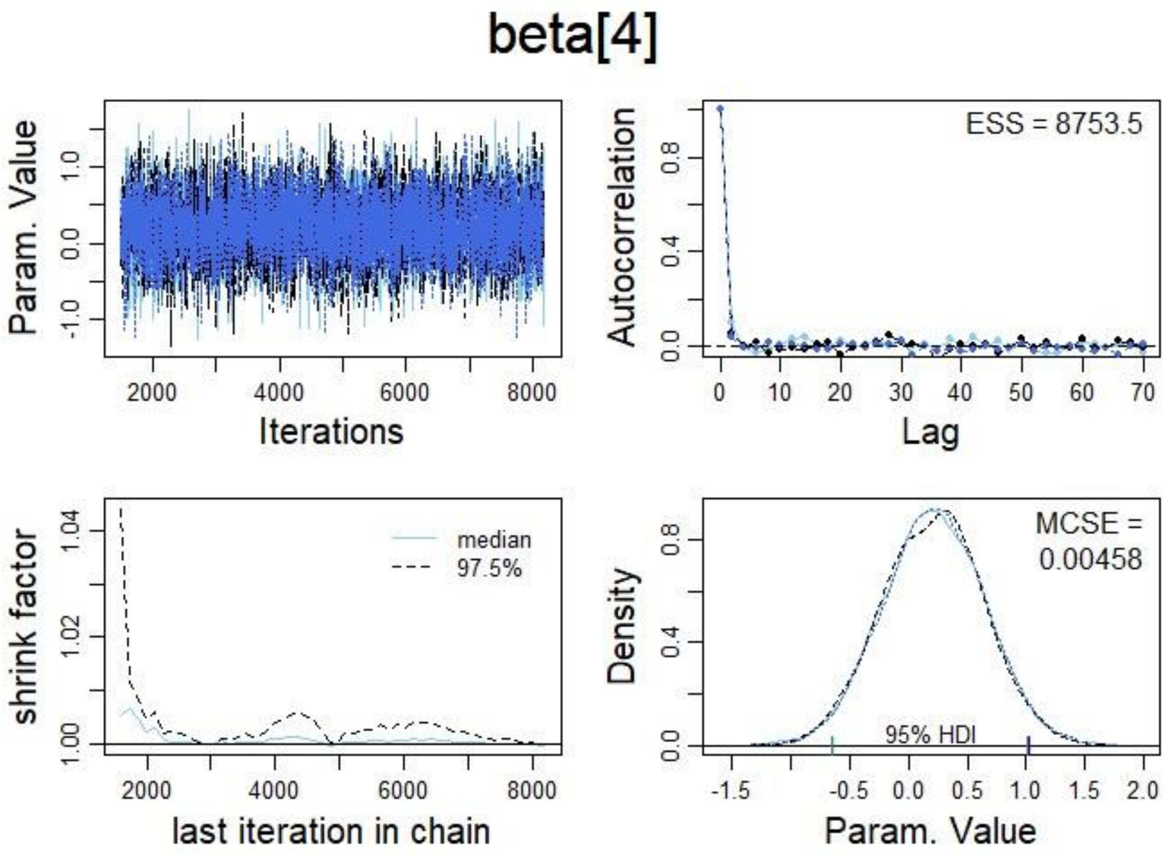
beta[1]

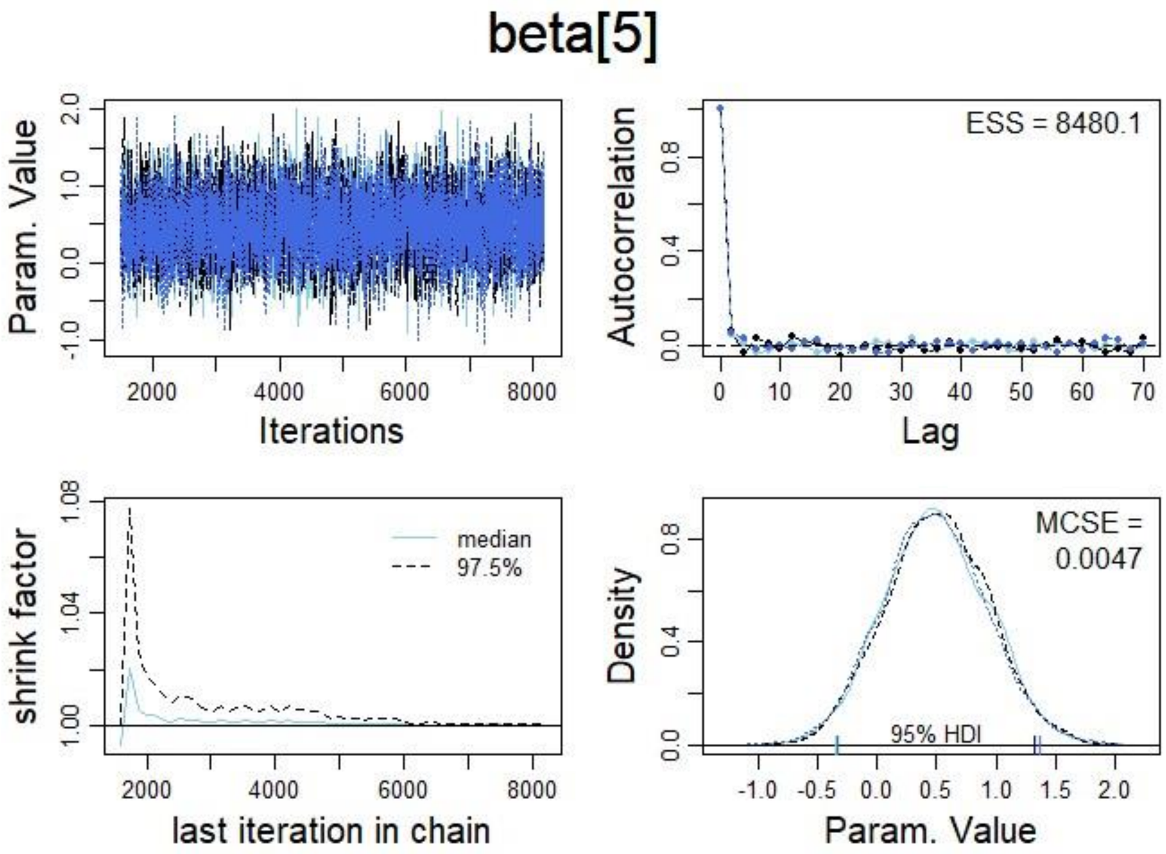


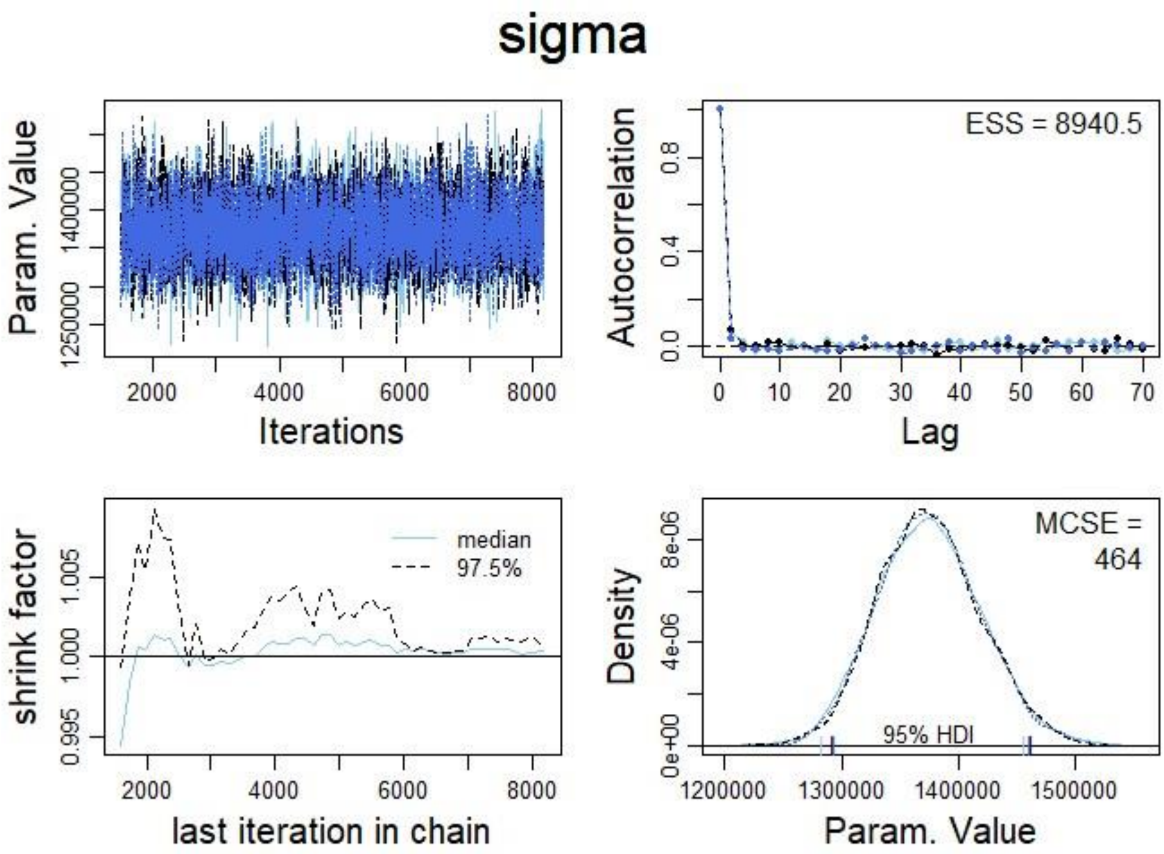
Param. Value
Iterations
Autocorrelation
ESS = 9864.6
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.0038
95% HDI
Param. Value

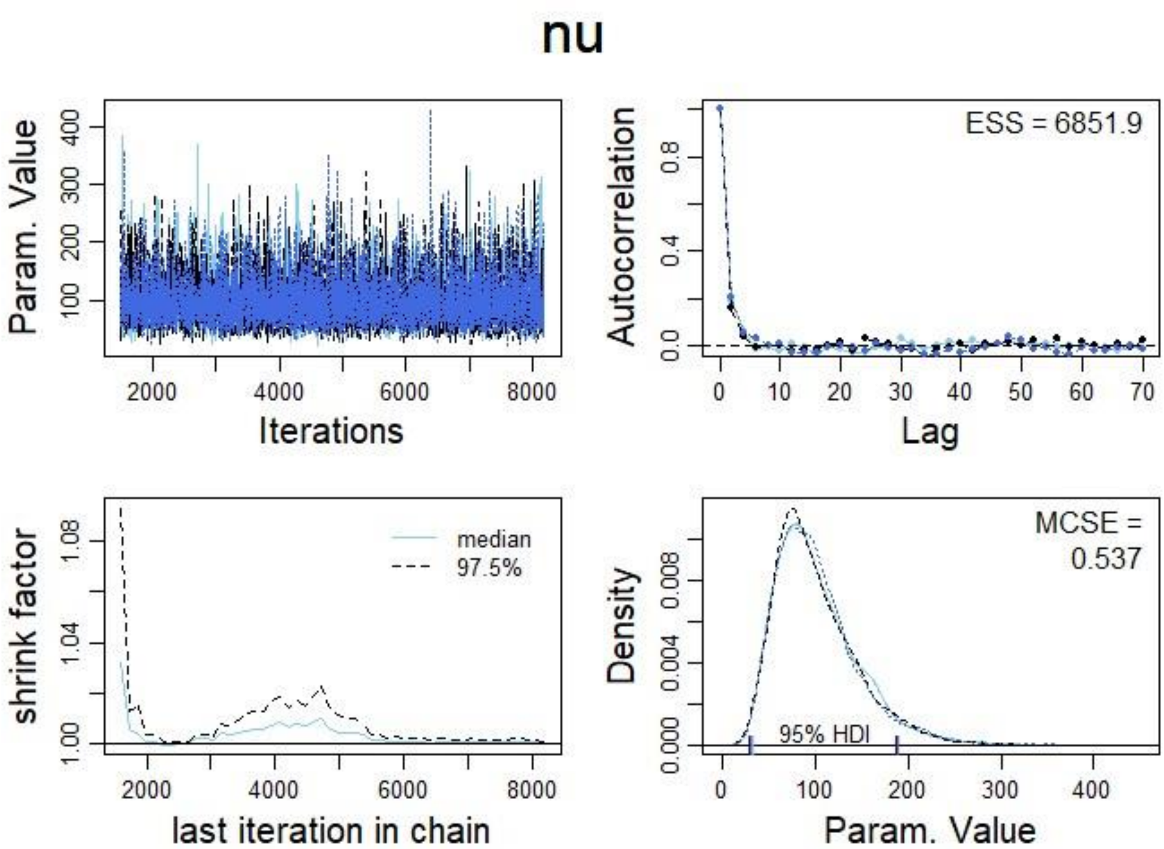

beta[2]
Param. Value
Iterations
Autocorrelation
ESS = 8958.4
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.112
95% HDI
Param. Value

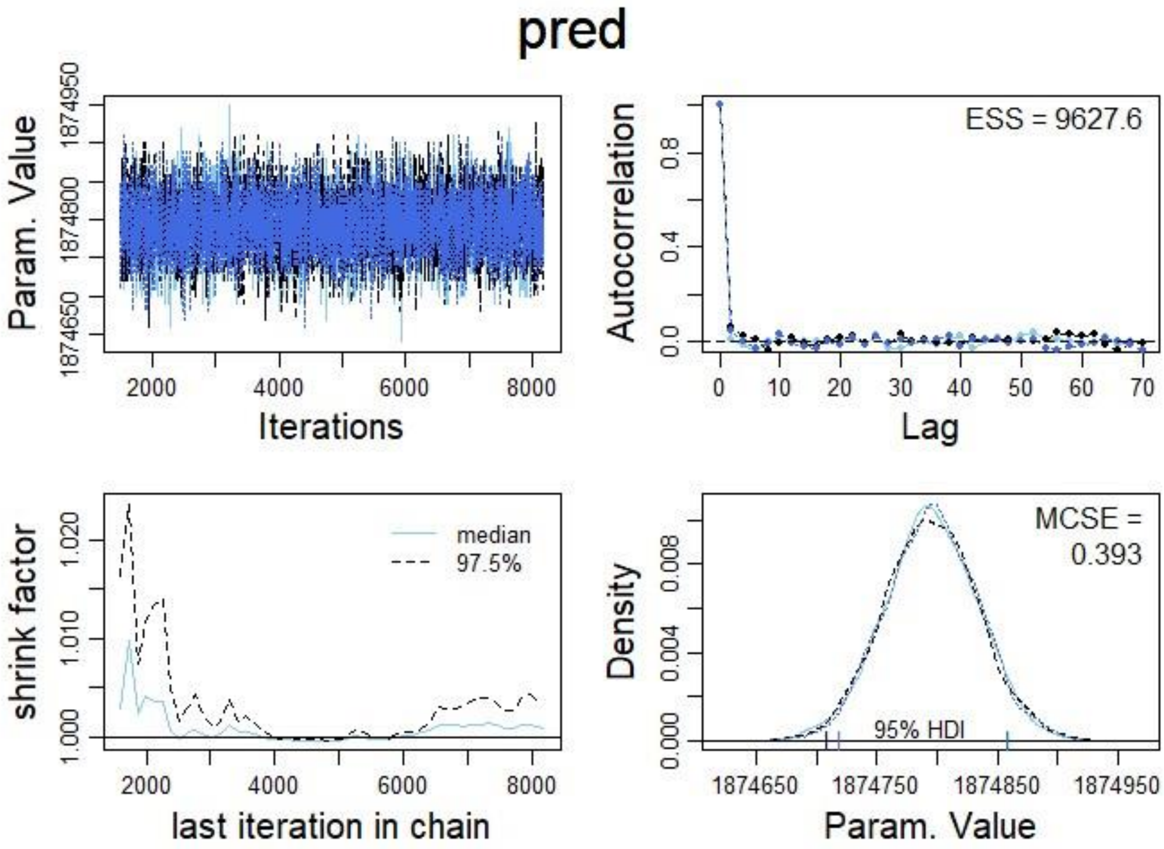

beta[3]
Param. Value
Iterations
Autocorrelation
ESS = 9036.1
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.104
95% HDI
Param. Value

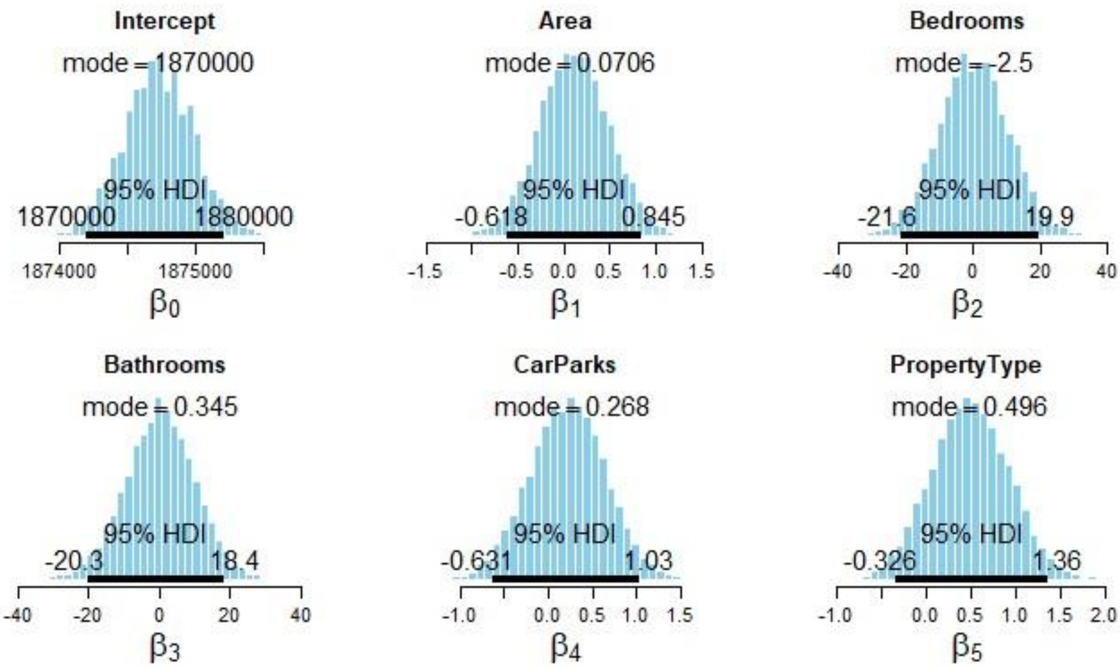

beta[4]
Param. Value
Iterations
Autocorrelation
ESS = 8753.5
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.00458
95% HDI
Param. Value


beta[5]
Param. Value
Iterations
Autocorrelation
ESS = 8480.1
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.0047
95% HDI
Param. Value


sigma
Param. Value
Iterations
Autocorrelation
Lag
ESS = 8940.5
shrink factor
last iteration in chain
median
97.5%
Density
Param. Value
MCSE = 464
95% HDI


nu
Param. Value
Iterations
Autocorrelation
Lag
ESS = 6851.9
shrink factor
last iteration in chain
median
97.5%
Density
Param. Value
MCSE = 0.537
95% HDI



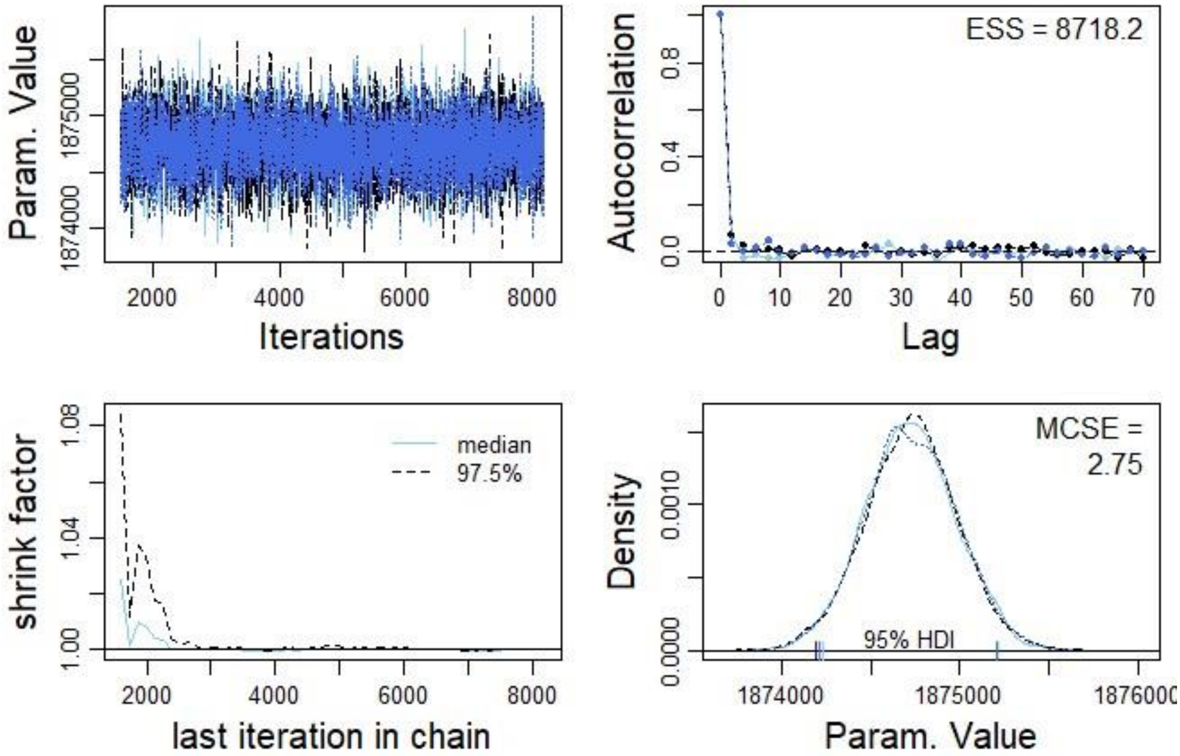


PREDICTION 2

Case 2: The condition for Prediction of the Salesprice

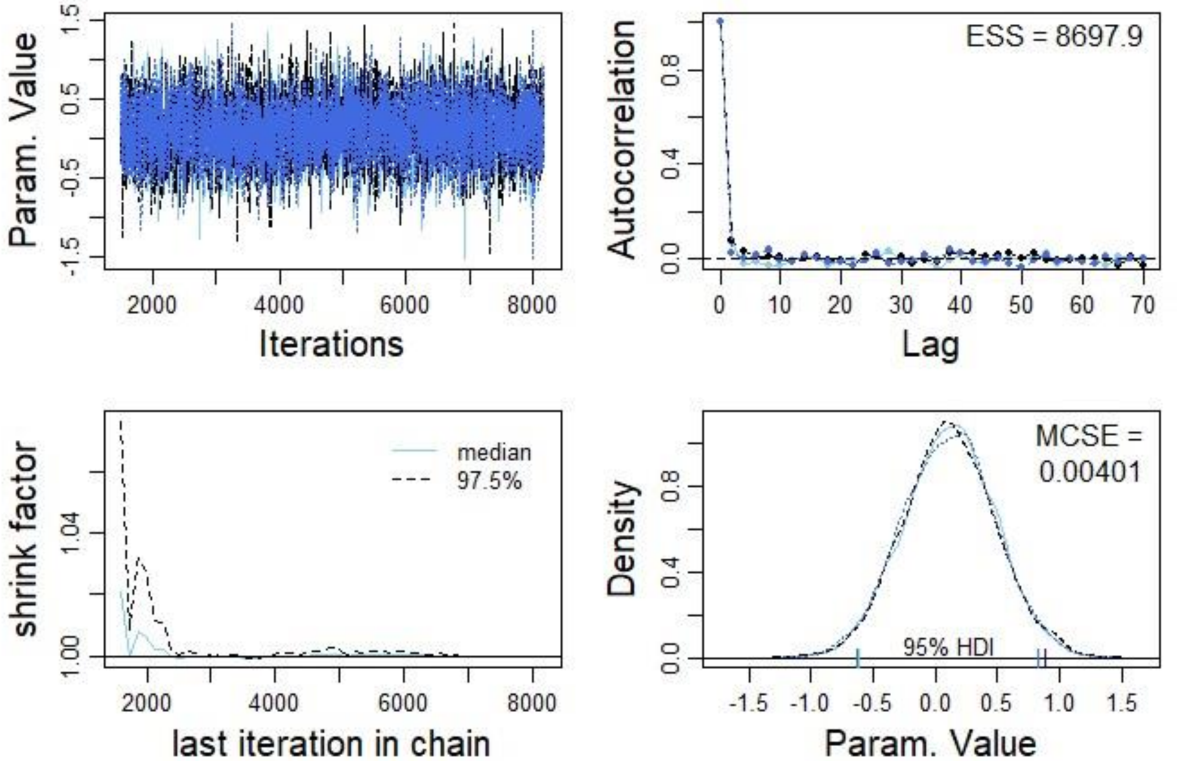
Area =800, bedrooms =3, bathrooms =1, car parks =2 and property type = house.

beta0

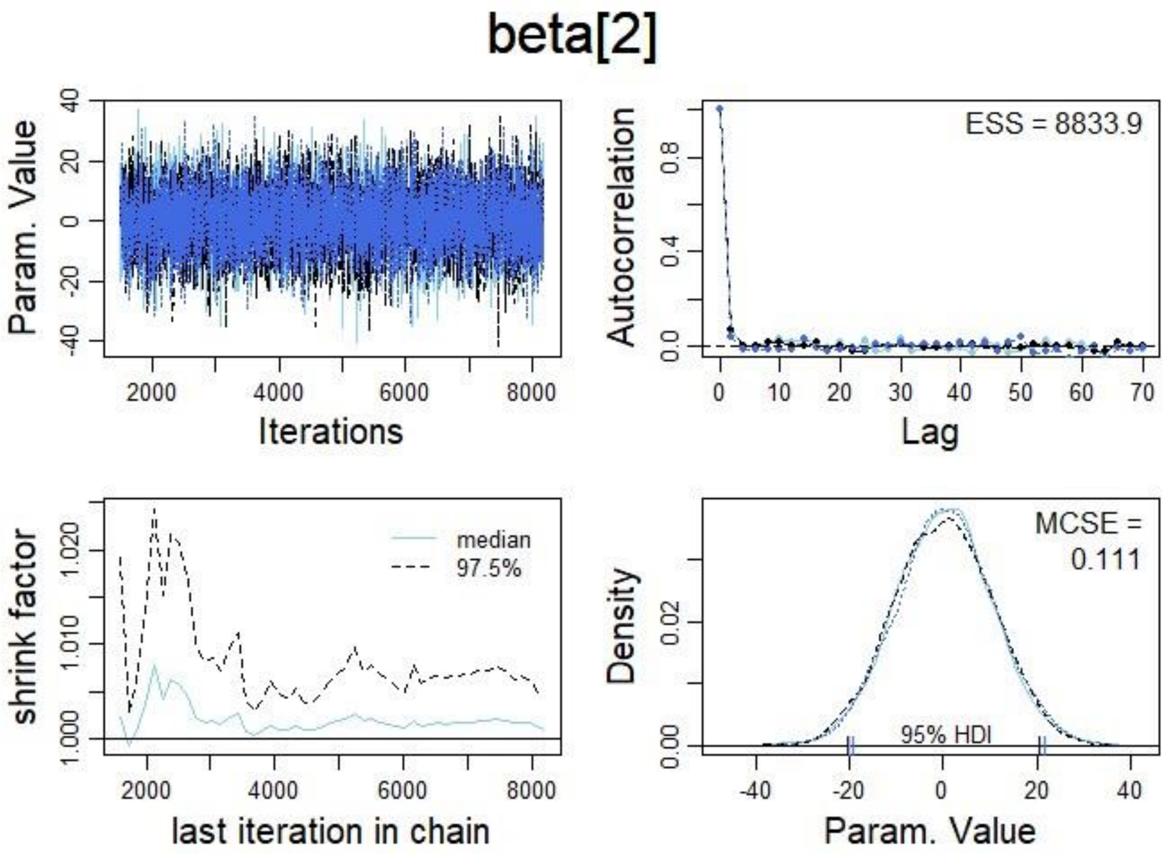


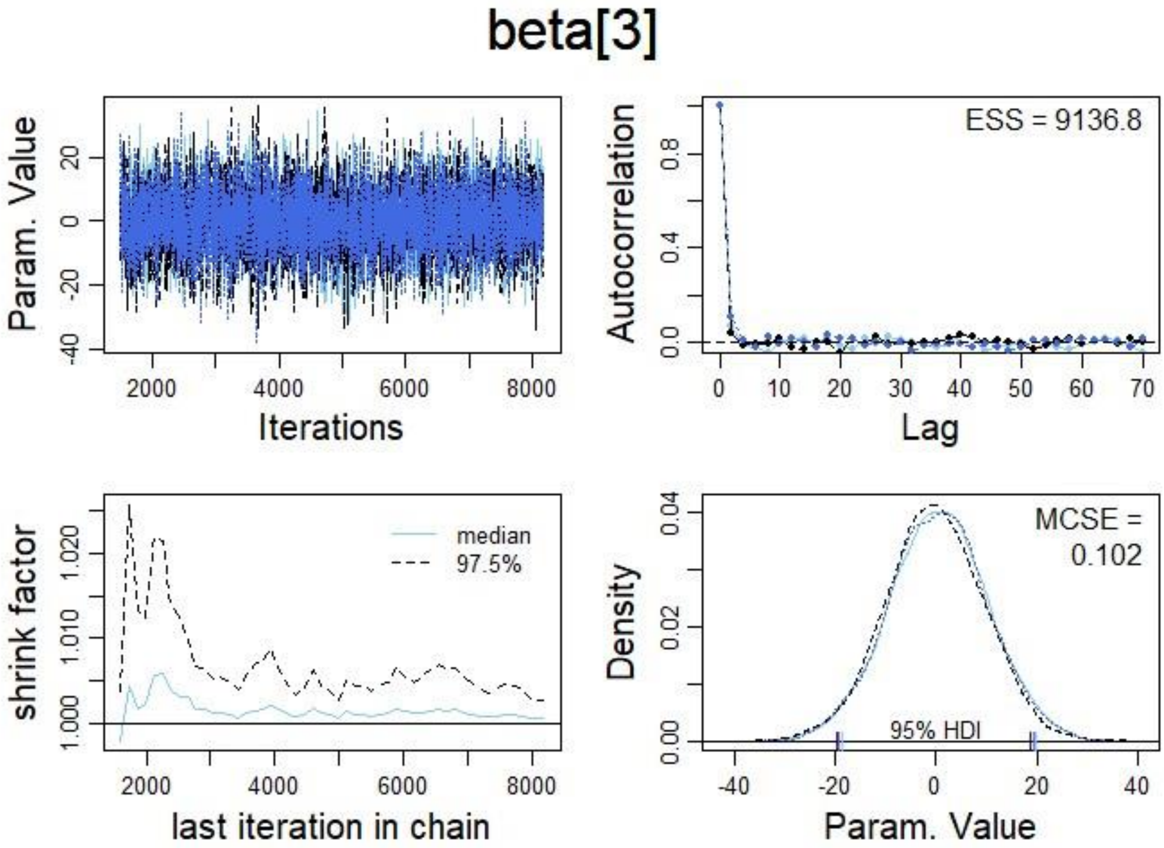
Param. Value
1874000
1875000
2000
4000
6000
8000
Iterations
Autocorrelation
ESS = 8718.2
0.0
0.4
0.8
0
10
20
30
40
50
60
70
Lag
shrink factor
1.00
1.04
1.08
median
97.5%
last iteration in chain
Density
0.0000
0.0010
MCSE =
2.75
95% HDI
1874000
1875000
Param. Value

beta[1]

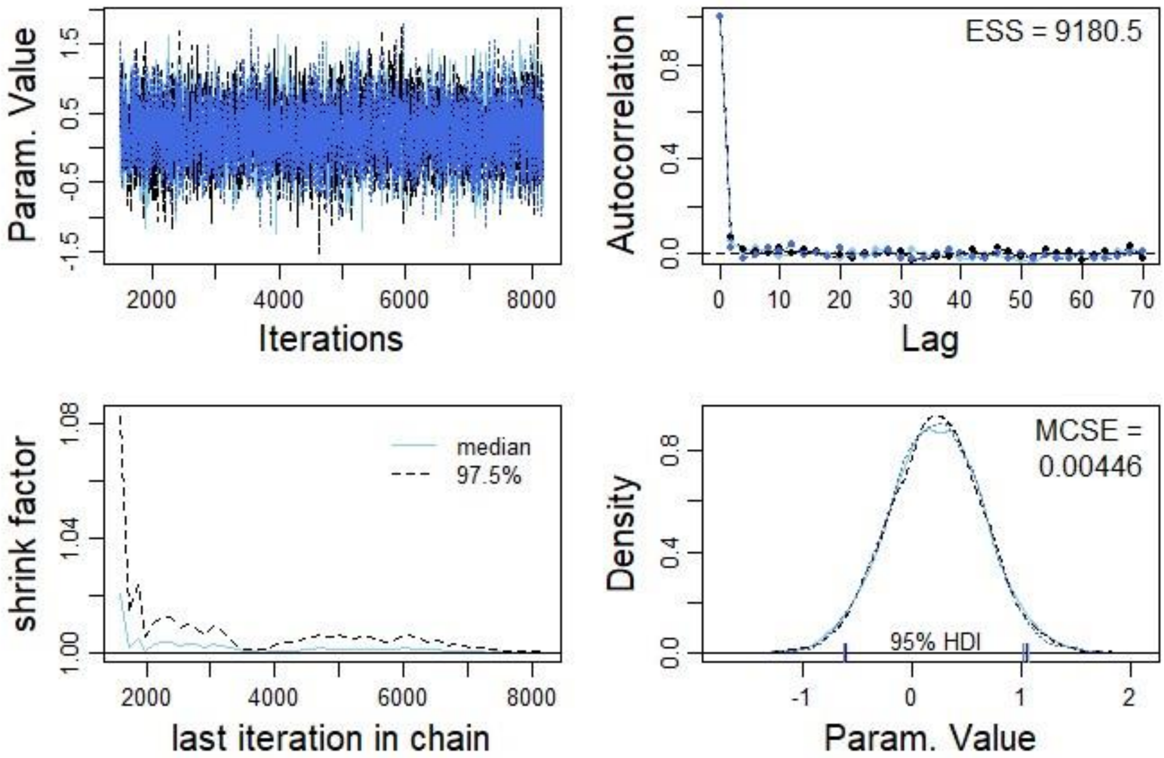


Param. Value
-1.5
-0.5
0.5
1.5
2000
4000
6000
8000
Iterations
Autocorrelation
ESS = 8697.9
0.0
0.4
0.8
0
10
20
30
40
50
60
70
Lag
shrink factor
1.00
1.04
median
97.5%
last iteration in chain
Density
0.0
0.4
0.8
MCSE =
0.00401
95% HDI
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Param. Value

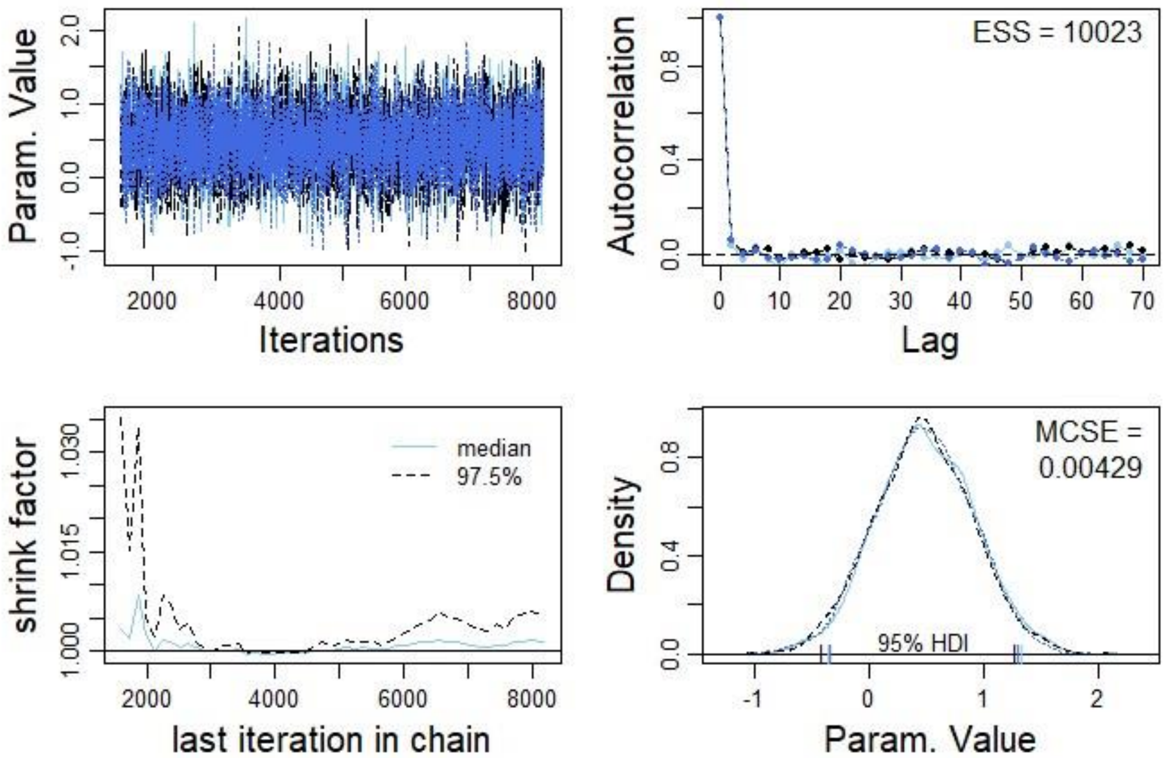

beta[2]
Param. Value
Iterations
Autocorrelation
ESS = 8833.9
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.111
95% HDI
Param. Value

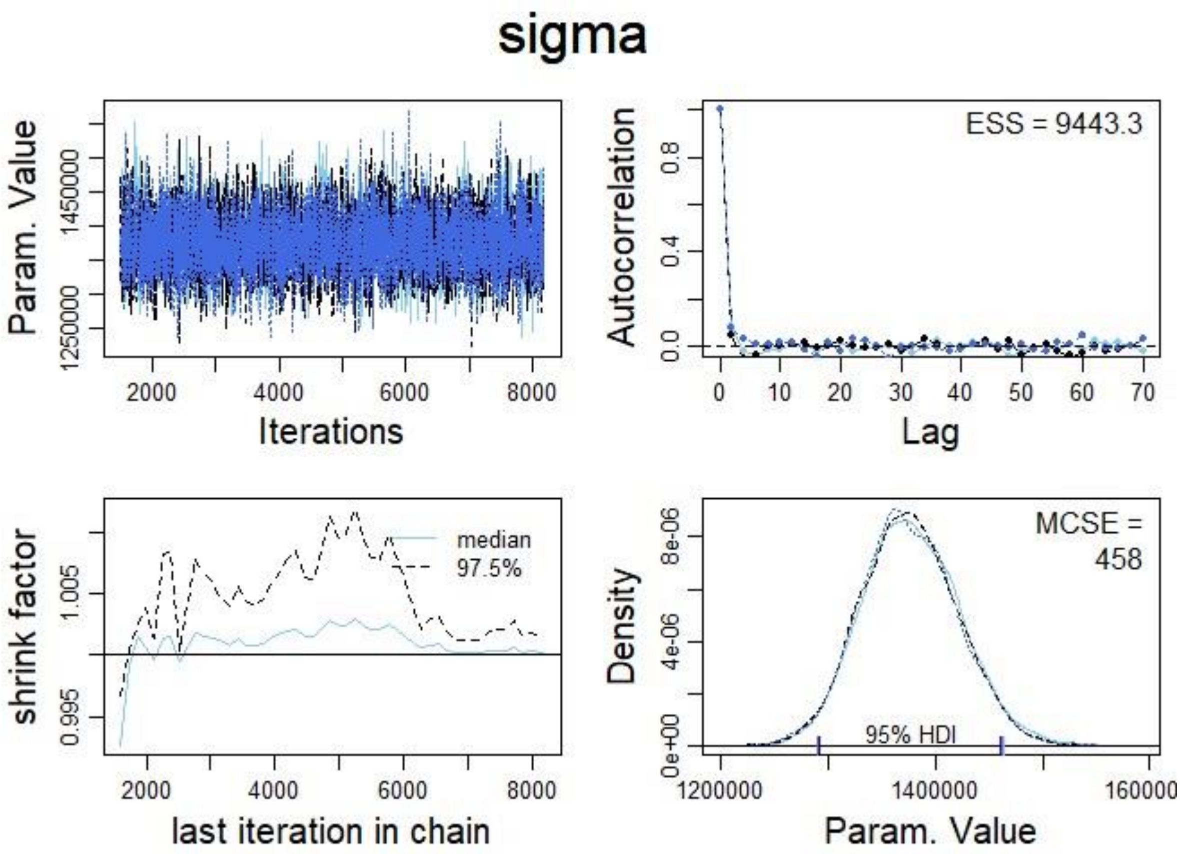

beta[3]
Param. Value
Iterations
Autocorrelation
ESS = 9136.8
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.102
95% HDI
Param. Value

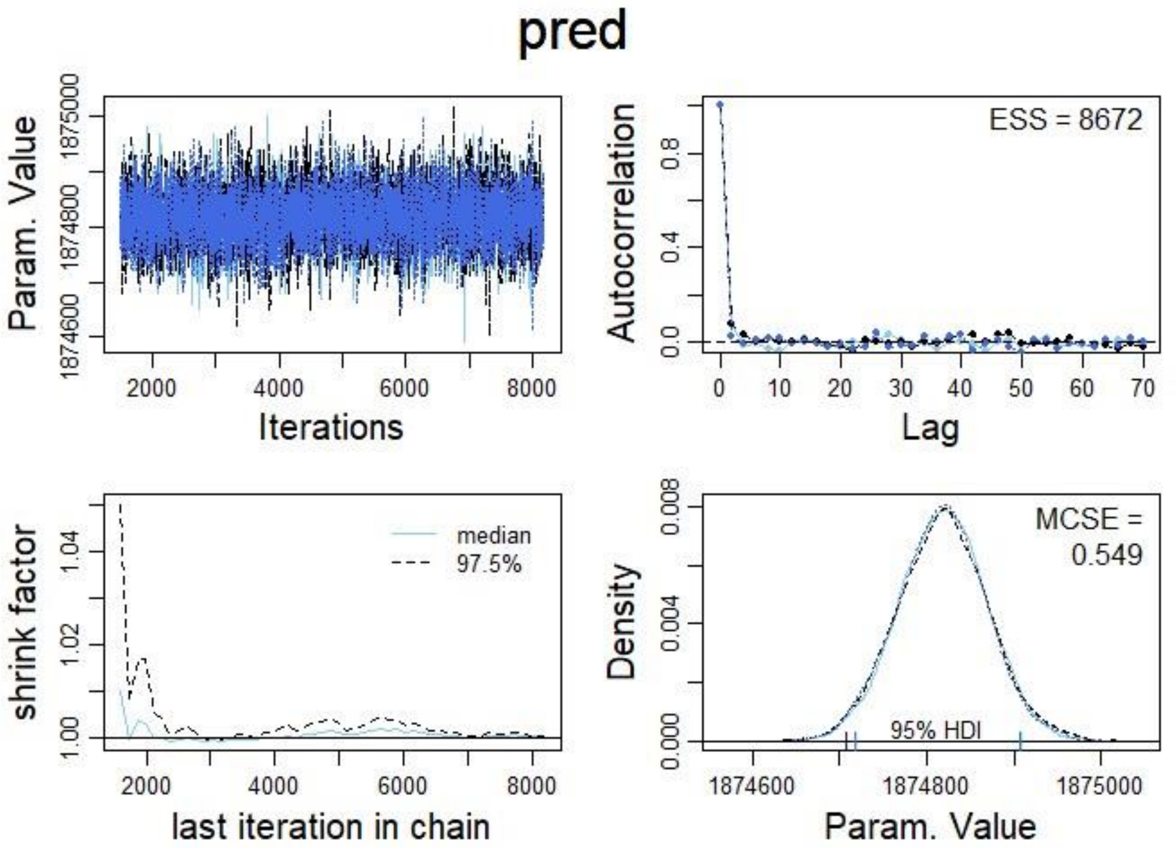
beta[4]

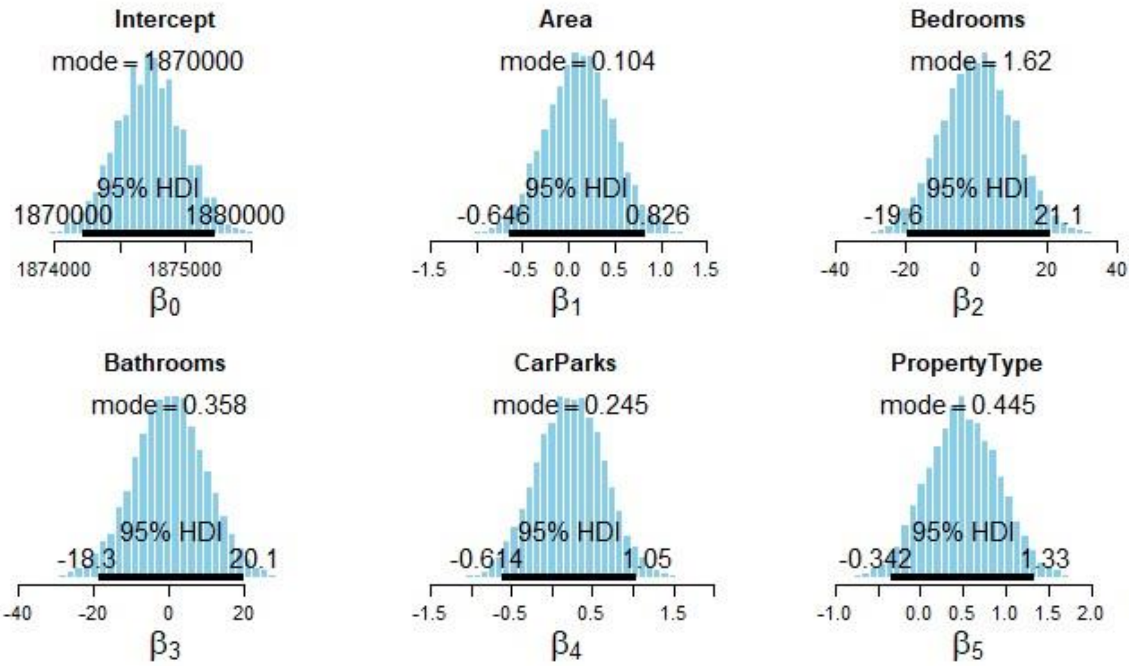

Param. Value
Iterations
ESS = 9180.5
Autocorrelation
Lag
shrink factor
median
97.5%
last iteration in chain
MCSE =
0.00446
Density
95% HDI
Param. Value

beta[5]


Param. Value
Iterations
ESS = 10023
Autocorrelation
Lag
shrink factor
median
97.5%
last iteration in chain
MCSE =
0.00429
Density
95% HDI
Param. Value


sigma
Param. Value
Iterations
Autocorrelation
Lag
ESS = 9443.3
shrink factor
last iteration in chain
median
97.5%
Density
MCSE =
458
95% HDI
Param. Value

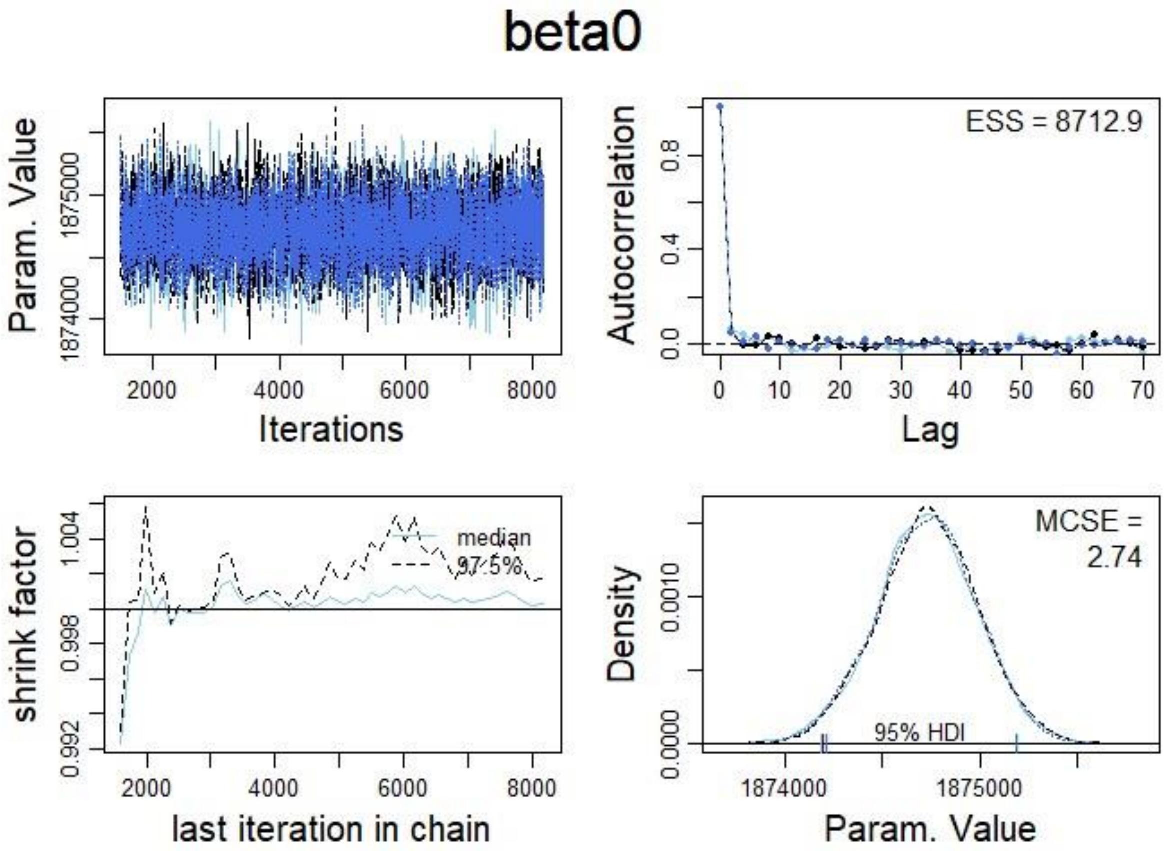

pred
Param. Value
Iterations
Autocorrelation
Lag
ESS = 8672
shrink factor
last iteration in chain
median
97.5%
Density
MCSE =
0.549
95% HDI
Param. Value

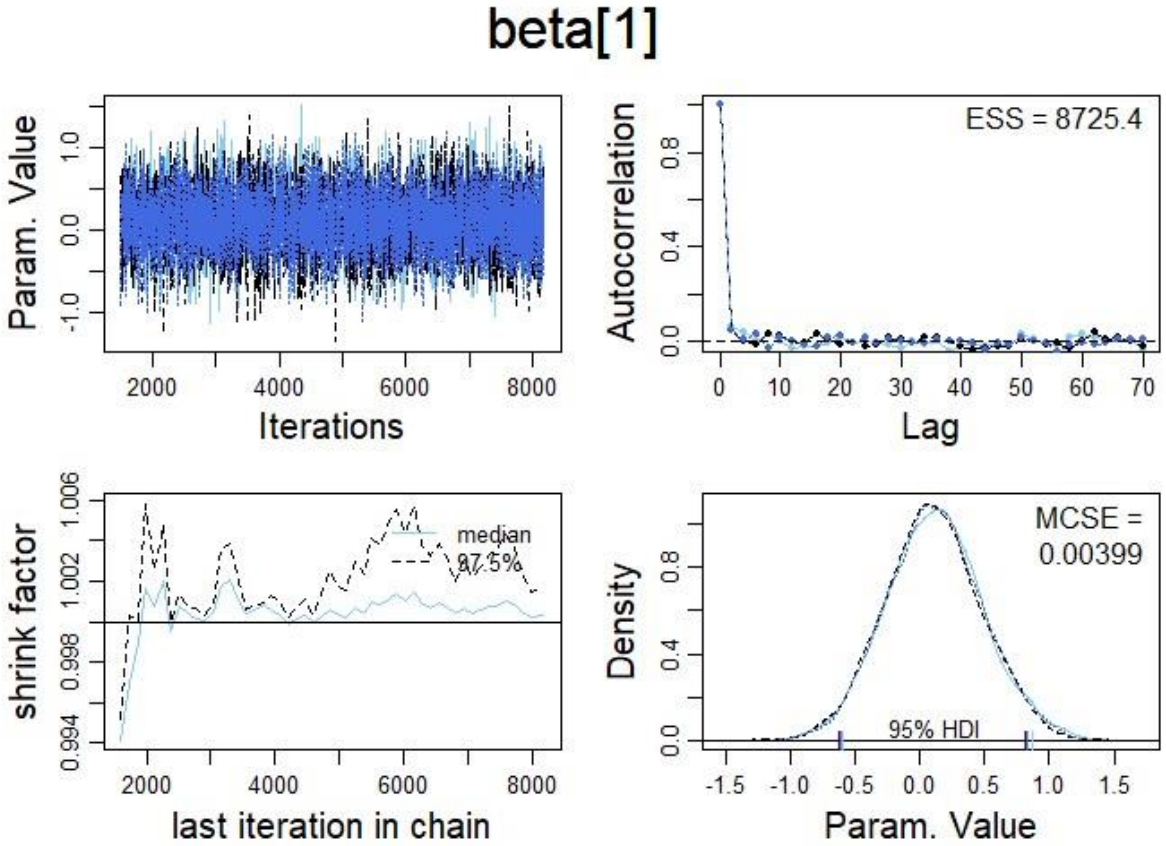


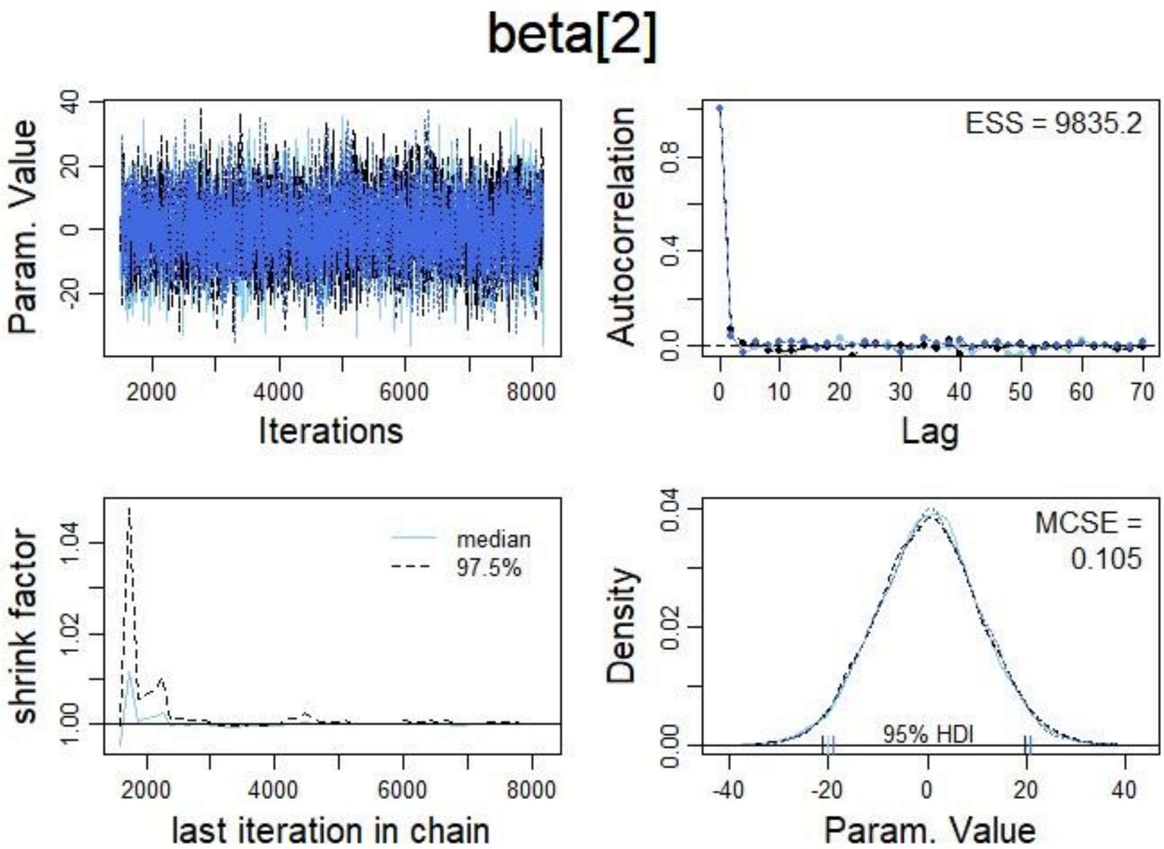
PREDICTION 3

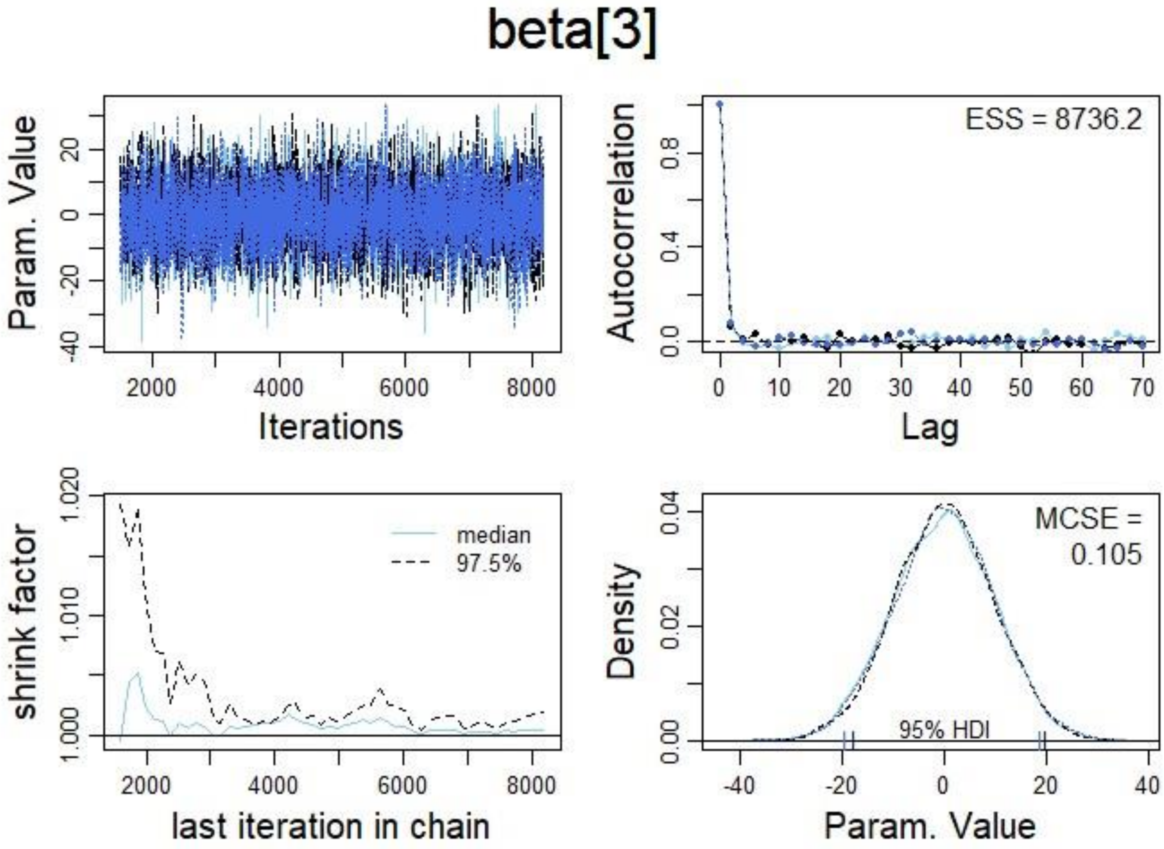
Case 3: The condition for Prediction of the Salesprice

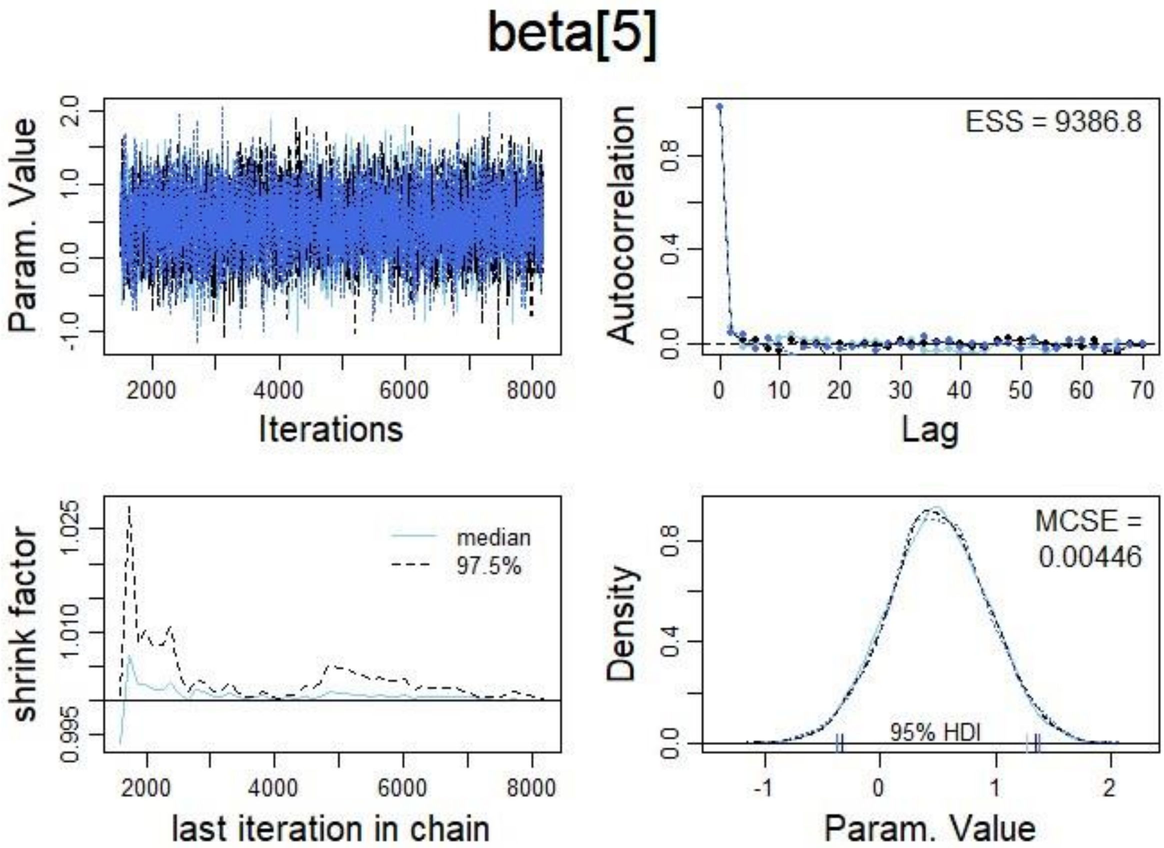
Area =1500, bedrooms =2, bathrooms =1, car parks =1 and property type = house.



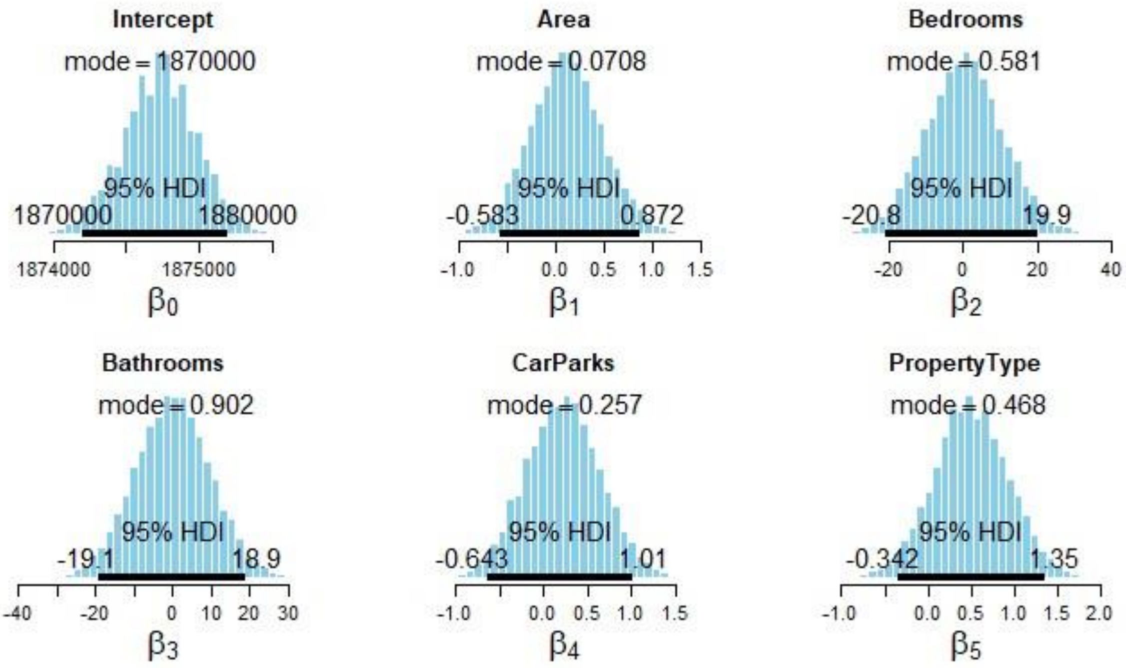

beta[1]
Param. Value
Iterations
Autocorrelation
ESS = 8725.4
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.00399
95% HDI
Param. Value


beta[2]
Param. Value
Iterations
Autocorrelation
ESS = 9835.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.105
95% HDI
Param. Value


beta[3]
Param. Value
Iterations
Autocorrelation
Lag
ESS = 8736.2
shrink factor
last iteration in chain
median
97.5%
Density
Param. Value
MCSE =
0.105
95% HDI


beta[5]
Param. Value
Iterations
Autocorrelation
Lag
ESS = 9386.8
shrink factor
last iteration in chain
median
97.5%
Density
Param. Value
MCSE =
0.00446
95% HDI

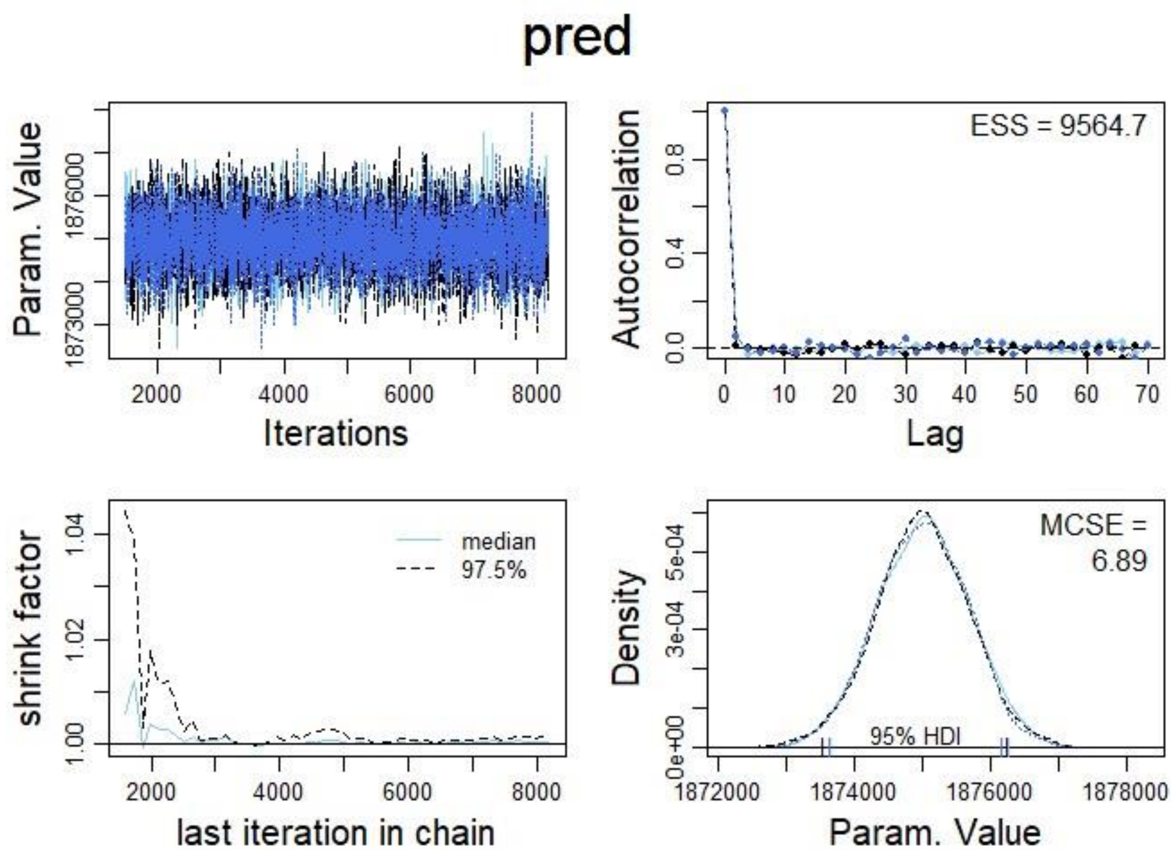
Summary:



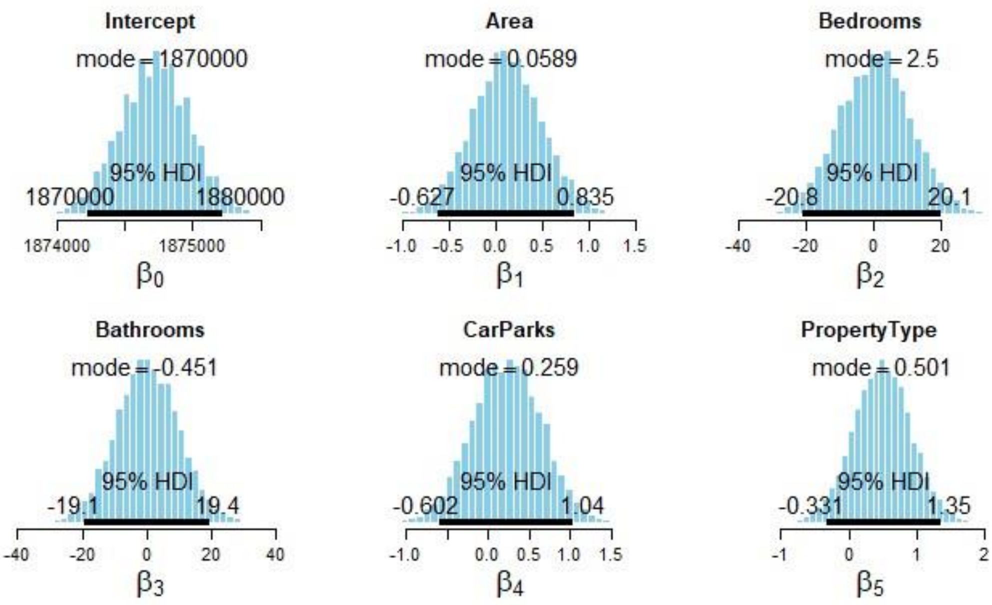
PREDICTION 4

Case 4: The condition for Prediction of the Salesprice

Area =2500, bedrooms =5, bathrooms =4, car parks =4 and property type = house.



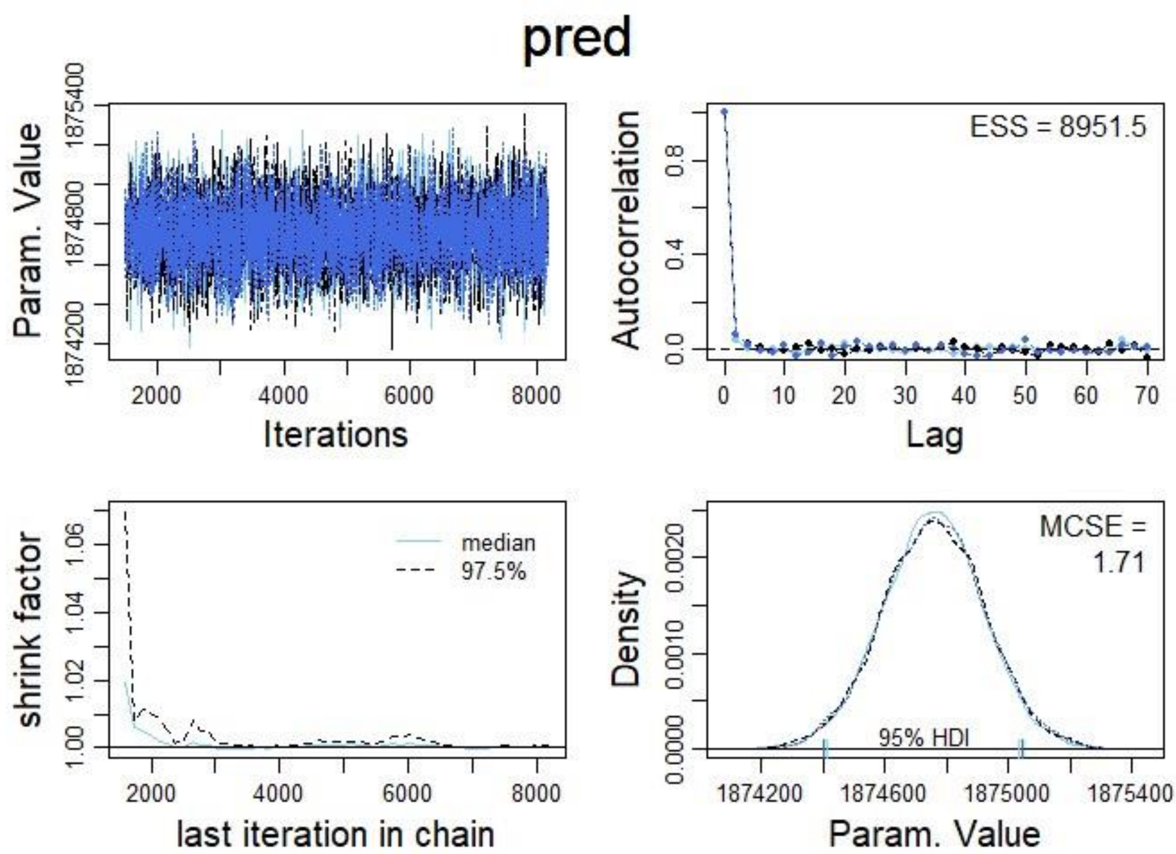
Summary:



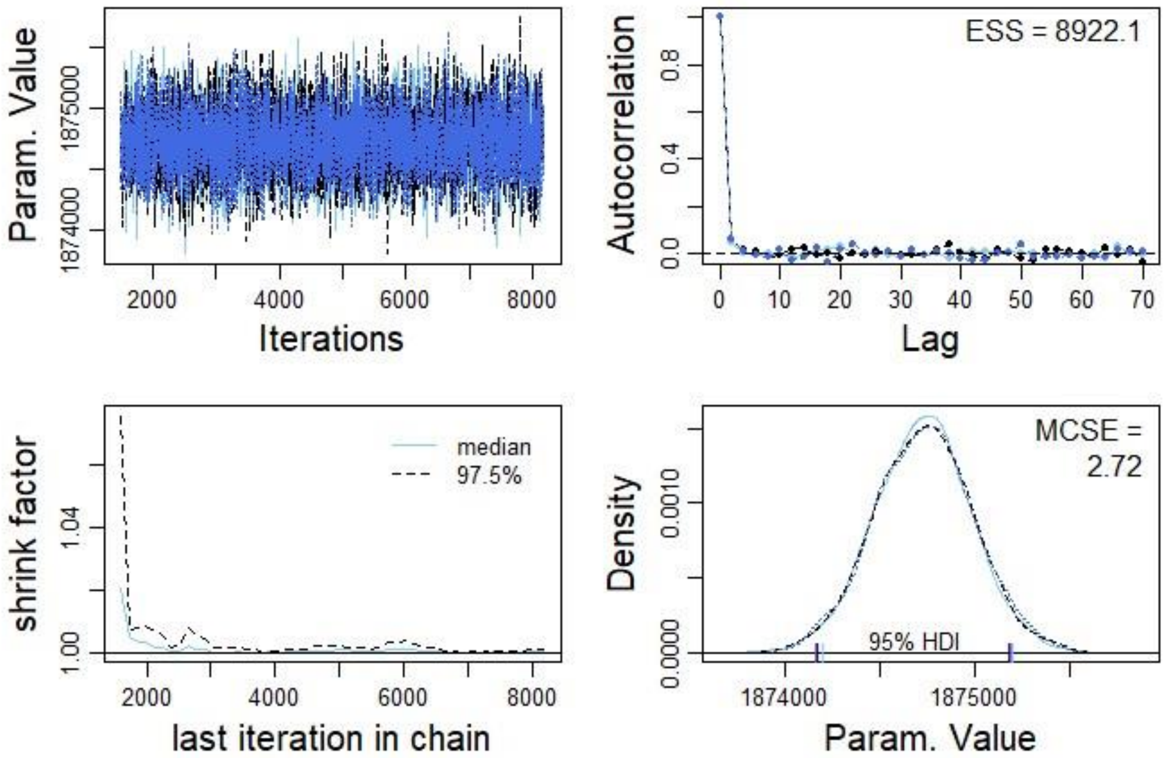
PREDICTION 5

Case 5: The prediction of sales for the properties if

Area =250, bedrooms =3, bathrooms =2, car parks =1 and property type = Unit.

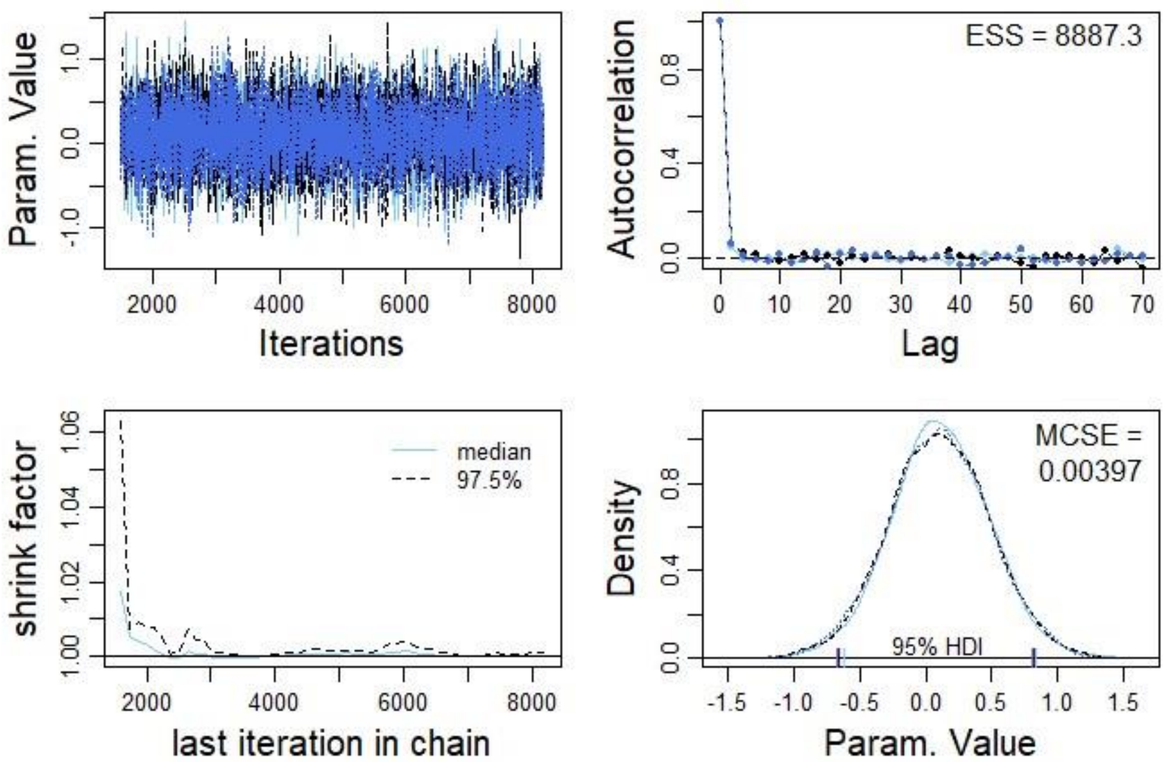


beta0

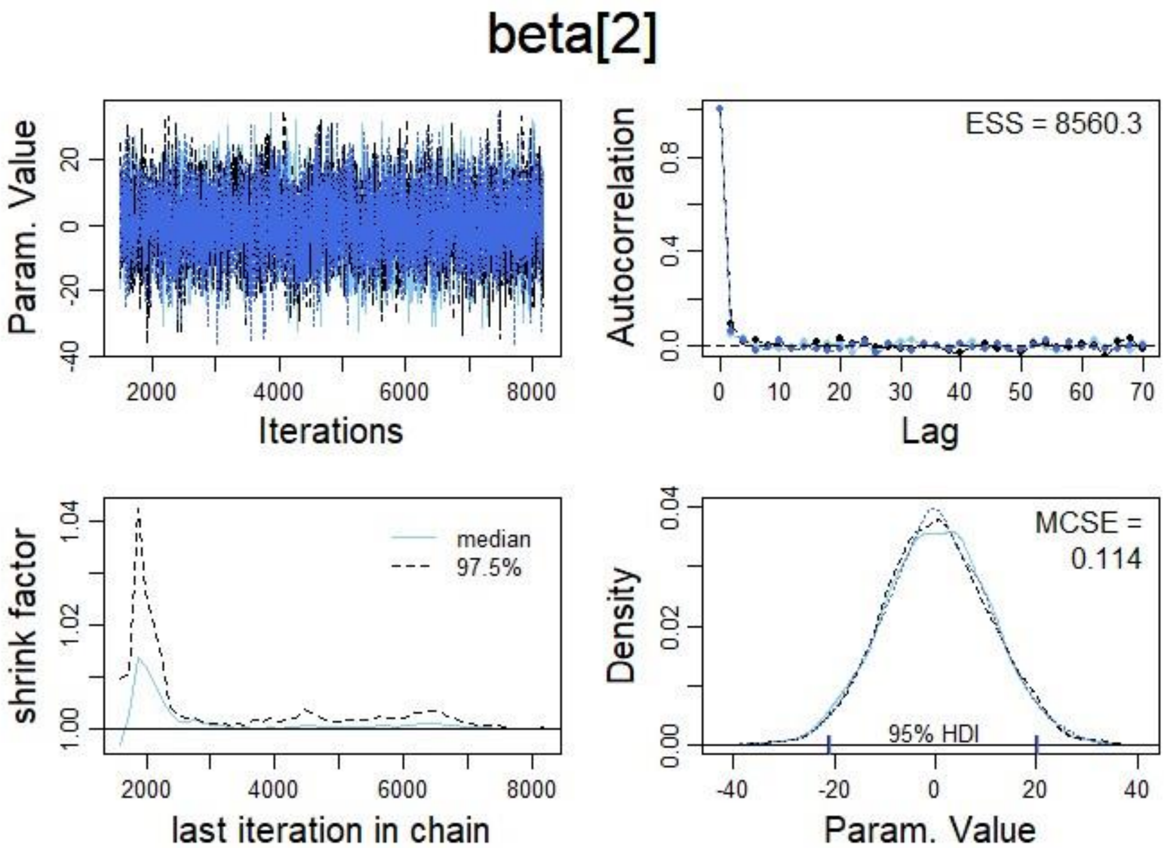


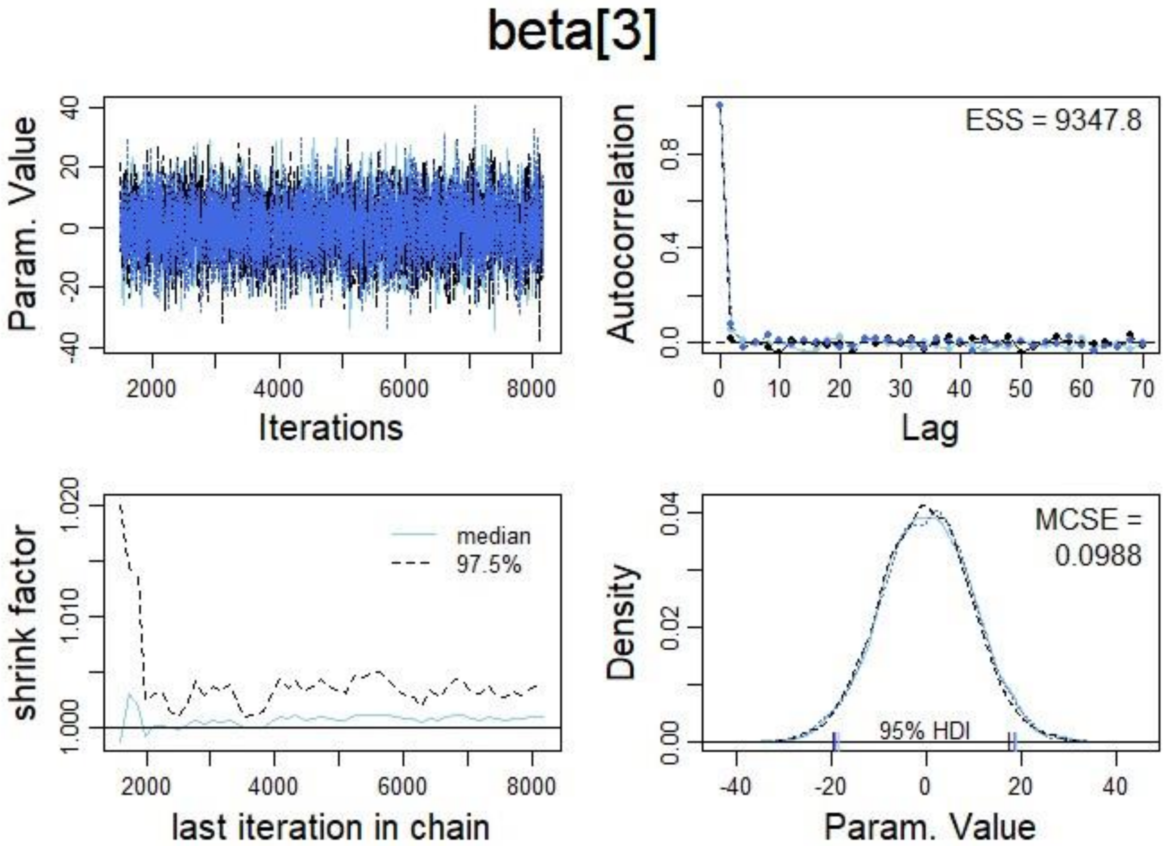
Param. Value
Iterations
ESS = 8922.1
Autocorrelation
Lag
shrink factor
median
97.5%
last iteration in chain
MCSE =
2.72
Density
95% HDI
Param. Value

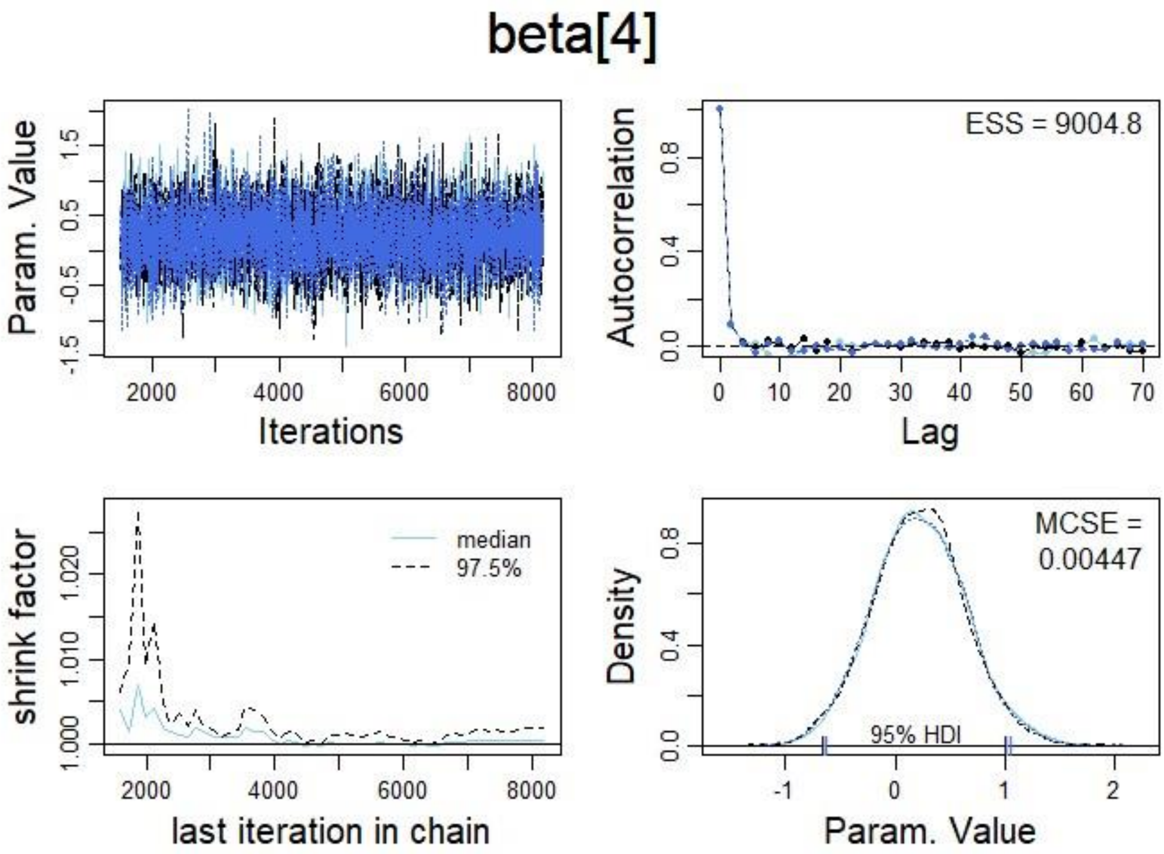
beta[1]

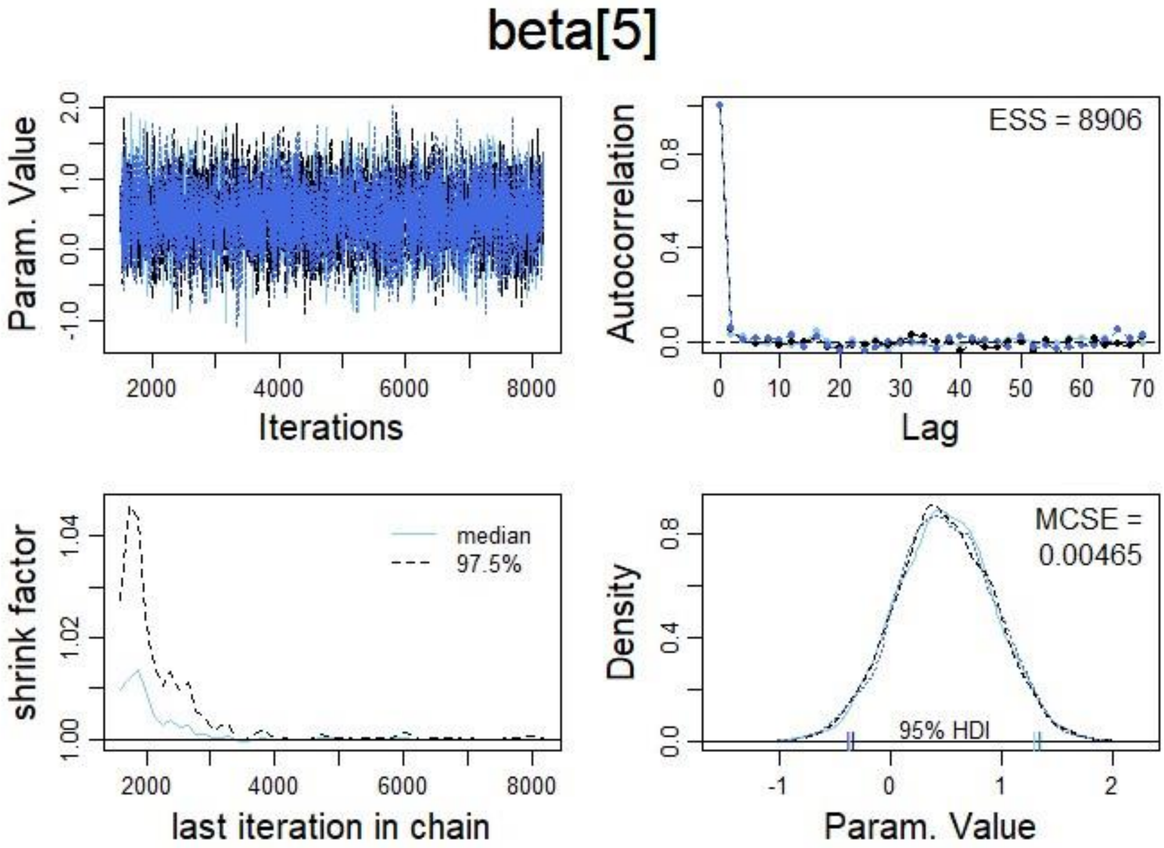


Param. Value
Iterations
ESS = 8887.3
Autocorrelation
Lag
shrink factor
median
97.5%
last iteration in chain
MCSE =
0.00397
Density
95% HDI
Param. Value

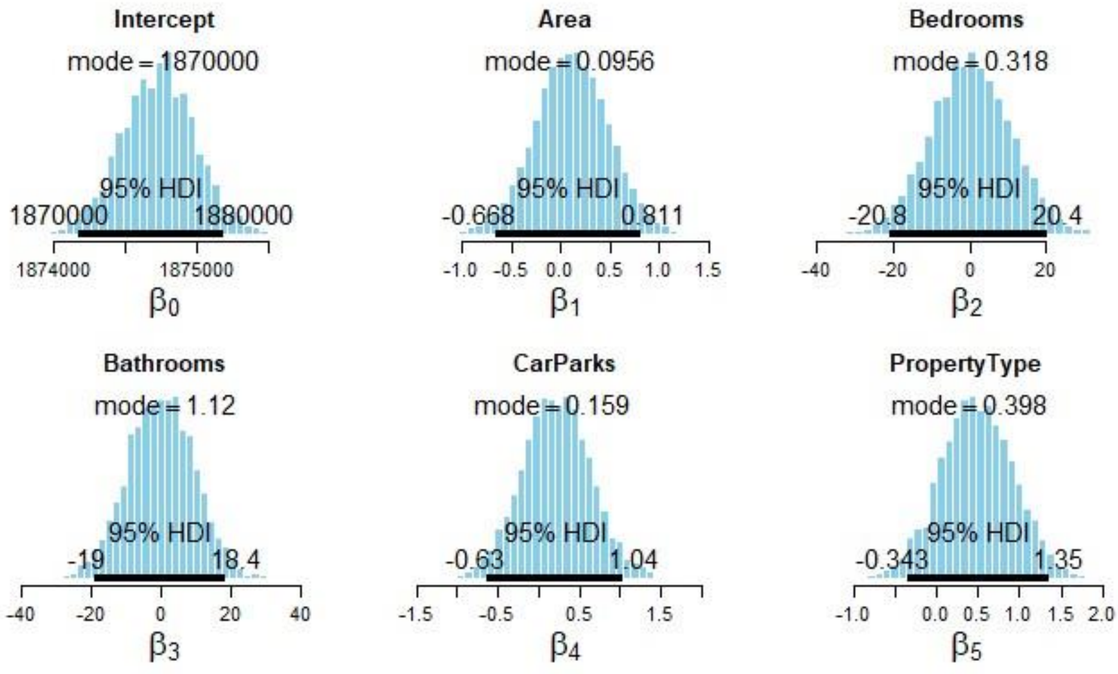

beta[2]
Param. Value
Iterations
Autocorrelation
ESS = 8560.3
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.114
95% HDI
Param. Value


beta[3]
Param. Value
Iterations
Autocorrelation
ESS = 9347.8
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.0988
95% HDI
Param. Value





Get summary statistics of chain:



CONCLUSION

The single and multiple linear regressions are performed successfully and the Salesprice is predicted.