Predictive Diamond Price with Bayesian Statistic

Shiting Yin (s3645072)

1. Introduction

This report based on data “diamonds” which recorded information relate to diamond price and other factors. The report contains seven sections. This section is introduction for basic information. Second section will show the details of dataset. Then, third section will load data into R software to do data pre-processing, scan the whole dataset and to see distribution for dependent attribute. The fourth section will visualize valuable insight into attributes, tidy and clean data into suitable format for Bayesian statistics. Moreover, with the help of jags, we will use Bayesian idea to build model for each random variable. The performance of each parameter distribution with MCMC diagnostics will be shown in section five. And then, compare each prediction price distribution with real one. Last section will give a summary to this report.

2. Dataset

The Kaggle (https://www.kaggle.com/shivam2503/diamonds) provides dataset, “diamonds” include 53940 observations and 11 attributes, cause by the limitation of running time, 5000 of instances selected randomly as training dataset, and last 5 observations will as the test data to valid the predicted value for diamond price. Applying fitted model to see whether the distribution for predicted diamond price is reasonable or not.

2.1 Descriptive Features

The information for this part based on Overview section which could be seen on Kaggle.

2.1.1 Independent variables:

- row_num: the number of row for each instance (1–53940) (numeric)
- caret: weight of the diamond (0.2–5.01) (numeric)
- cut: quality of the cut (categorical: Fair, Good, Very Good, Premium, Ideal)
- color: diamond color (categorical: from J(worst), I, H, G, F, E, D(best))
- clarity: a measurement of how clear the diamond is (categorical: I1(worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF(best))
- depth: total depth percentage = z/mean(x,y) = 2*z/(x+y) (43–79) (numeric)
- table: width of top of diamond relative to widest point (43–95) (numeric)

- x: length in mm (0–10.74) (numeric)
- y: width in mm (0–58.9) (numeric)
- z: depth in mm (0–31.8) (numeric)

2.1.2 Dependent variable (desired target):

- price: the price in US dollars ($326–$18823) (numeric)

3. Data Pre-processing

3.1 Preliminaries

Before doing further processing, considering whether there are some missing values in this dataset, if there are some missing value, this issue should be done at this stage with reasonable method. Fortunately, there is not any missing value in this data. Through data description, the values of depth calculate by z, in order to avoid autocorrelation in Bayesian, attribute z will be removed later. The row_num will be removed as well, because this one does not contribute for diamonds. For other attributes, there is no intuitional correlation information for them, considering through plots to see the relation between them.

3.2 Dependent Attribute (DiamondPrice)

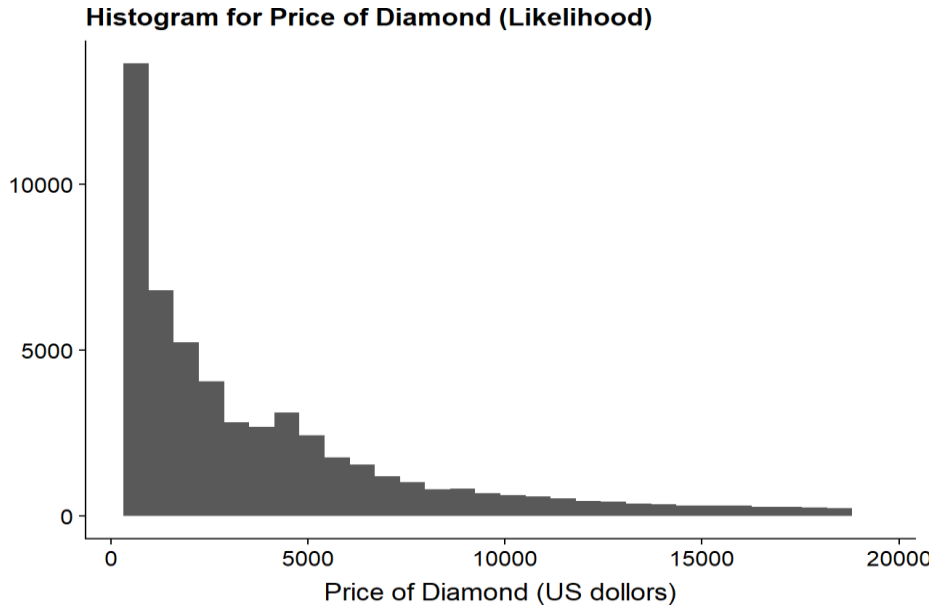


Figure1: Histogram for Diamond Price

Based on figure1, it seems that diamond price with whole dataset as gamma distribution; we could consider use gamma distribution for our model. However, larger enough dataset always could fit as normal distribution. Therefore, we could consider t distribution or normal distribution as well. For this report, we will using t distribution to see whether it return a good prediction for diamond price or not.

4. Data Exploration and Transformation

4.1 Exploration

This section, we will apply "ggplot" package to complete data visualization. Through these figures, whether there is a relation between two attributes could be seen clearly.

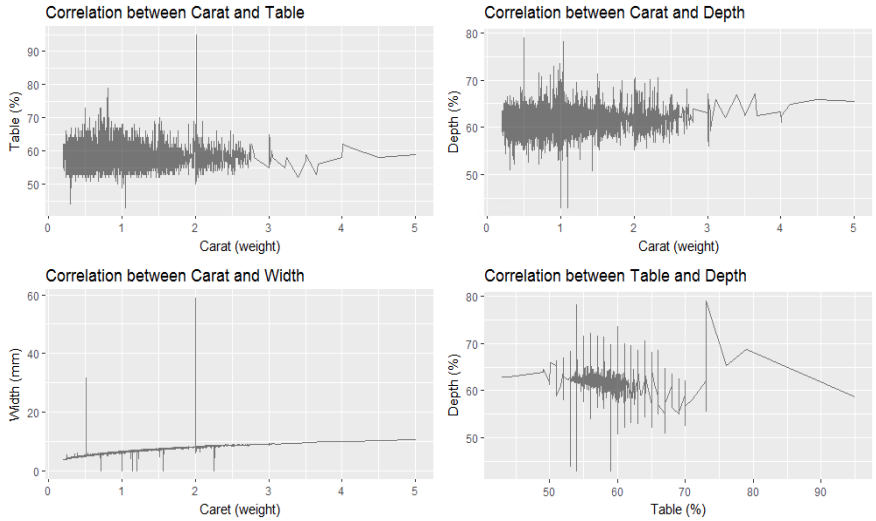


Figure2: Correlation between Carat and Other Attribute

Figure2 shows that the correlation between carat and table, carat and depth, carat and width, table and depth respectively. It seems that with the increase weight of carat, the width increase as well, looks like linear regression between them. For table and depth, looks like a slight correlation between them. Other two plots seem walk around their own mean level.

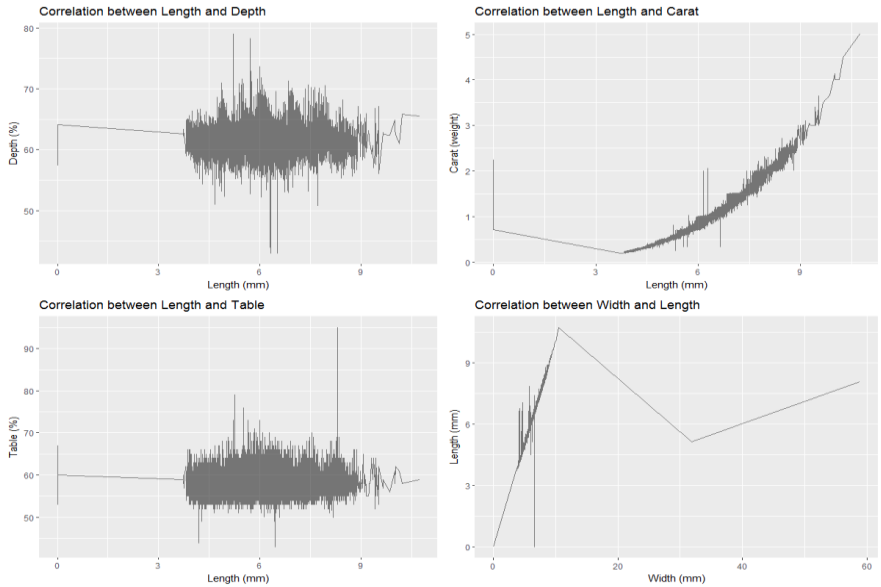


Figure3: Correlation between Length and Other Attributes

Figure3 shows an obvious linear trend between carat and length. Other plots show that no correlation between each other.

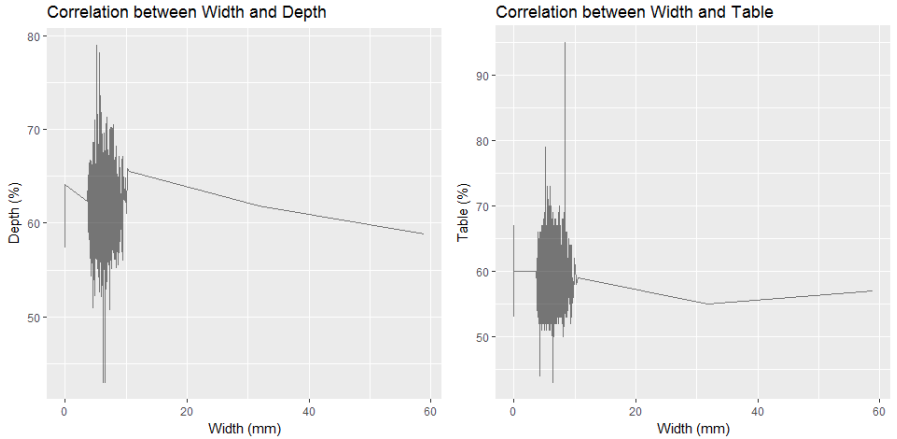


Figure4: Correlation between Width and Other Attributes

According to figure4, there is no correlation between them, all of observations around mean level without trends.

Based on data pre-processing, considering remove Width, Length and z attributes from dataset. So, attributes carat, cut, color, clarity, depth, table, and price as the final dataset for further Bayesian statistic.

4.2 Data Transformation

There are 3 categorical attributes in this data, in order to apply Bayesian statistic with regression, transforming them into binary number (0/1) should be done. The sample final dataset would be used could be seen as below.

Carat	Depth	Table	Cut fair	Cut good	Cut very G	Cut premi	Color J	Color I	Color H	Color G	Color F	Color E	Clarity I1	Clarity S	Clarity S	Clarity V	Clarity V	Clarity V	Clarity V	DiamondPrice
0.23	61.5	55	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	326
0.21	59.8	61	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	326
0.23	56.9	65	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	327
0.29	62.4	58	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	334
0.31	63.3	58	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	335
0.24	62.8	57	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	336
0.24	62.3	57	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	336
0.26	61.9	55	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	337
0.22	65.1	61	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	337
0.23	59.4	61	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	338
0.3	64	55	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	339
0.23	62.8	56	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	340
0.22	60.4	61	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	342
0.31	62.2	54	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	344

5. Jags Processing

5.1 Jags model diagram

The special idea for this diagram is that the normal distributions for betas are non-informative.

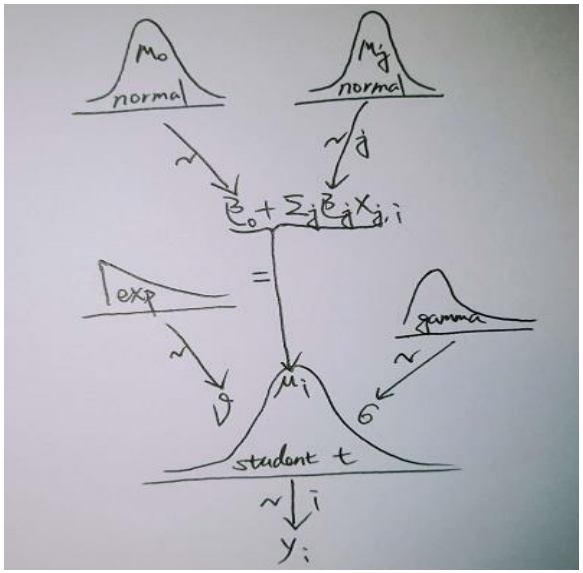


Figure5: Model Diagram

5.2 Model definition for JAGS

5.2.1 Data for JAGS

```
modelString = "
# Standardize the data:
data {
    ym <- mean(y)
    ysd <- sd(y)
    for ( i in 1:Ntotal ) {
    zy[i] <- ( y[i] - ym ) / ysd
    }
    for ( j in 1:Nx ) {
    xm[j]  <- mean(x[,j])
    xsd[j] <-   sd(x[,j])
    for ( i in 1:Ntotal ) {
    zx[i,j] <- ( x[i,j] - xm[j] ) / xsd[j]
}
}

# Specify the priors for original beta parameters
# Prior locations to reflect the expert information
    mu0 <- ym
# Set to overall mean a prior based on the interpretation of constant term in regression
    for ( j in 1:Nx ) {
      mu[j] = 0
    }


# Prior variances to reflect the expert information
# variance base on sample variance
    Var0 <- 1.0E+3 * (ysd)^2 # inference to sample variance

    for ( j in 1:Nx ) {
      Var[j] = 1.0E+3 * (ysd)^2
    }
```

```
# 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)

}
```

5.2.2 Comparison for Model1 and Model2

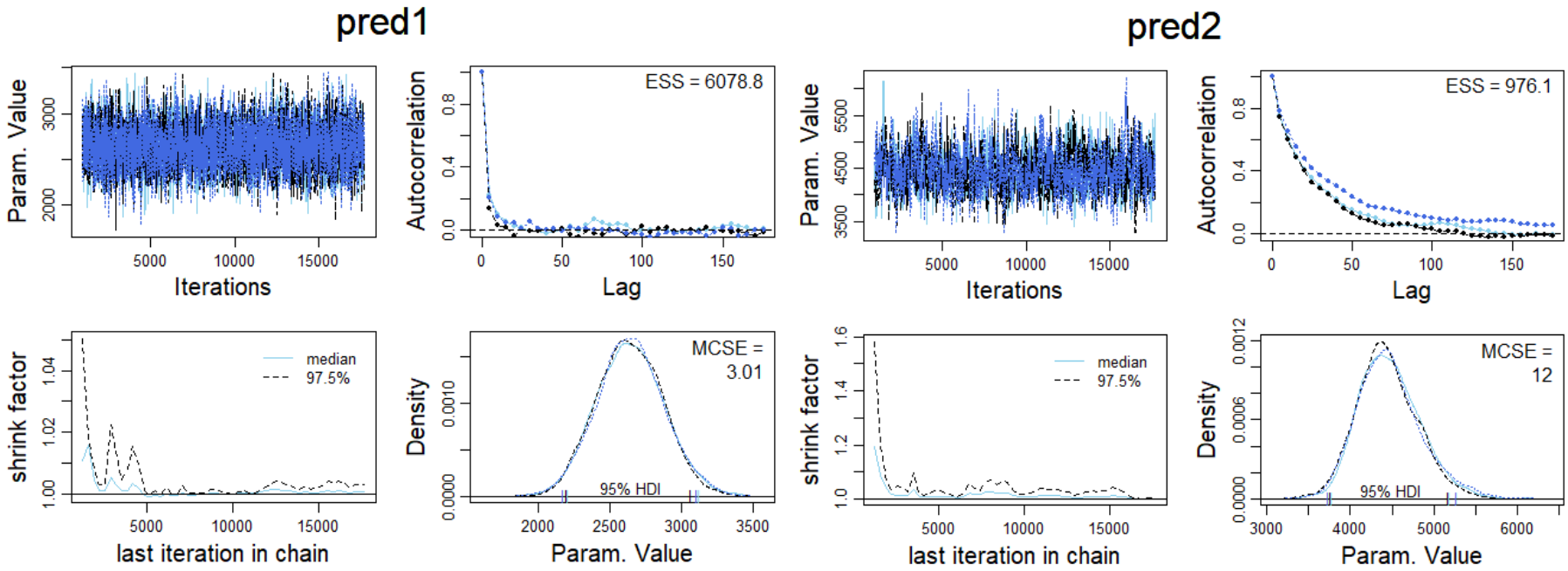
For this part, using 200 sample dataset that selected from population randomly to see which model is better for prediction diamond price. As we have not any idea about that which model is better, so, assign prior probability 0.5 for each model. The whole independent attributes contribute for model1, first 14 attributes as independent attributes for model2.

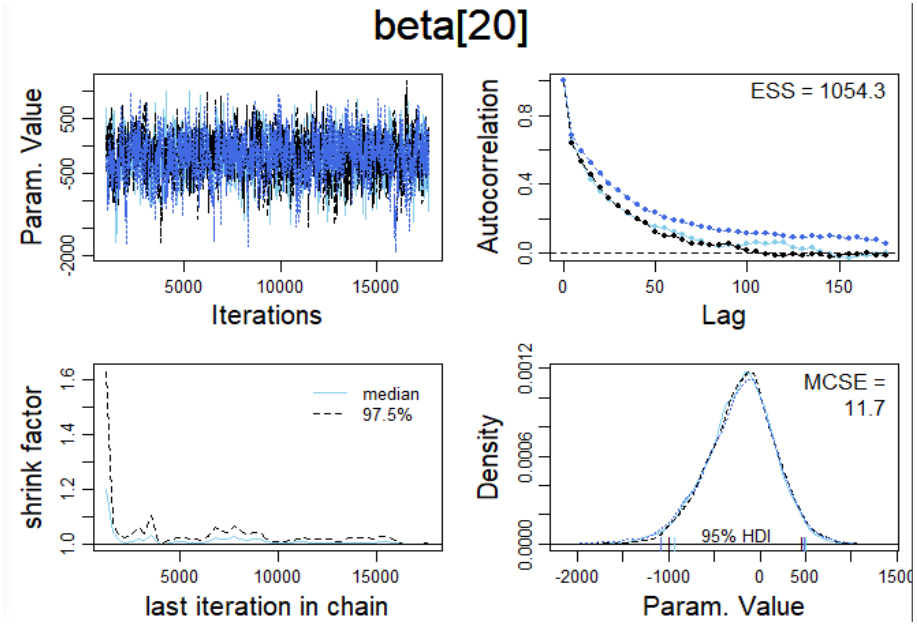
5.2.2.1 Model definition for 200 sample instances

```
# Specify the model for standardized data:
model {
    for ( i in 1:Ntotal ) {
    zy[i] ~ dt( ifelse(m==1, model1[i] , model2[i]) , 1/zsigma^2 , nu )

    model1[i] <- ( zbeta0 + sum(zbeta[1:Nx] * zx[i,1:Nx] ) )
    model2[i] <- (zbeta02 +sum(zbeta2[1:3] * zx[i,1:3]) )
    }

# Priors vague on standardized scale:
# set all of prior parameters for mu[i] as normal distribution
# cause domain is (-infinite, +infinite)
    zbeta0 ~ dnorm( muz0 , 1/varz0 )
    for ( j in 1:Nx ) {
    zbeta[j] ~ dnorm( muz[j] , 1/varz[j] )
    }

    zbeta02 ~ dnorm( muz0 , 1/varz0 )
    for ( j in 1:14 ) {
      zbeta2[j] ~ dnorm( muz[j] , 1/varz[j] )
    }

    zsigma ~ dgamma(0.01, 0.01) #zsigma is noninformative, similar as uniform
    nu ~ dexp(1/30.0) # nu is noninformative

  # Prior model probabilities
    m ~ dcat( mPriorProb[] )
    mPriorProb[1] <- .5
    mPriorProb[2] <- .5
# 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
  pred1 <- beta0 + beta[1] * xPred[1] + beta[2] * xPred[2] + beta[3] * xPred[3] +
  beta[4] * xPred[4] + beta[5] * xPred[5] + beta[6] * xPred[6] + beta[7] * xPred[7] +
  beta[8] * xPred[8] + beta[9] * xPred[9] + beta[10] * xPred[10] + beta[11] * xPred[11] +
  beta[12] * xPred[12] + beta[13] * xPred[13] + beta[14] * xPred[14] + beta[15] * xPred[15] +
  beta[16] * xPred[16] + beta[17] * xPred[17] + beta[18] * xPred[18] + beta[19] * xPred[19] +
  beta[20] * xPred[20]

  pred2 <- beta0 + beta[1] * xPred[1] + beta[2] * xPred[2] + beta[3] * xPred[3] +
  beta[4] * xPred[4] + beta[5] * xPred[5] + beta[6] * xPred[6] + beta[7] * xPred[7] +
  beta[8] * xPred[8] + beta[9] * xPred[9] + beta[10] * xPred[10] + beta[11] * xPred[11] +
  beta[12] * xPred[12] + beta[13] * xPred[13] + beta[14] * xPred[14]
```

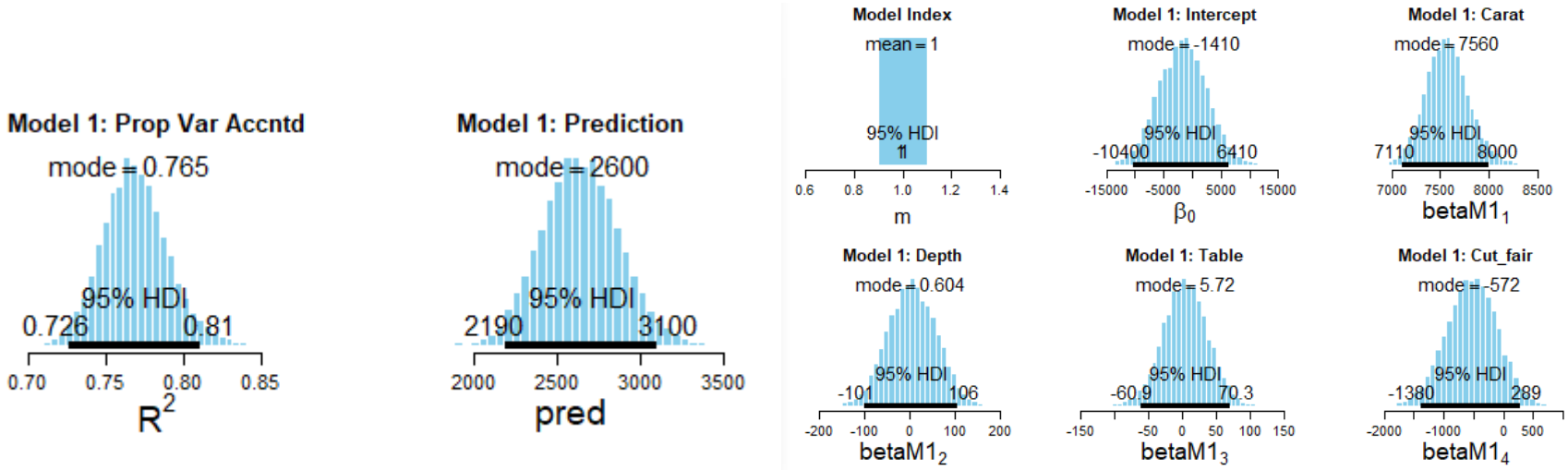
5.2.2.2 MCMC Diagnostics

According to MCMC diagnostics, prediction for model1 is better than model2 could be seen clearly. For both plots, the shrink factor is less than 1.1, iteration mixed well. However, auto-correlation is smaller for model1, so, the overlap with three chains, MCSE and ESS is better than model2. For model1, there are still some parameters have issue of auto-correlation like beta20 shows below.


pred1
Param. Value
Iterations
Autocorrelation
ESS = 6078.8
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 3.01
95% HDI
Param. Value
pred2
Param. Value
Iterations
Autocorrelation
ESS = 976.1
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 12
95% HDI
Param. Value


beta[20]
Param. Value
Iterations
Autocorrelation
ESS = 1054.3
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 11.7
95% HDI
Param. Value

5.2.2.3 Prediction for Model



Through this part, it seems that all of data go to model1, cause by the mean of model index is 1 with 95% HDI interval, and model1 shows reasonable prediction and good R square for us, although it just include 200 instances. Therefore, we will deploy model1 on the larger dataset to see the performance.

5.2.3 Deploy Model1 on Larger Dataset (5k instances)

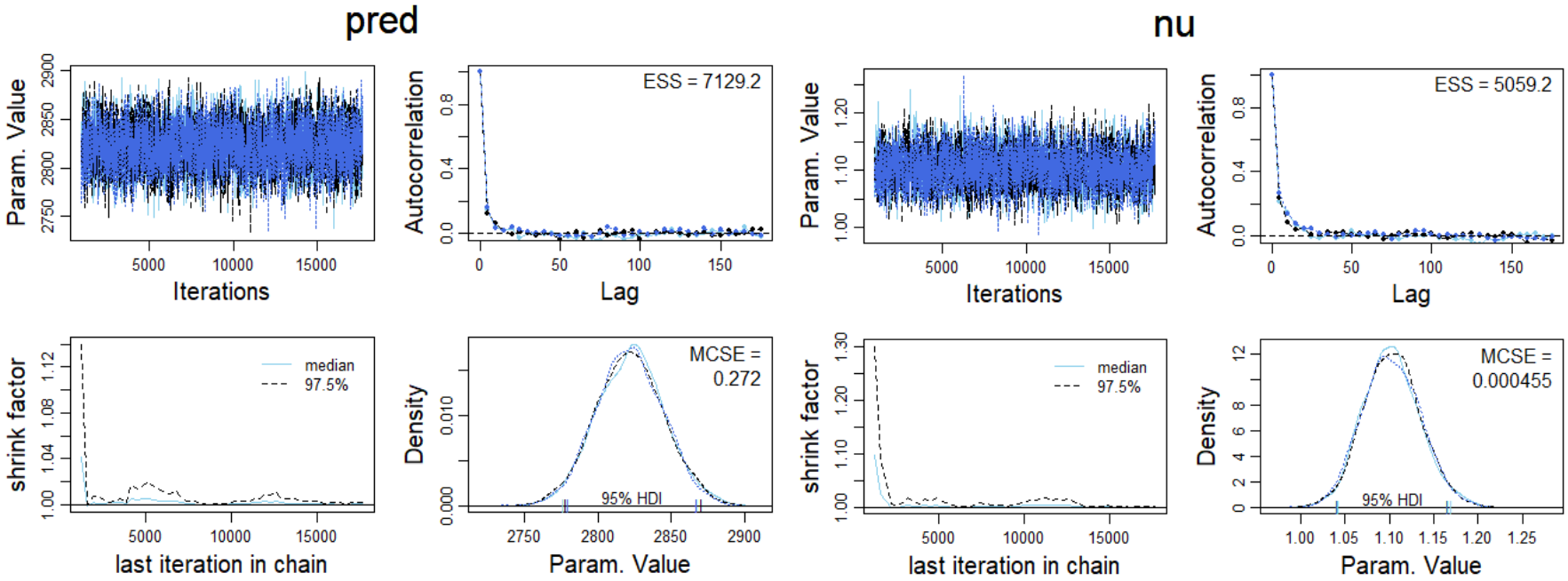
```
# Specify the model for standardized data:
model {
    for ( i in 1:Ntotal ) {
    zy[i] ~ dt( ifelse(m==1, model1[i] , model2[i]) , 1/zsigma^2 , nu )

    model1[i] <- ( zbeta0 + sum(zbeta[1:Nx] * zx[i,1:Nx] ) )
    model2[i] <- (zbeta02 +sum(zbeta2[1:3] * zx[i,1:3]) )
    }

# Priors vague on standardized scale:
# set all of prior parameters for mu[i] as normal distribution
# cause domain is (-infinite, +infinite)
    zbeta0 ~ dnorm( muz0 , 1/varz0 )
    for ( j in 1:Nx ) {
    zbeta[j] ~ dnorm( muz[j] , 1/varz[j] )
    }

    zbeta02 ~ dnorm( muz0 , 1/varz0 )
    for ( j in 1:14 ) {
      zbeta2[j] ~ dnorm( muz[j] , 1/varz[j] )
    }

    zsigma ~ dgamma(0.01, 0.01) #zsigma is noninformative, similar as uniform
    nu ~ dexp(1/30.0) # nu is noninformative

  # Prior model probabilities
    m ~ dcat( mPriorProb[] )
    mPriorProb[1] <- .5
    mPriorProb[2] <- .5
```

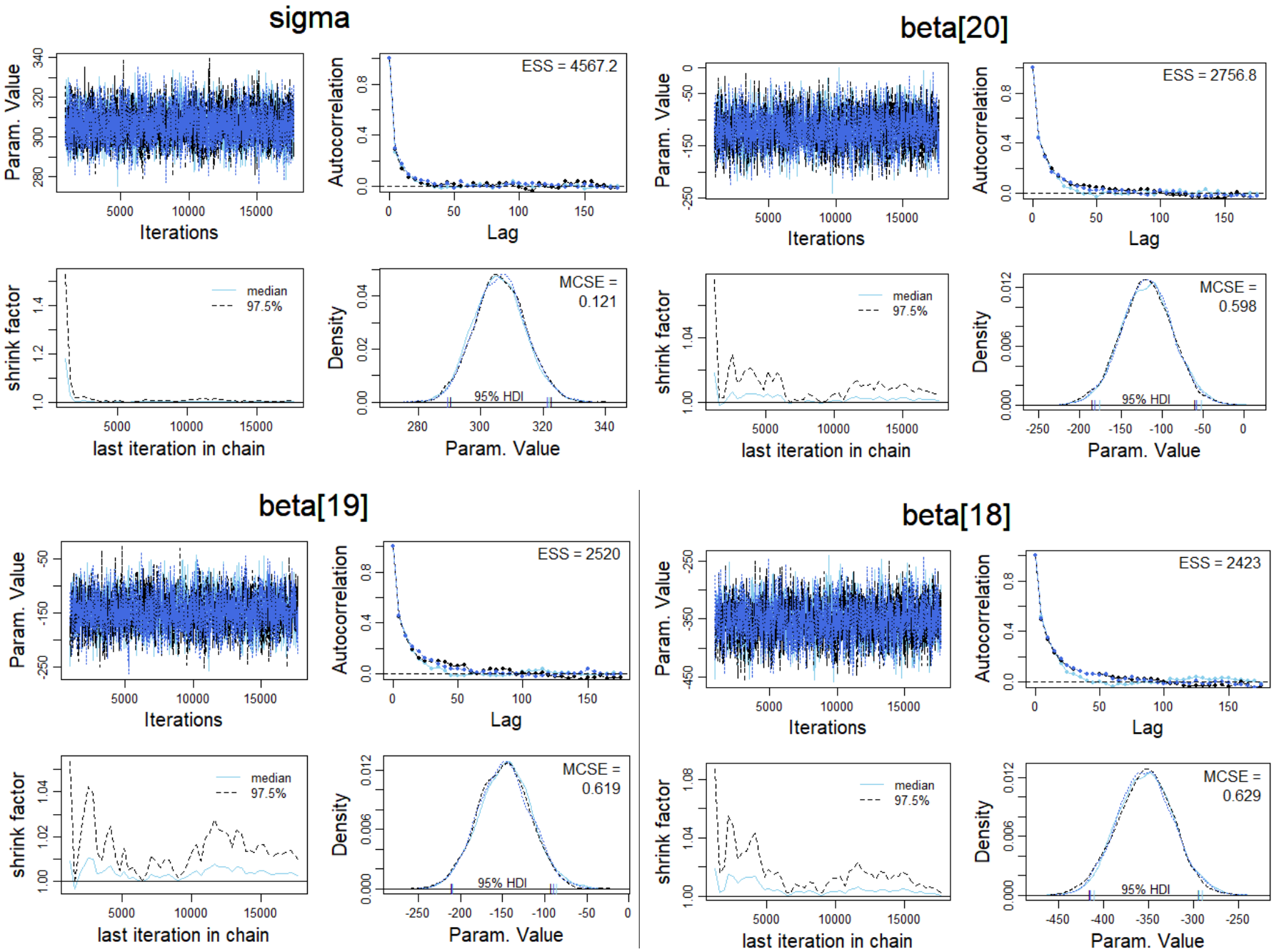
5.3 MCMC Diagnostics for 5k Instances

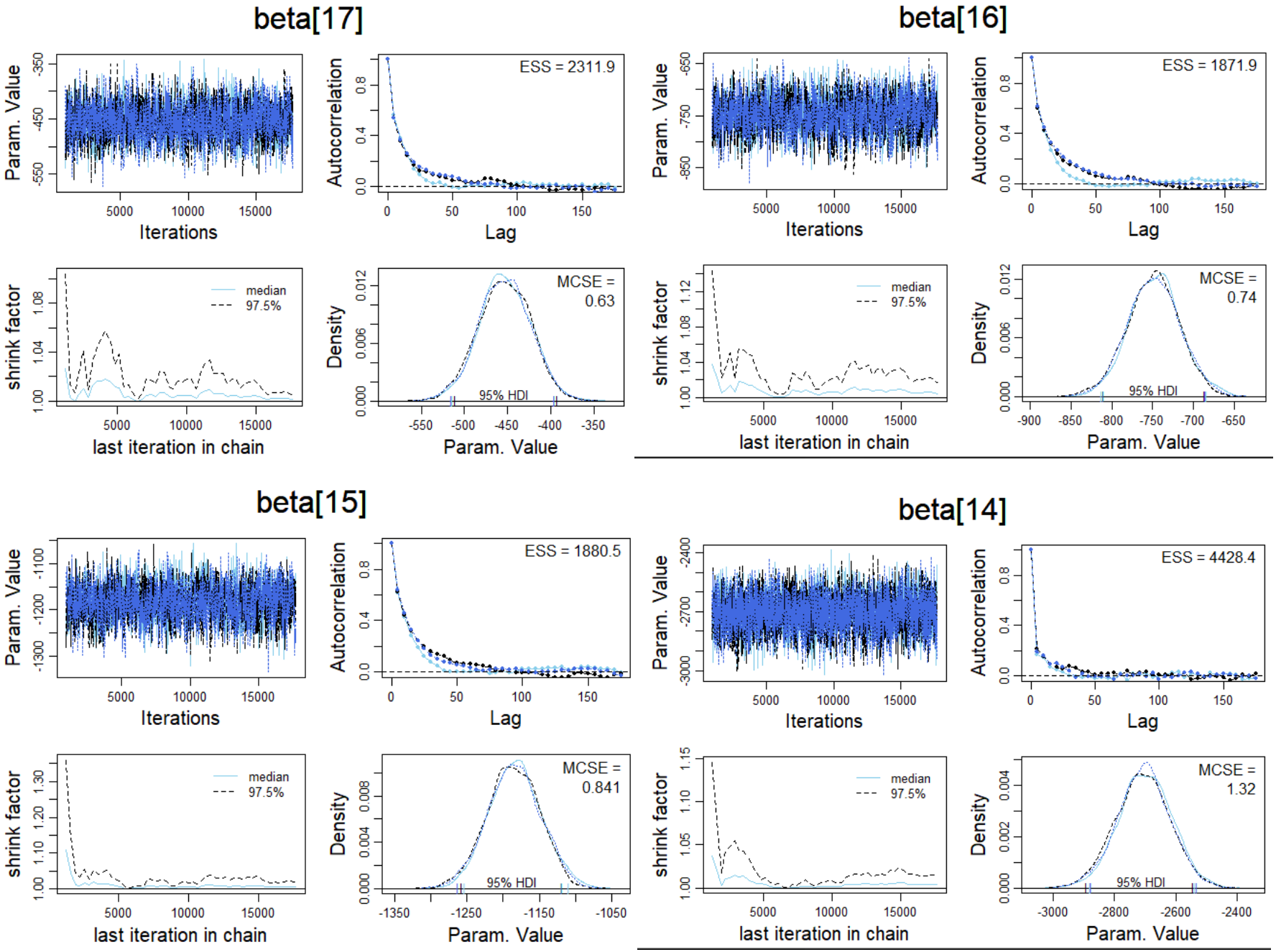
The following figures will show the MCMC diagnostics details for each test instance. It seems that all of diagnostics are good, all shrink factor are less than 1.1, the iteration part mixed well, 3 chains overlap well with lower MCSE, and ESS is high enough with lower autocorrelation that close to zero. Furthermore, for this part, we could see the issue of autocorrelation with betas has been improved by the increase number of instances. Because of the limitation of report, from instance 2 to instance 5, just some of diagnostics display in the following part. In fact, they all show the good diagnostics like instance 1.

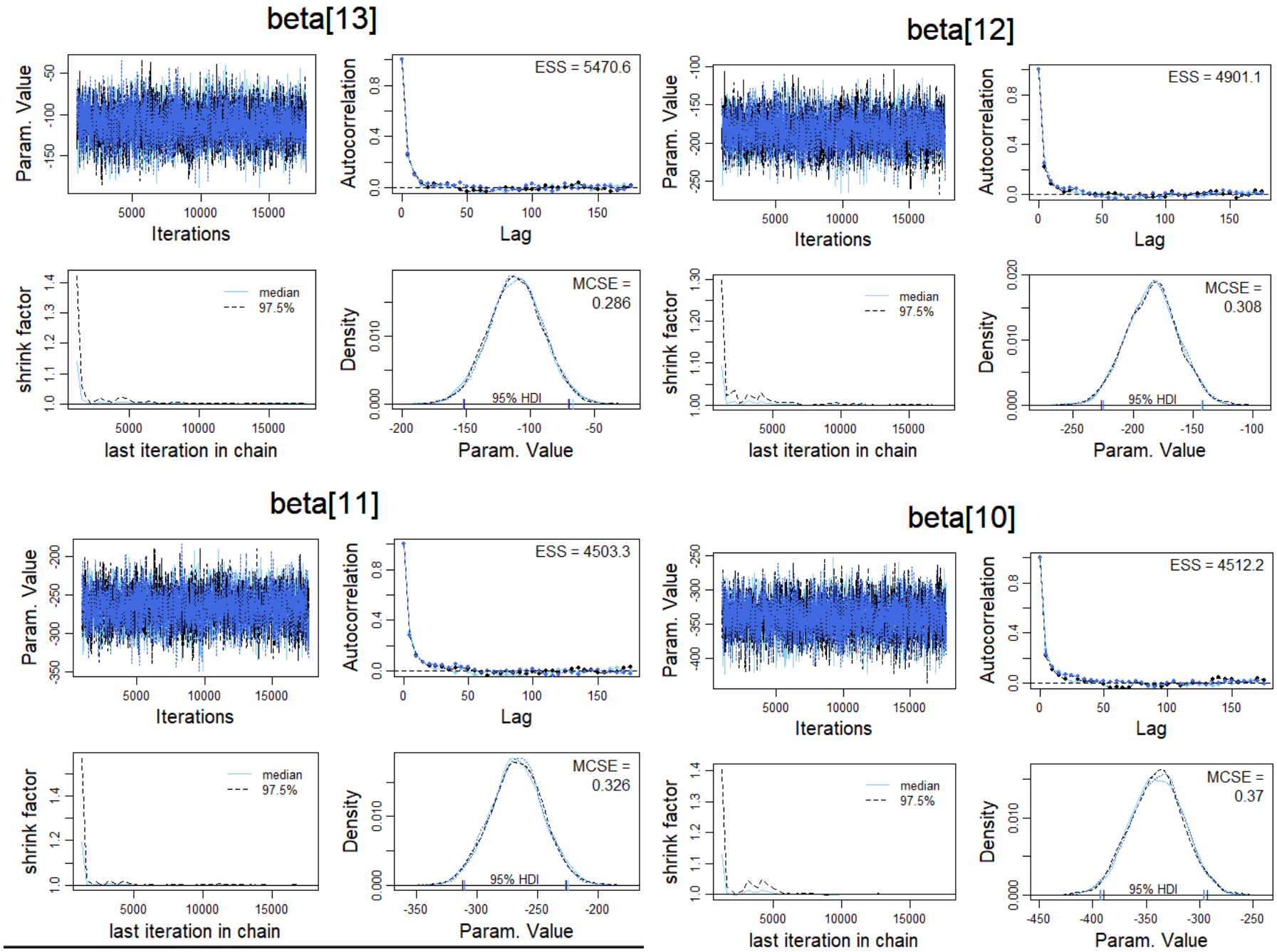
5.3.1 Diagnostic for instance (carat=0.75, cut="Ideal", color="D", clarity="SI2", depth=62.2, table=55)

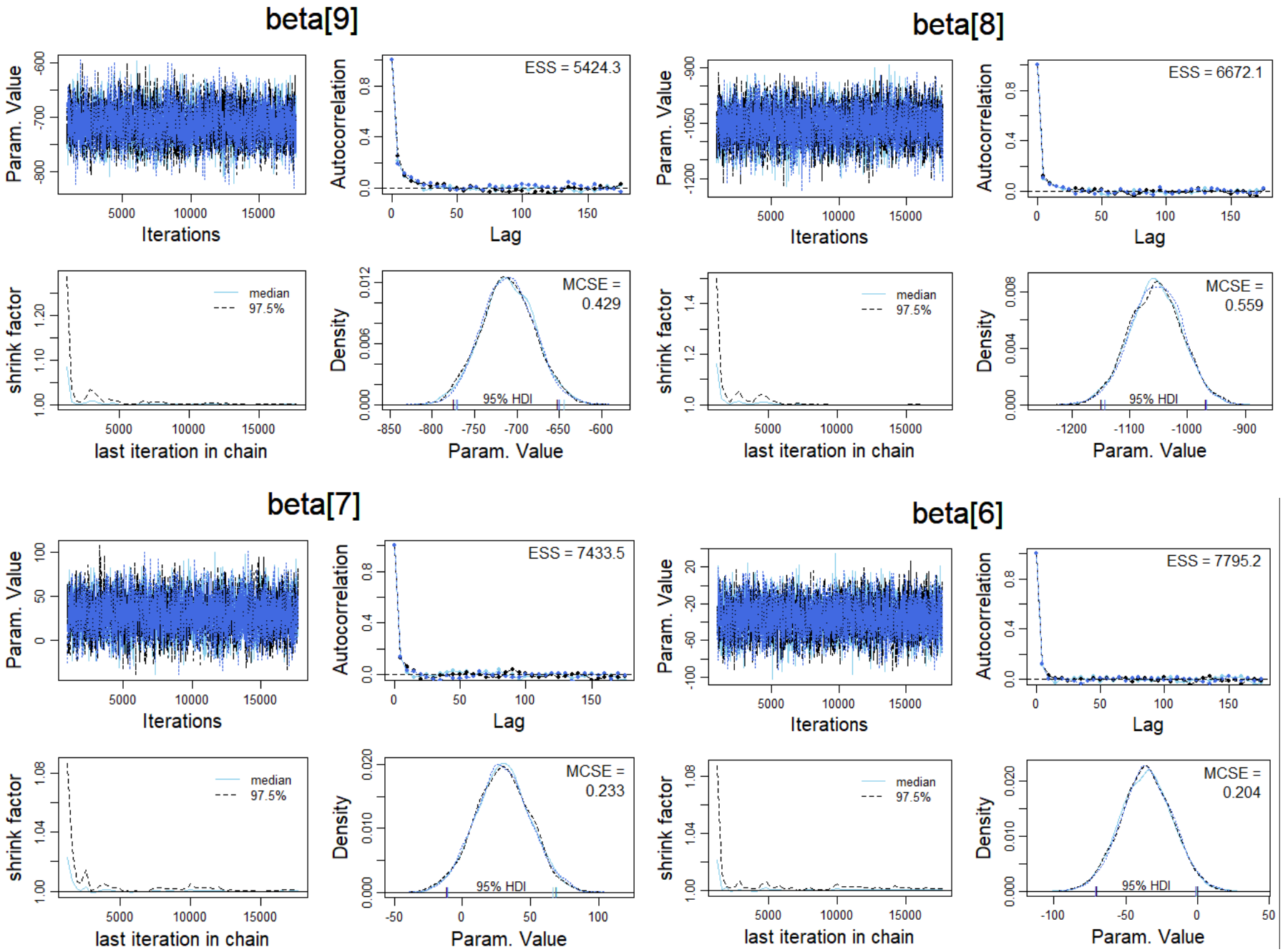
c (0.75, 62.2, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)

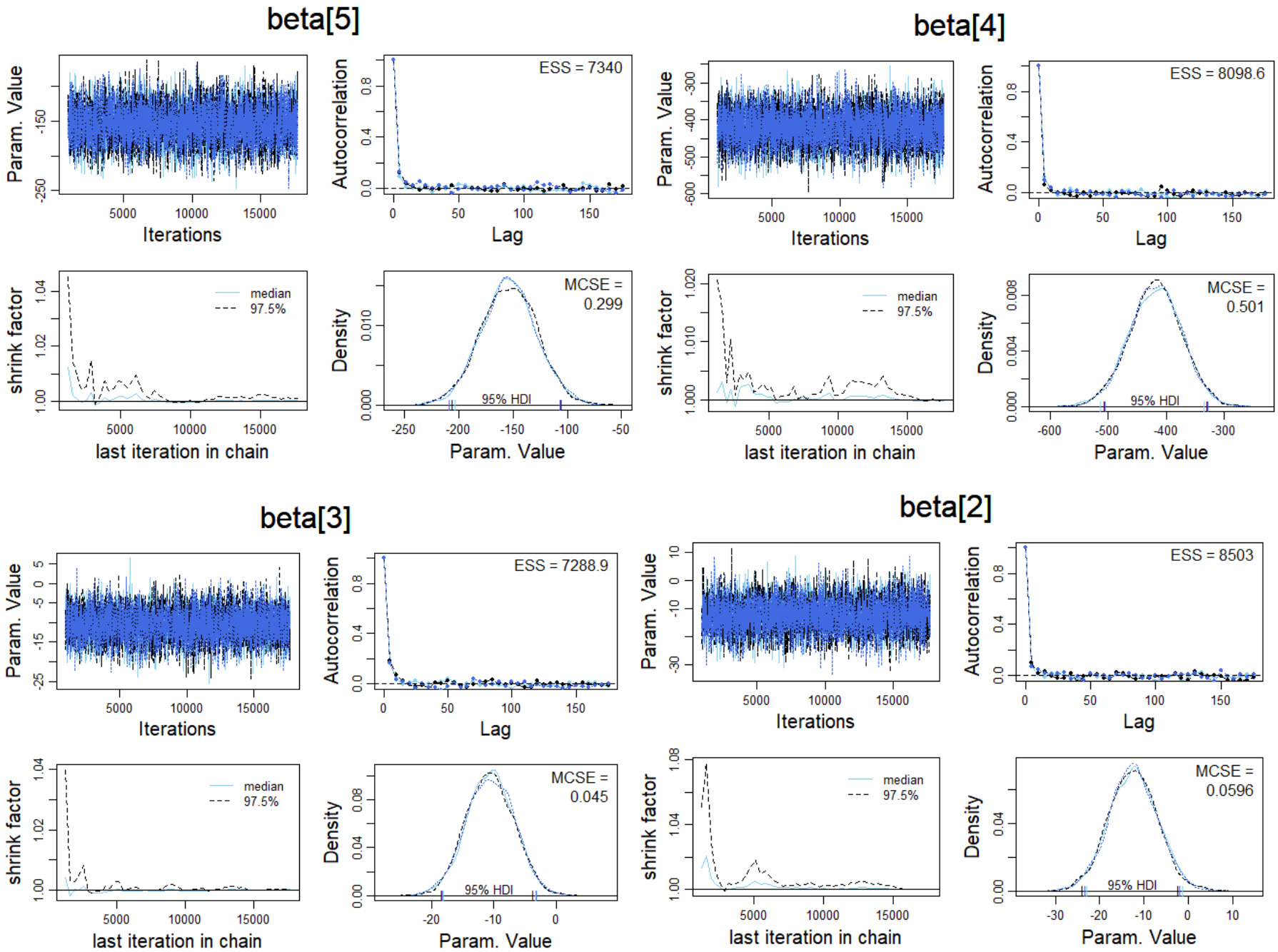


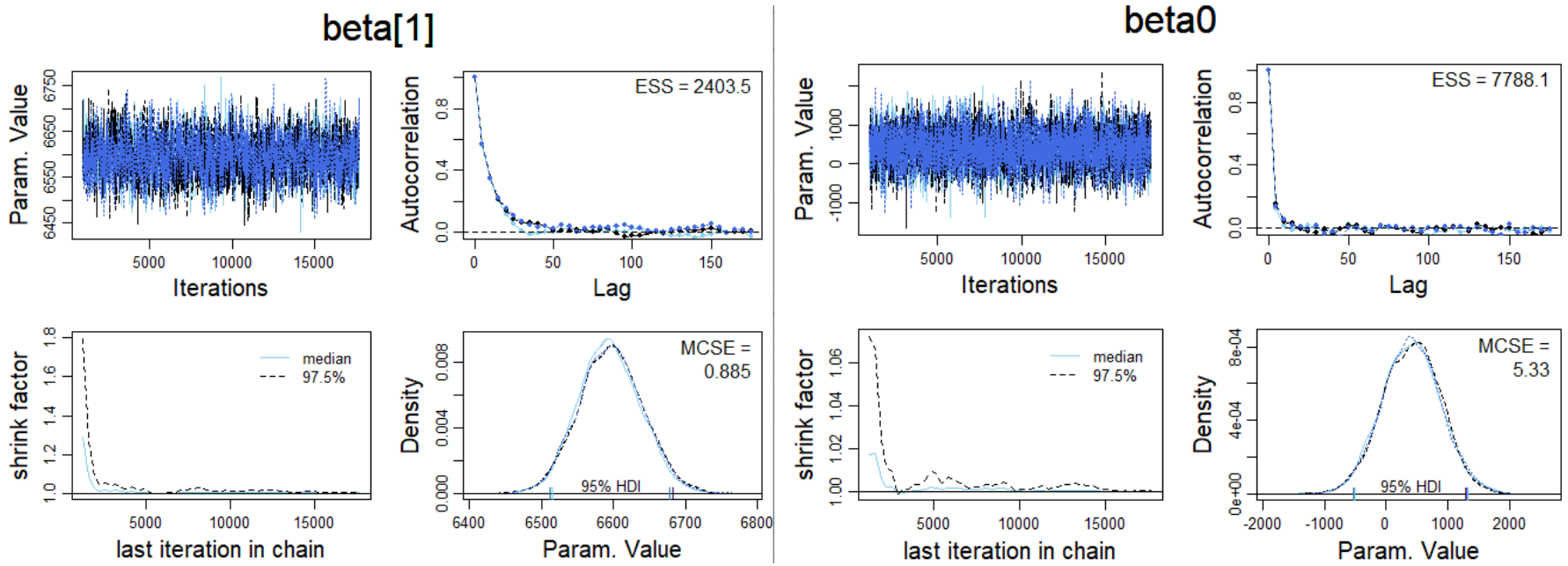

sigma
Param. Value
Iterations
Autocorrelation
ESS = 4567.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.121
95% HDI
Param. Value
beta[20]
Param. Value
Iterations
Autocorrelation
ESS = 2756.8
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.598
95% HDI
Param. Value
beta[19]
Param. Value
Iterations
Autocorrelation
ESS = 2520
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.619
95% HDI
Param. Value
beta[18]
Param. Value
Iterations
Autocorrelation
ESS = 2423
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.629
95% HDI
Param. Value


beta[17]
Param. Value
Iterations
Autocorrelation
ESS = 2311.9
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.63
95% HDI
Param. Value
beta[16]
Param. Value
Iterations
Autocorrelation
ESS = 1871.9
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.74
95% HDI
Param. Value
beta[15]
Param. Value
Iterations
Autocorrelation
ESS = 1880.5
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.841
95% HDI
Param. Value
beta[14]
Param. Value
Iterations
Autocorrelation
ESS = 4428.4
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 1.32
95% HDI
Param. Value


beta[13]
Param. Value
Iterations
Autocorrelation
ESS = 5470.6
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.286
95% HDI
Param. Value
beta[12]
Param. Value
Iterations
Autocorrelation
ESS = 4901.1
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.308
95% HDI
Param. Value
beta[11]
Param. Value
Iterations
Autocorrelation
ESS = 4503.3
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.326
95% HDI
Param. Value
beta[10]
Param. Value
Iterations
Autocorrelation
ESS = 4512.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.37
95% HDI
Param. Value

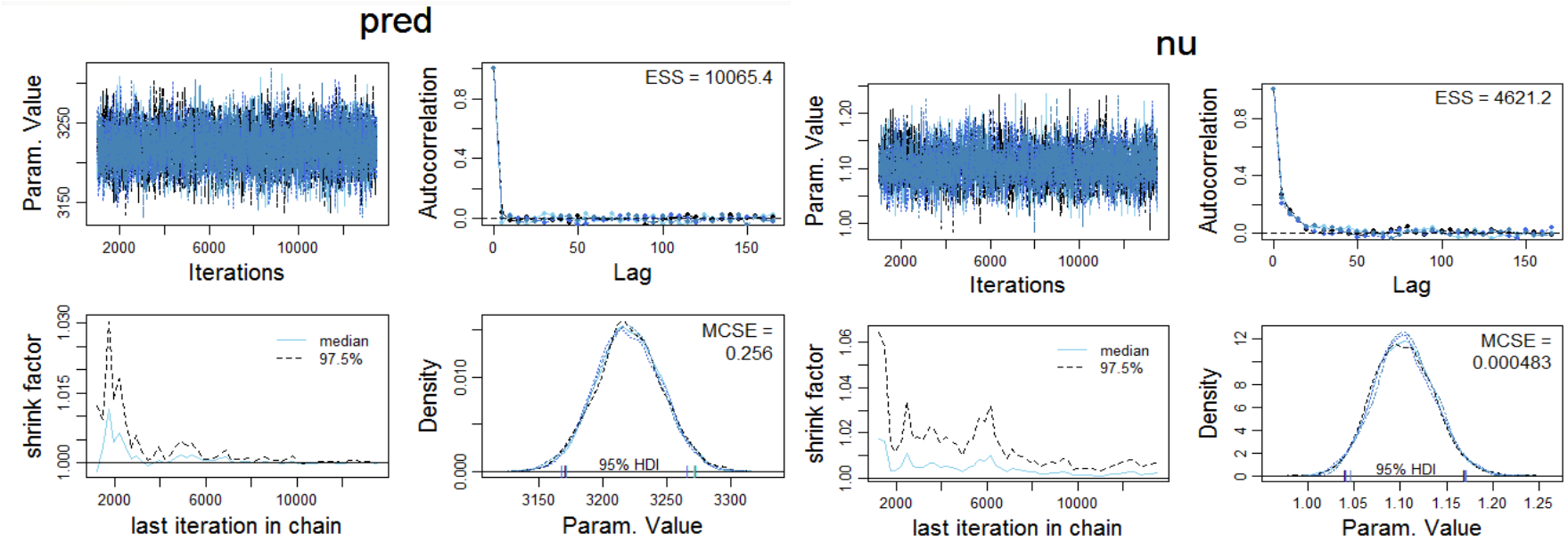

beta[9]
Param. Value
Iterations
Autocorrelation
ESS = 5424.3
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.429
95% HDI
Param. Value
beta[8]
Param. Value
Iterations
Autocorrelation
ESS = 6672.1
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.559
95% HDI
Param. Value
beta[7]
Param. Value
Iterations
Autocorrelation
ESS = 7433.5
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.233
95% HDI
Param. Value
beta[6]
Param. Value
Iterations
Autocorrelation
ESS = 7795.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.204
95% HDI
Param. Value

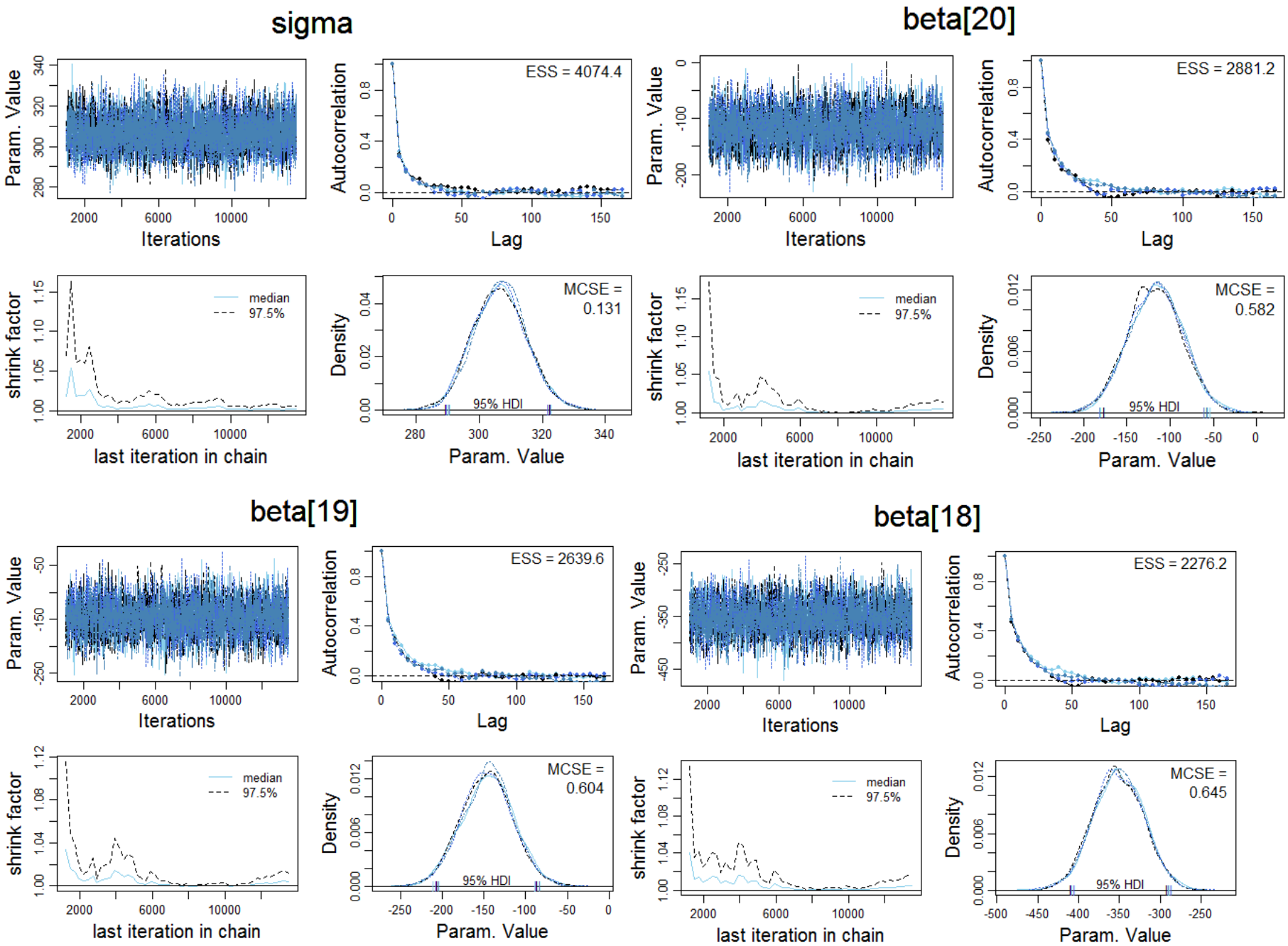

beta[5]
Param. Value
Iterations
Autocorrelation
ESS = 7340
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.299
95% HDI
Param. Value
beta[4]
Param. Value
Iterations
Autocorrelation
ESS = 8098.6
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.501
95% HDI
Param. Value
beta[3]
Param. Value
Iterations
Autocorrelation
ESS = 7288.9
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.045
95% HDI
Param. Value
beta[2]
Param. Value
Iterations
Autocorrelation
ESS = 8503
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.0596
95% HDI
Param. Value



5.3.2 Diagnostic for instance (carat=0.86, cut="Premium", color="H", clarity="SI2", depth=61, table=58)

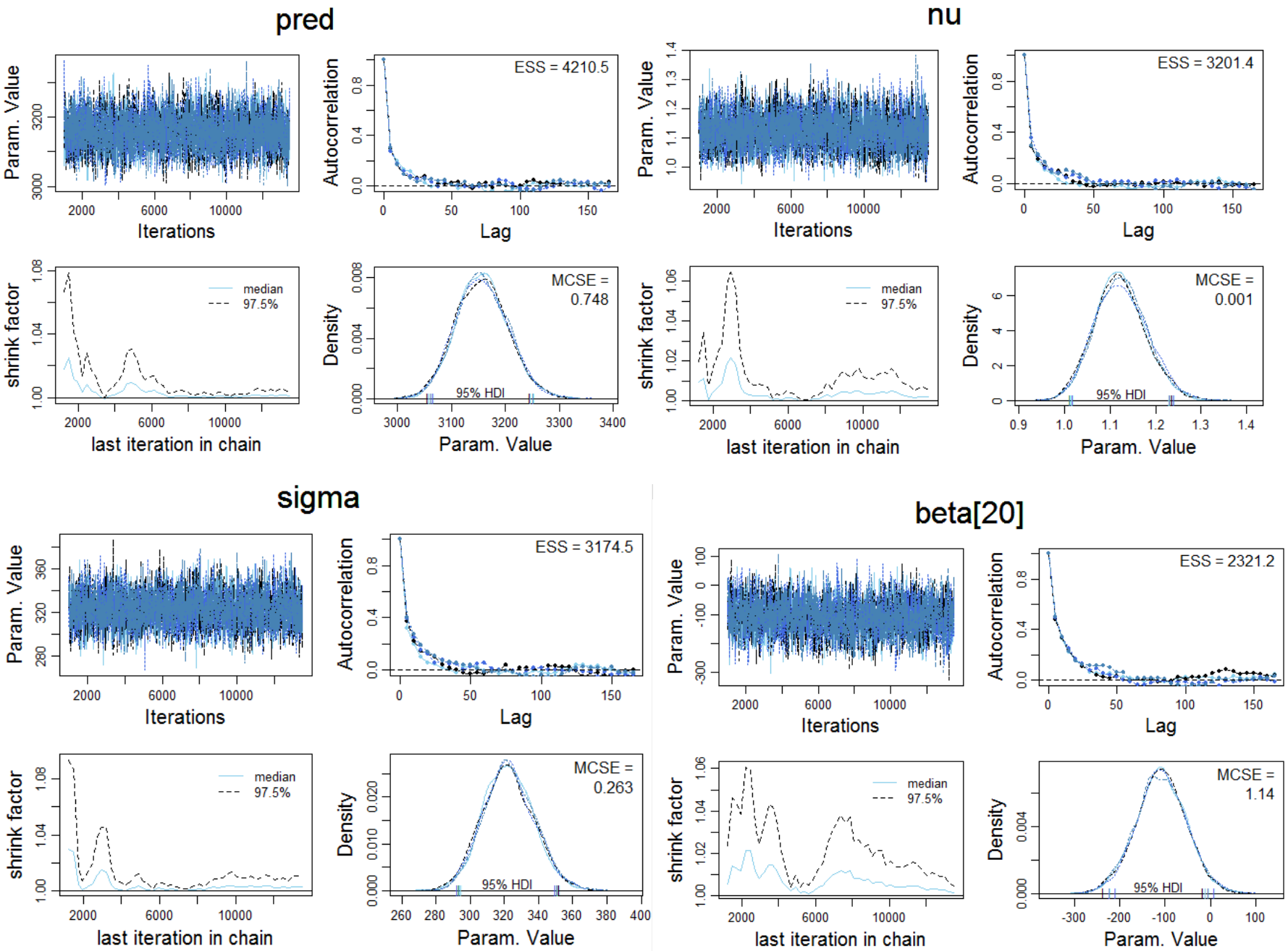
c(0.86, 61, 58, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)

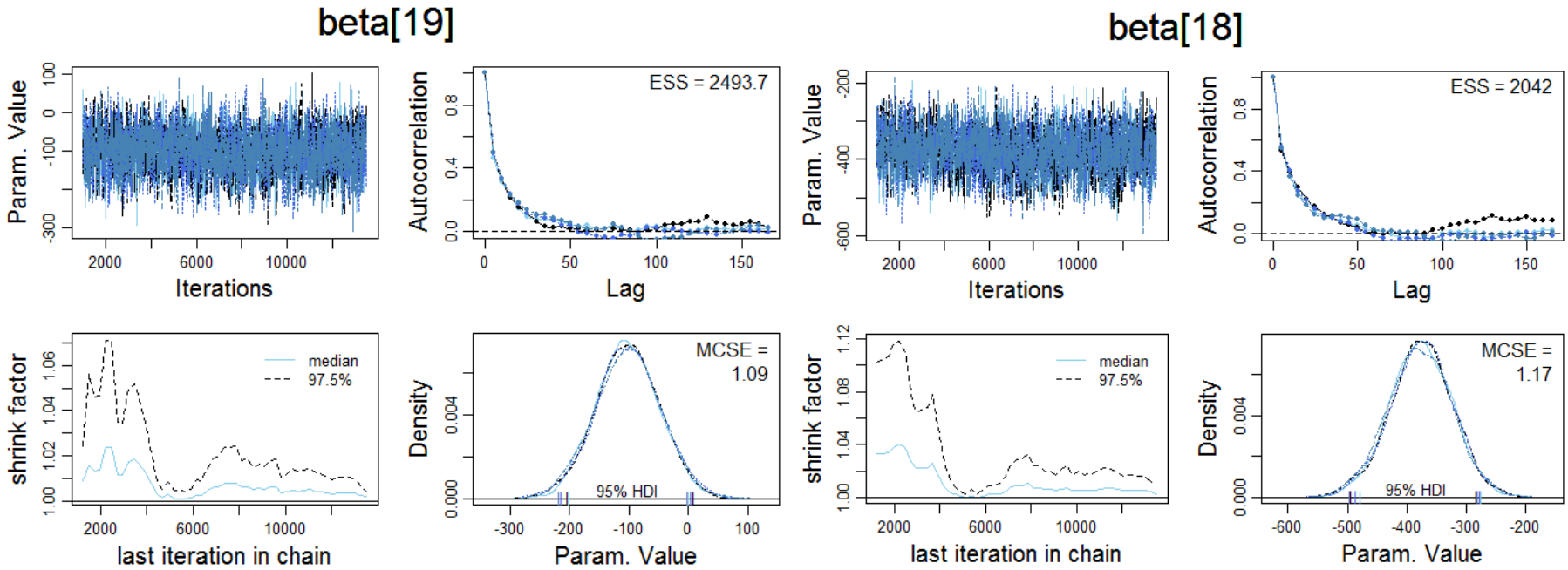



sigma
Param. Value
Iterations
Autocorrelation
ESS = 4074.4
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.131
95% HDI
Param. Value
beta[20]
Param. Value
Iterations
Autocorrelation
ESS = 2881.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.582
95% HDI
Param. Value
beta[19]
Param. Value
Iterations
Autocorrelation
ESS = 2639.6
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.604
95% HDI
Param. Value
beta[18]
Param. Value
Iterations
Autocorrelation
ESS = 2276.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.645
95% HDI
Param. Value

5.3.3 Diagnostic for instance (carat=0.7, cut="Very Good", color="D", clarity="SI1", depth=62.8, table=60)

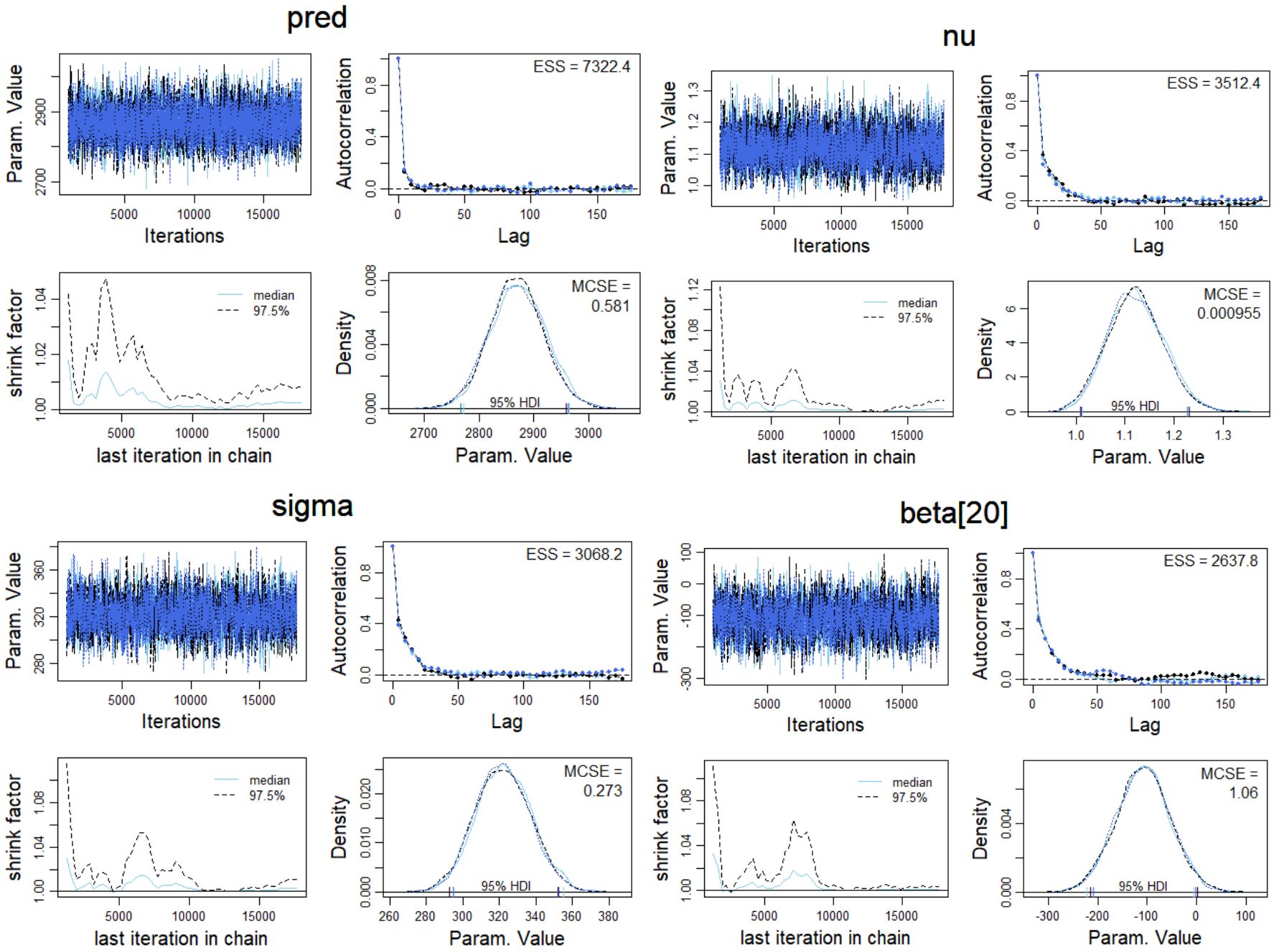
c(0.7, 62.8, 60, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)

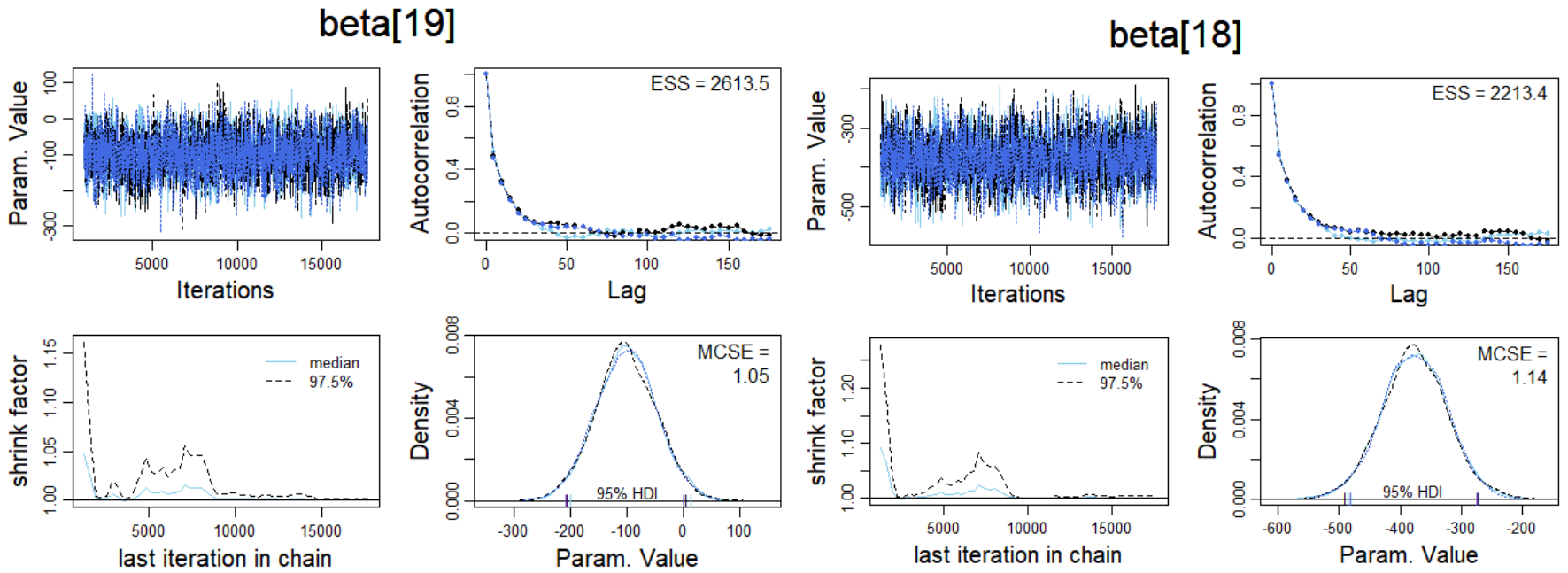




5.3.4 Diagnostic for instance (carat=0.72, cut="Good", color="D", clarity="SI1", depth=63.1, table=55)

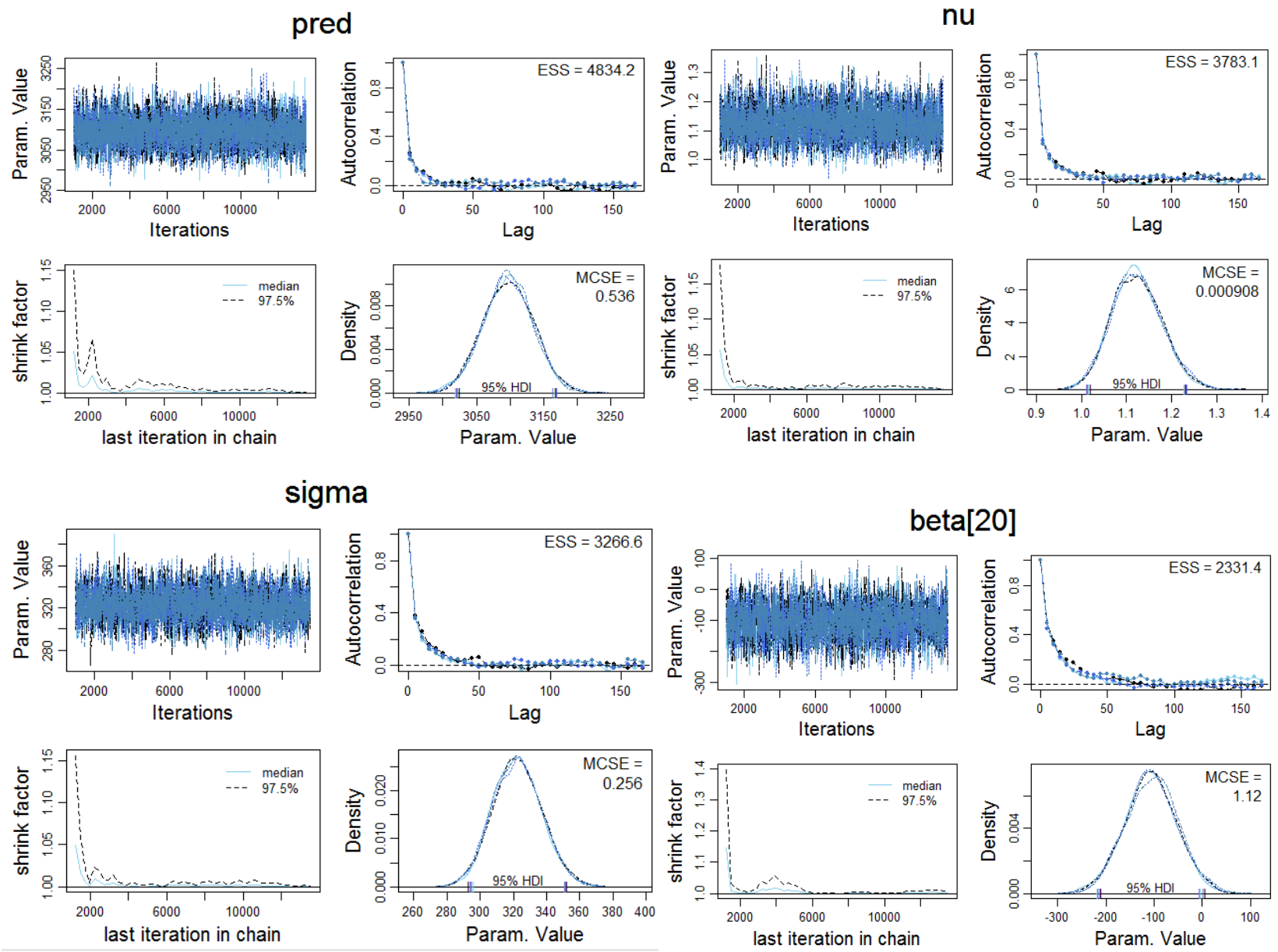
c(0.72, 63.1, 55, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)

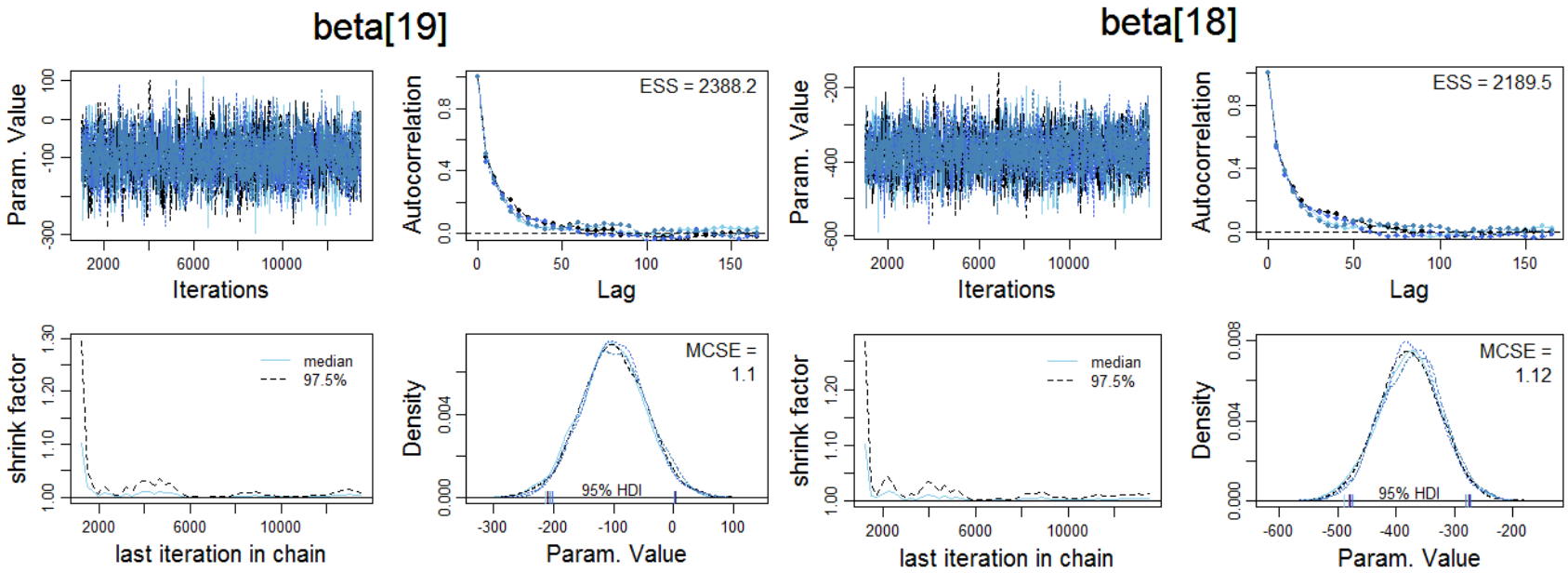

pred
Param. Value
Iterations
Autocorrelation
ESS = 7322.4
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.581
95% HDI
Param. Value
nu
Param. Value
Iterations
Autocorrelation
ESS = 3512.4
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.000955
95% HDI
Param. Value
sigma
Param. Value
Iterations
Autocorrelation
ESS = 3068.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 0.273
95% HDI
Param. Value
beta[20]
Param. Value
Iterations
Autocorrelation
ESS = 2637.8
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE = 1.06
95% HDI
Param. Value



5.3.5 Diagnostic for instance (carat=0.72, cut="Ideal", color="D", clarity="SI1", depth=60.8, table=57)

c(0.72, 60.8, 57, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)

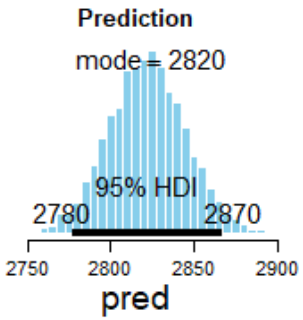

pred
Param. Value
Iterations
Autocorrelation
ESS = 4834.2
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.536
95% HDI
Param. Value
nu
Param. Value
Iterations
Autocorrelation
ESS = 3783.1
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.000908
95% HDI
Param. Value
sigma
Param. Value
Iterations
Autocorrelation
ESS = 3266.6
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
0.256
95% HDI
Param. Value
beta[20]
Param. Value
Iterations
Autocorrelation
ESS = 2331.4
Lag
shrink factor
median
97.5%
last iteration in chain
Density
MCSE =
1.12
95% HDI
Param. Value

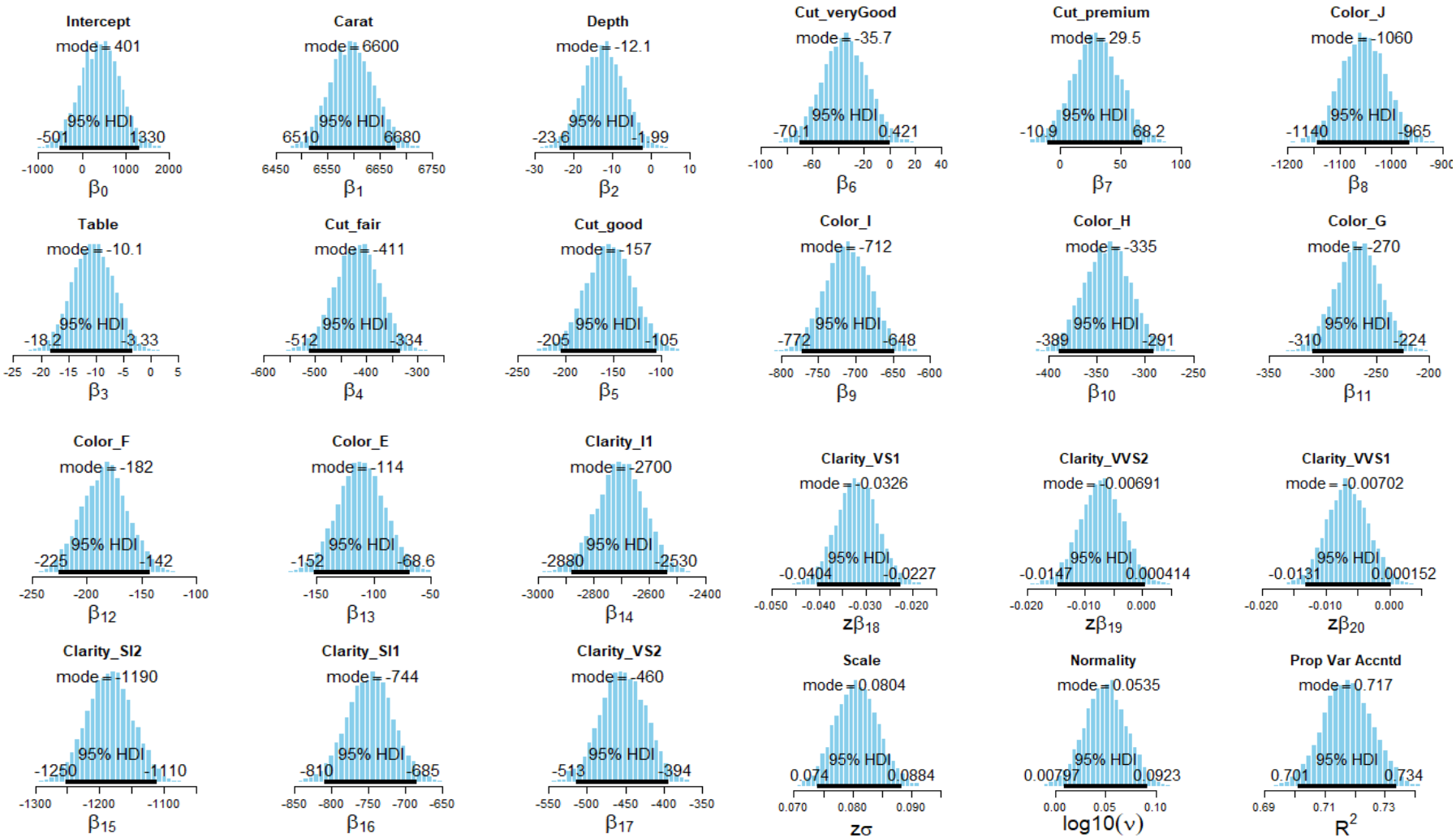


5.4 Prediction Distribution

In this section, the prediction distribution could be seen clearly. It seems that all of predictive diamond price are reasonable, and the prediction distribution with a narrow 95% HDI interval. The R square seems good for each instance that around 0.72. All of parameters show the normal distribution which is good. For some of parameters, the 95% HDI interval include 0 with a high probability. But, the HDI interval is wide, we could not say they are insignificant for dependent attribute.

5.4.1 Prediction for instance c(0.75, 62.2, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)

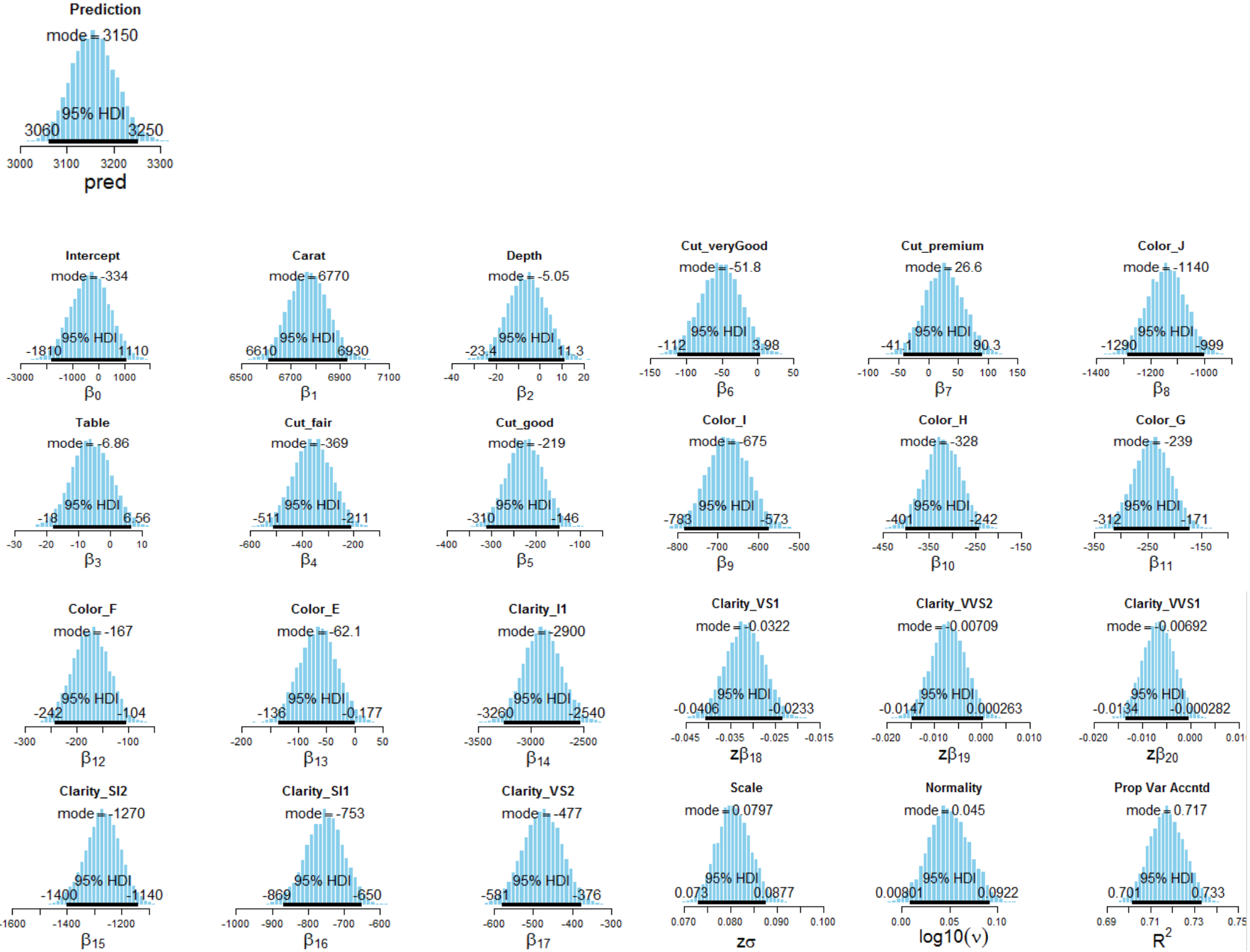




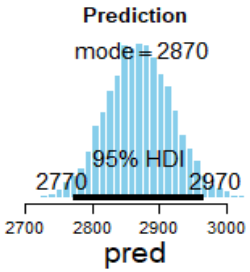
5.4.2 Prediction for instance c(0.86, 61, 58, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)

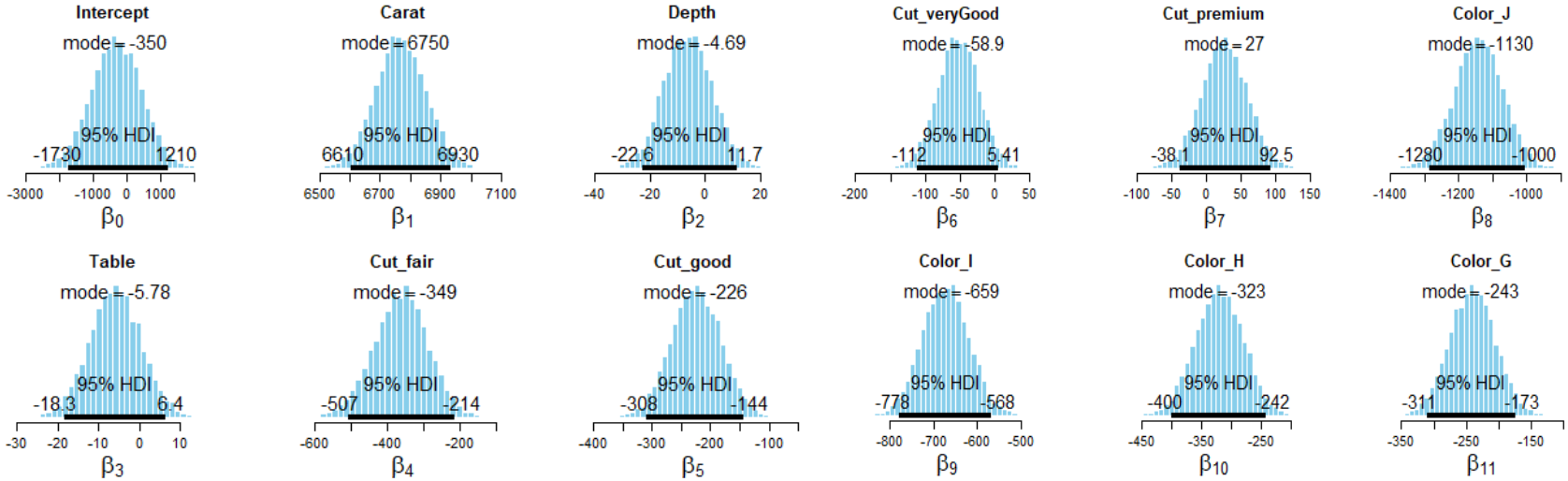

Prediction
mode = 3230
95% HDI
3140
3310
3100 3200 3300 3400
pred
Intercept
mode = -303
95% HDI
-1690
1190
-3000 -1000 0 1000
β0
Carat
mode = 6780
95% HDI
6610
6930
6500 6700 6900
β1
Depth
mode = -5.52
95% HDI
-22.8
11.1
-40 -20 0 10 20
β2
Cut_veryGood
mode = -51
95% HDI
-112
3.91
-150 -100 -50 0 50
β6
Cut_premium
mode = 30
95% HDI
-41.3
89.9
-100 -50 0 50 100 150
β7
Color_J
mode = -1150
95% HDI
-1290
-1000
-1400 -1200 -1000
β8
Table
mode = -4.93
95% HDI
-18.5
5.63
-30 -20 -10 0 10
β3
Cut_fair
mode = -350
95% HDI
-512
-211
-600 -400 -200
β4
Cut_good
mode = -226
95% HDI
-305
-141
-350 -250 -150
β5
Color_I
mode = -686
95% HDI
-778
-568
-800 -700 -600 -500
β9
Color_H
mode = -326
95% HDI
-404
-245
-500 -400 -300 -200
β10
Color_G
mode = -241
95% HDI
-312
-173
-350 -250 -150
β11
Color_F
mode = -169
95% HDI
-244
-105
-300 -200 -100
β12
Color_E
mode = -60.1
95% HDI
-131
2.89
-150 -100 -50 0 50
β13
Clarity_I1
mode = -2880
95% HDI
-3250
-2520
-3500 -3000 -2500
β14
Clarity_VS1
mode = -0.0322
95% HDI
-0.0414
-0.0235
-0.050 -0.040 -0.030 -0.020
zβ18
Clarity_VVS2
mode = -0.00738
95% HDI
-0.0148
0.000573
-0.020 -0.010 0.000
zβ19
Clarity_VVS1
mode = -0.00669
95% HDI
-0.0137
-8.99e-0
-0.020 -0.010 0.000
zβ20
Clarity_SI2
mode = -1260
95% HDI
-1390
-1140
-1500 -1300 -1100
β15
Clarity_SI1
mode = -747
95% HDI
-857
-646
-1000 -900 -800 -700 -600
β16
Clarity_VS2
mode = -480
95% HDI
-574
-379
-600 -500 -400 -300
β17
Scale
mode = 0.081
95% HDI
0.0738
0.0883
0.070 0.080 0.090
zσ
Normality
mode = 0.0523
95% HDI
0.00758
0.0918
0.00 0.05 0.10 0.15
log10(ν)
Prop Var Accntd
mode = 0.717
95% HDI
0.702
0.734
0.69 0.71 0.73 0.75
R2

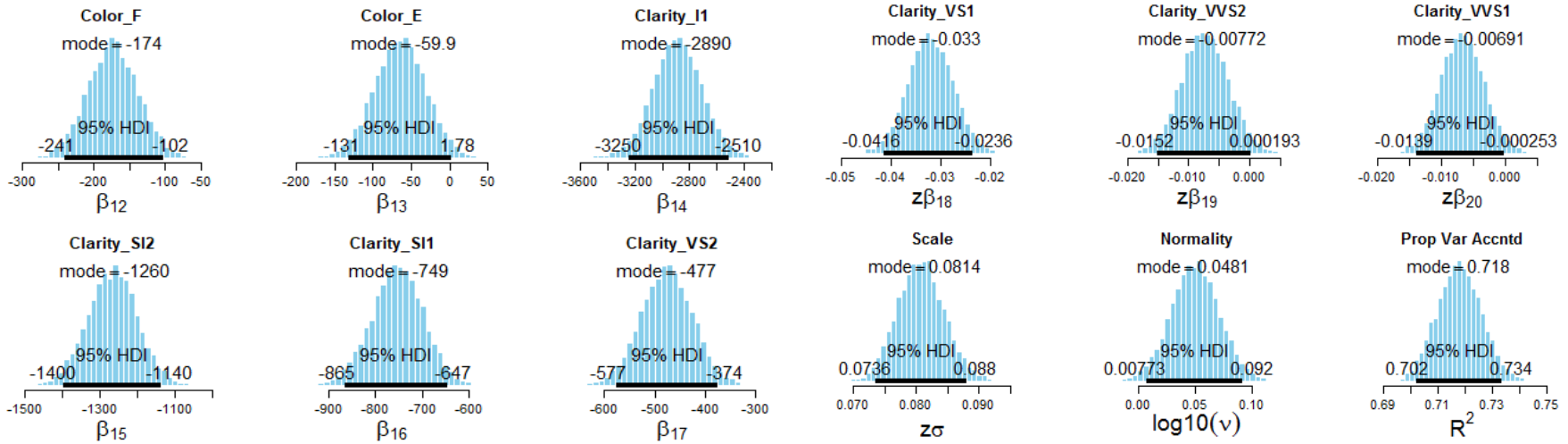
5.4.3 Prediction for instance c(0.7, 62.8, 60, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)



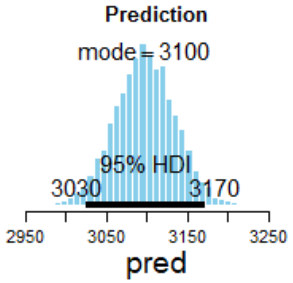
5.4.4 Prediction for instance c(0.72, 63.1, 55, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)

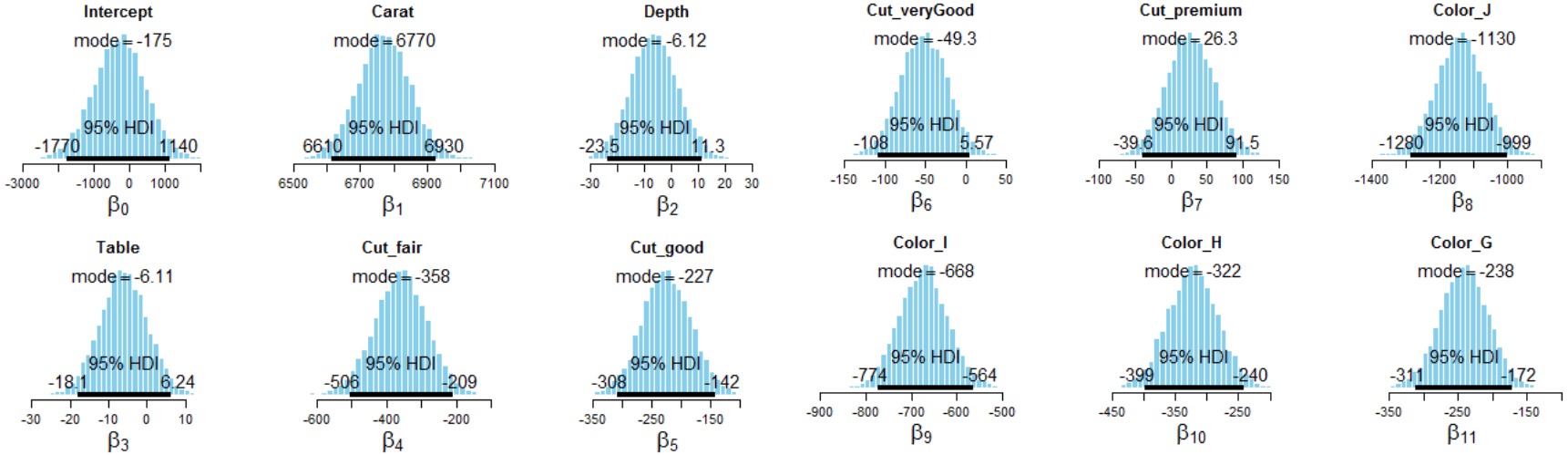


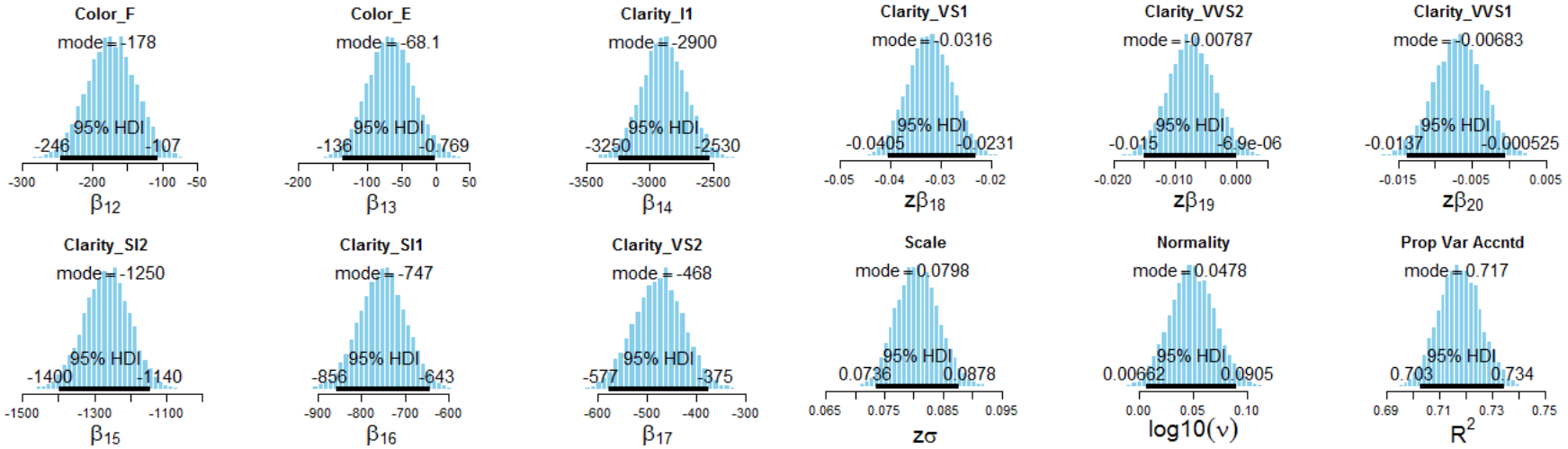




5.4.5 Prediction for instance c(0.72, 60.8, 57, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)







6. Comparison

y1 = 401 + 6600*beta[1] + (-12.1)*beta[2] + (-10.1)*beta[3] + (-411)*beta[4] + (-157)*beta[5] + (-35.7)*beta[6] + 29.5*beta[7] + (-1060)*beta[8] + (-712)*beta[9] + (-335)*beta[10] + (-270)*beta[11] + (-182)*beta[12] + (-114) *beta[13] + (-2700)*beta[14] + (-1190)*beta[15] + (-744)*beta[16] + (-460)*beta[17] + (-382)*beta[18] + (-96)*beta[19] + (-111)*beta[20]

y2 = (-303) + 6780*beta[1] + (-5.52)*beta[2] + (-4.93)*beta[3] + (-350)*beta[4] + (-226)*beta[5] + (-51)*beta[6] + 30*beta[7] + (-1150)*beta[8] + (-686)*beta[9] + (-326)*beta[10] + (-241)*beta[11] + (-169)*beta[12] + (-60.1) *beta[13] + (-2880)*beta[14] + (-1260)*beta[15] + (-747)*beta[16] + (-480)*beta[17] + (-375)*beta[18] + (-96)*beta[19] + (-106)*beta[20]

y3 = (-334) + 6770*beta[1] + (-5.05)*beta[2] + (-6.86)*beta[3] + (-369)*beta[4] + (-219)*beta[5] + (-51.8)*beta[6] + 26.6*beta[7] + (-1140)*beta[8] + (-675)*beta[9] + (-328)*beta[10] + (-239)*beta[11] + (-167)*beta[12] + (-62.1) *beta[13] + (-2900)*beta[14] + (-1270)*beta[15] + (-753)*beta[16] + (-477)*beta[17] + (-378)*beta[18] + (-98)*beta[19] + (-109)*beta[20]

y4 = (-350) + 6750*beta[1] + (-4.69)*beta[2] + (-5.78)*beta[3] + (-349)*beta[4] + (-226)*beta[5] + (-58.9)*beta[6] + 27*beta[7] + (-1130)*beta[8] + (-659)*beta[9] + (-323)*beta[10] + (-243)*beta[11] + (-174)*beta[12] + (-59.9) *beta[13] + (-2890)*beta[14] + (-1260)*beta[15] + (-749)*beta[16] + (-477)*beta[17] + (-386)*beta[18] + (-107)*beta[19] + (-109)*beta[20]

y5 = -175 + 6770*beta[1] + (-6.12)*beta[2] + (-6.11)*beta[3] + (-358)*beta[4] + (-227)*beta[5] + (-49.3)*beta[6] + 26.3*beta[7] + (-1130)*beta[8] + (-668)*beta[9] + (-322)*beta[10] + (-238)*beta[11] + (-178)*beta[12] + (-68.1)*beta[13] + (-2900)*beta[14] + (-1250)*beta[15] + (-747)*beta[16] + (-468)*beta[17] + (-378)*beta[18] + (-108)*beta[19] + (-107)*beta[20]

Table1: Prediction diamond price for test instance

carat	cut	color	clarity	depth	table	price	prediction(mode)	Error Rate
0.75	Ideal	D	SI2	62.2	55	2757	2820	2%
0.86	Premium	H	SI2	61	58	2757	3230	17%
0.7	Very Good	D	SI1	62.8	60	2757	3150	14%
0.72	Good	D	SI1	63.1	55	2757	2870	4%
0.72	Ideal	D	SI1	60.8	57	2757	3100	12%

Comparing with 5 test instances, coefficients are similar with each other. Based on criteria for each attribute, all of them make sense, the worst one given a high weight but negative, the negative value for better one getting smaller or change to positive. According to table1, it shows that all of predictive diamond price are reasonable especially for instance 1 and 4 that got the lower error rate. We might apply these two formulas to predict the diamond price on the other test dataset.

7. Summary

Through this report, how applied Bayesian statistic work with real dataset could be seen clearly. Moreover, the big idea for this report is multiple linear regression. Compare with classical statistic, instead of exact one estimate value, Bayesian gives us the distribution of predictive variable and each parameter, which more make sense. And we have not much limitation with basic assumption; it means that more distribution could be played with dataset. In fact, we just tried non-informative for prior knowledge. In real life, doing conversation with expert to get more informative knowledge is important, applying expert knowledge into models, the better result might come up for us.