Real Estate Application of Bayesian Linear Regression

Location: New Taipei City, Taiwan

Genevieve Anderson, Gabriella Nina, Sidney Papas, Emma Pullen

December 13, 2022

Data Exploratory Analysis

About The Dataset

In this project we will be looking at an application of bayesian linear regression relating to real estate prices in New Taipei City Taiwan in 2013. New Taipei City is the economic, political, educational and cultural center of Taiwan and one of the major hubs in East Asia. For this reason it is a desirable place for people in Taiwan to live. Taipei is part of a major high-tech industrial area where Mass Rapid Transit (MRT) connects Taipei with all parts of the island. We will be studying the realtionship between cost of living and other predictor variables.

Variables:

1. **Transaction date (for example, 2013.250=2013 March, 2013.500=2013 June, etc.)**
2. **House age (unit: year)**
3. **Distance to the nearest MRT station (unit: meter)**
4. **Number of convenience stores in the living circle on foot (integer)**
5. **Geographic coordinate, latitude. (unit: degree)**
6. **Geographic coordinate, longitude. (unit: degree)**
7. **House price of unit area: price per ping (1 ping = 35.5 sqft)**
8. **No: categorization of each home**

From the transaaction date, we were able to deduce that the data in this dataset was from 2013. Similarly we used the longitutde and latitude variables to pinpoint the location that this dataset was based on. From there we researched the unit of measurement (ping) that homes are measured in to better understand the relationship between cost per size of home (1 ping = 35.5 sqft).

From the variables provided we selected 3 predictor variables to perform a Bayesian Linear Regression. In order to make the dataset clean and efficient to work with we removed variables that would not be utilized.

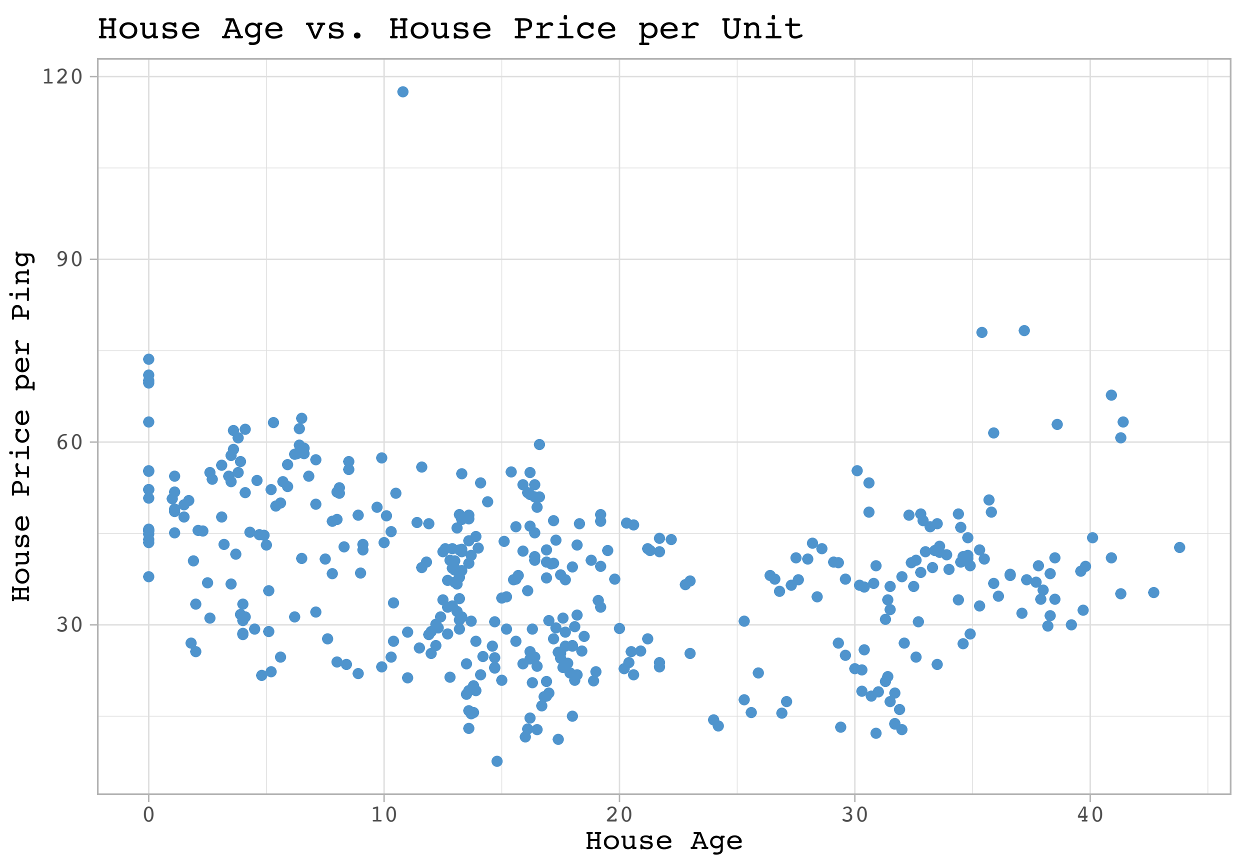
```
realestate1 <- realestate %>%
  select(-X6.longitude, -X5.latitude)
head(realestate1)
```

```
##   No X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station
## 1  1            2012.917         32.0                               84.87882
## 2  2            2012.917         19.5                              306.59470
## 3  3            2013.583         13.3                              561.98450
## 4  4            2013.500         13.3                              561.98450
## 5  5            2012.833          5.0                              390.56840
## 6  6            2012.667          7.1                             2175.03000
##   X4.number.of.convenience.stores Y.house.price.of.unit.area
```

```
## 1                                 10                       37.9
## 2                                  9                       42.2
## 3                                  5                       47.3
## 4                                  5                       54.8
## 5                                  5                       43.1
## 6                                  3                       32.1
```

Data Visualization for Exploratory Analysis:

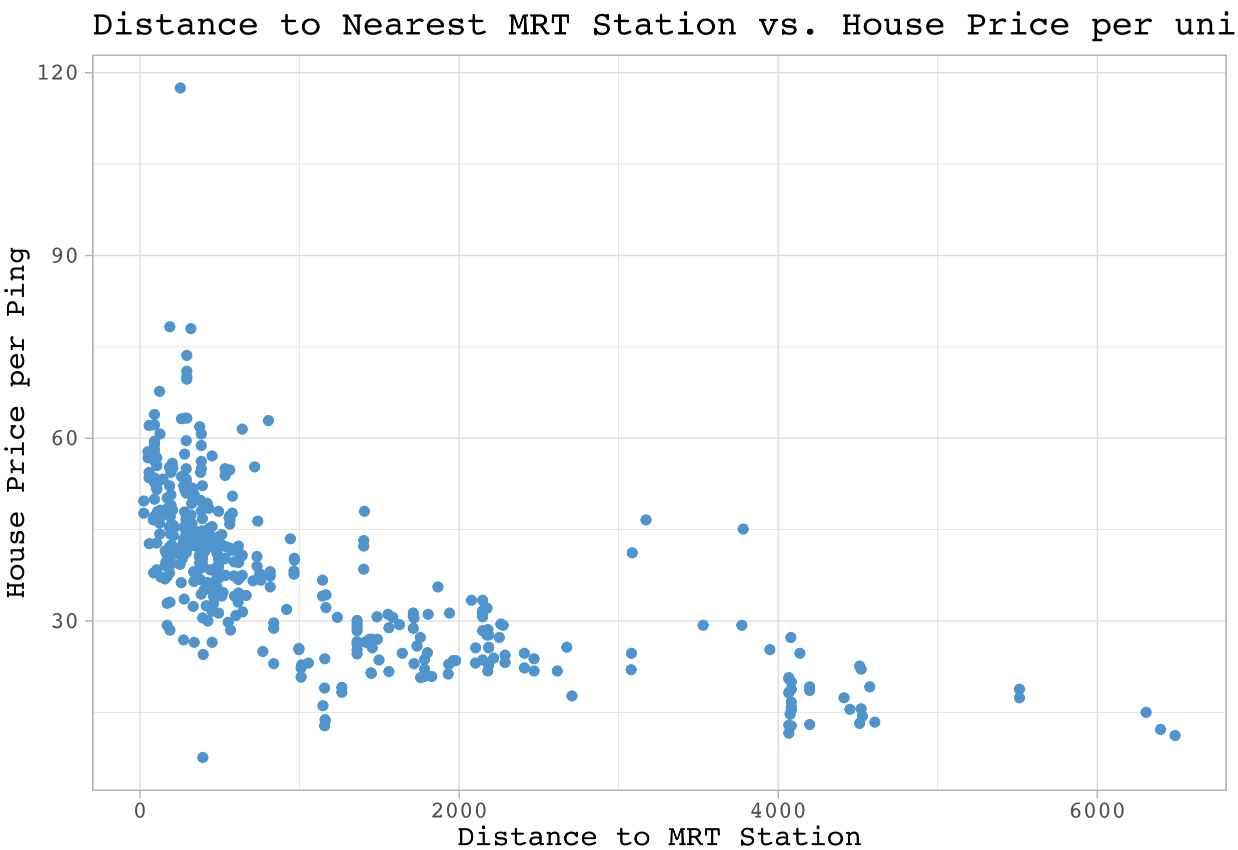
```
ggplot(realestate, aes(x=X2.house.age, y = Y.house.price.of.unit.area))+
  geom_point(color="steelblue3",size=1.3)+
  theme_light()+ggtitle("House Age vs. House Price per Unit")+
  xlab("House Age")+ylab("House Price per Ping") + theme(text=element_text(family = "mono"))
```



House Age Scatter plot: Based on the scatterplot, we can see that most homes were built between 10 and 20 years ago. We can also see a very slight correlation between the age of the house and the price of the house as the younger the house is, the more expensive the home is. However, since the correlation is so slight, we can not say that there is a major effect on the price of the house based on age.

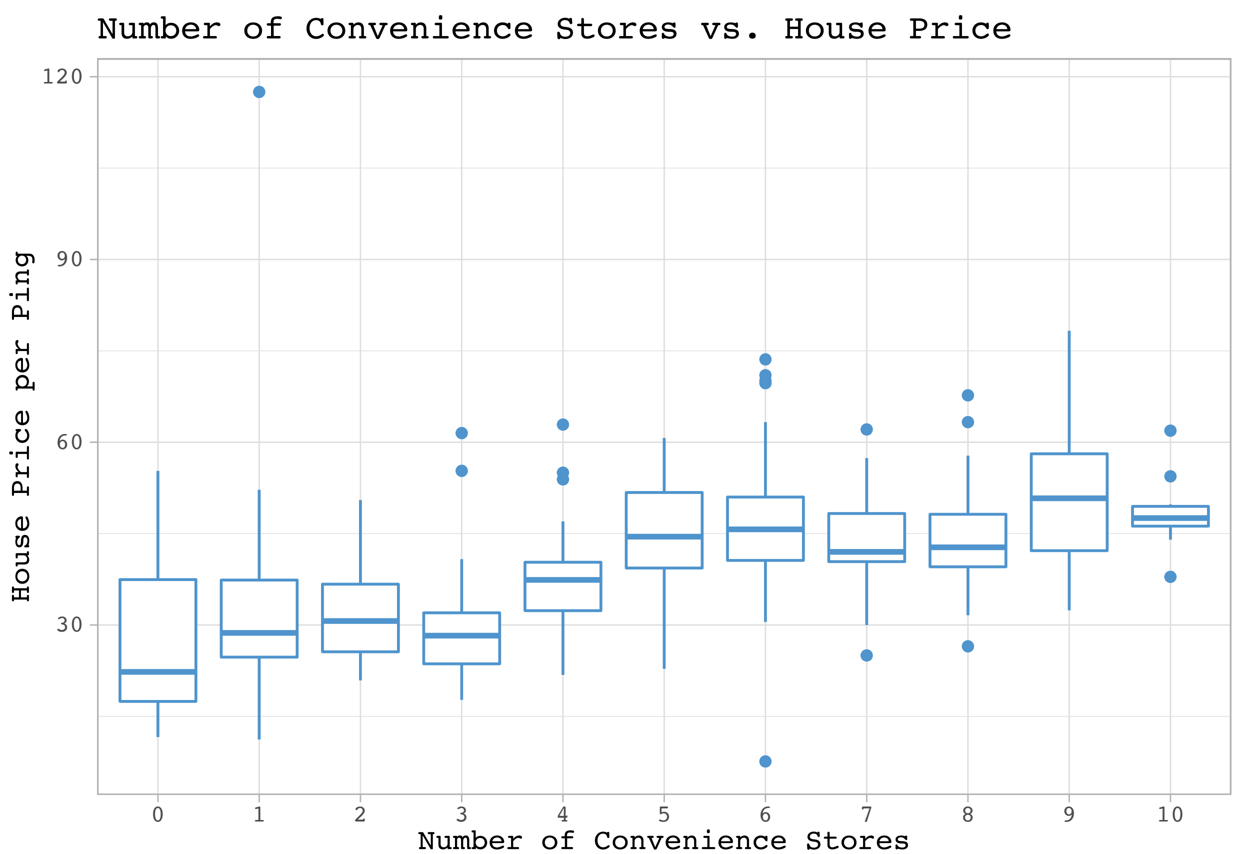
```
ggplot(realestate, aes(x=X3.distance.to.the.nearest.MRT.station, y = Y.house.price.of.unit.area))+
  geom_point(color="steelblue3",size=1.3)+ theme_light()+
```

```
  ggtitle("Distance to Nearest MRT Station vs. House Price per unit")+
  xlab("Distance to MRT Station")+ylab("House Price per Ping")+
  theme(text=element_text(family = "mono"))
```



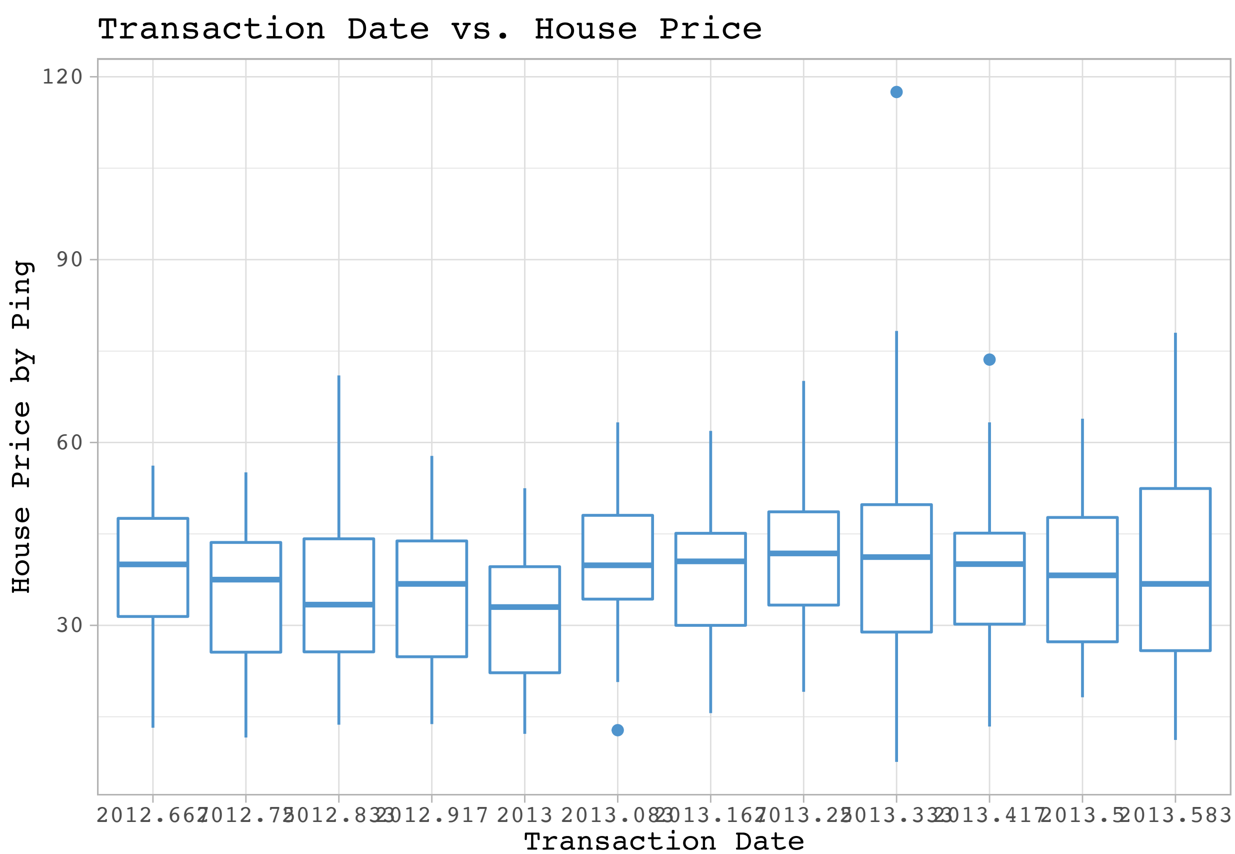
MRT Station Plot:Based on this plot, we can see that many of the homes are within a short distance of MRT stations. Additionally, we can see a strong correlation between the distance to MRT stations and house prices. As the distance increases, the price of the home decreases. This clearly makes sense as people want to live close to transportation systems in order to travel to different parts of the city and other areas of Taiwan. Therefore, it makes sense that close proximity to an MRT station would increase the price of the home.

```
ggplot(realestate, aes(x=as.factor(X4.number.of.convenience.stores), y = Y.house.price.of.unit.area))+
  geom_boxplot(color="steelblue3")+theme_light()+
  ggtitle("Number of Convenience Stores vs. House Price")+
  xlab("Number of Convenience Stores")+
  ylab("House Price per Ping")+theme(text=element_text(family = "mono"))
```



Convenience Store Boxplot: Based on the boxplot, we can see that there is a correlation betweeen the number of convenience stores within a certain distance of the home and the price of the house. As there are more convenience stores nearby, the price of the home increases. Excluding obvious outliers like (1,118), we can conclude that the number of convenience stores effects the price of the home.

```
realestate%>%
ggplot( aes(x =as.factor(X1.transaction.date) ,y = Y.house.price.of.unit.area)) +
  geom_boxplot(color="steelblue3") +
  theme_light() +
  labs(title = "Transaction Date vs. House Price ", x = "Transaction Date", y = "House Price by Ping")+
  theme(text=element_text(family = "mono"))
```



Transaction Date Boxplot: This plot compares the transaction date of the purchase of the home and the price of the home. Here, there is clearly not much of a difference in the price of the homes based on when the home was purchased. We can therefore assume that the real estate market was pretty steady from 2012-2013 in Taiwan. For this reason, we will not be regressing the house price on this variable.

Performing Simple Bayesian Linear Regression

Single linear regression on Price per unit and house age:

```
modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
}

## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

```r
y <- as.vector(realestate1$Y.house.price.of.unit.area)
x <- as.vector(realestate1$X2.house.age)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 0.0001,
                 "mu1" = 0, "g1" = 0.0001,
                 "a" = 1, "b" = 1)

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}
```

Pass the data and hyperparameter values to JAGS:

```r
model.houseage <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

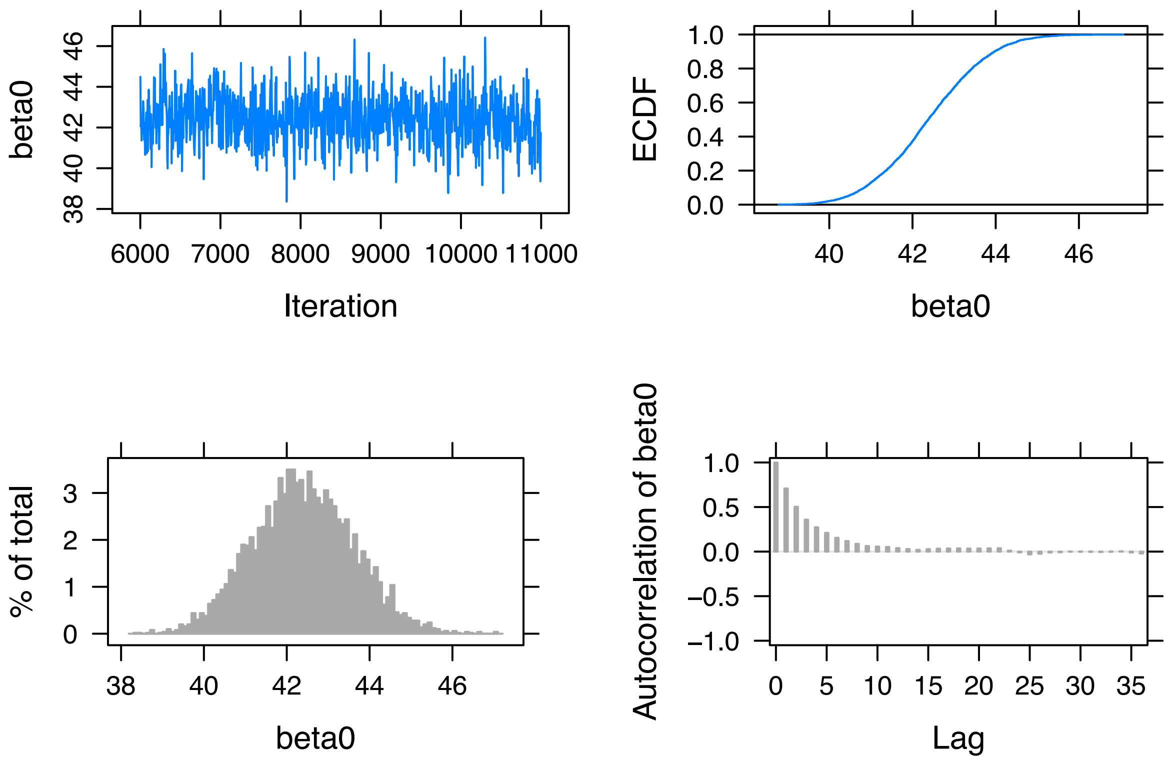
JAGS model for house price based on age of house

```
## Calling the simulation...
## Welcome to JAGS 4.3.1 (official binary) on Mon Dec 12 19:49:18 2022
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 414
##    Unobserved stochastic nodes: 3
##    Total graph size: 1314
## . Reading parameter file inits1.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
```

```
## . . . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Finished running the simulation
```

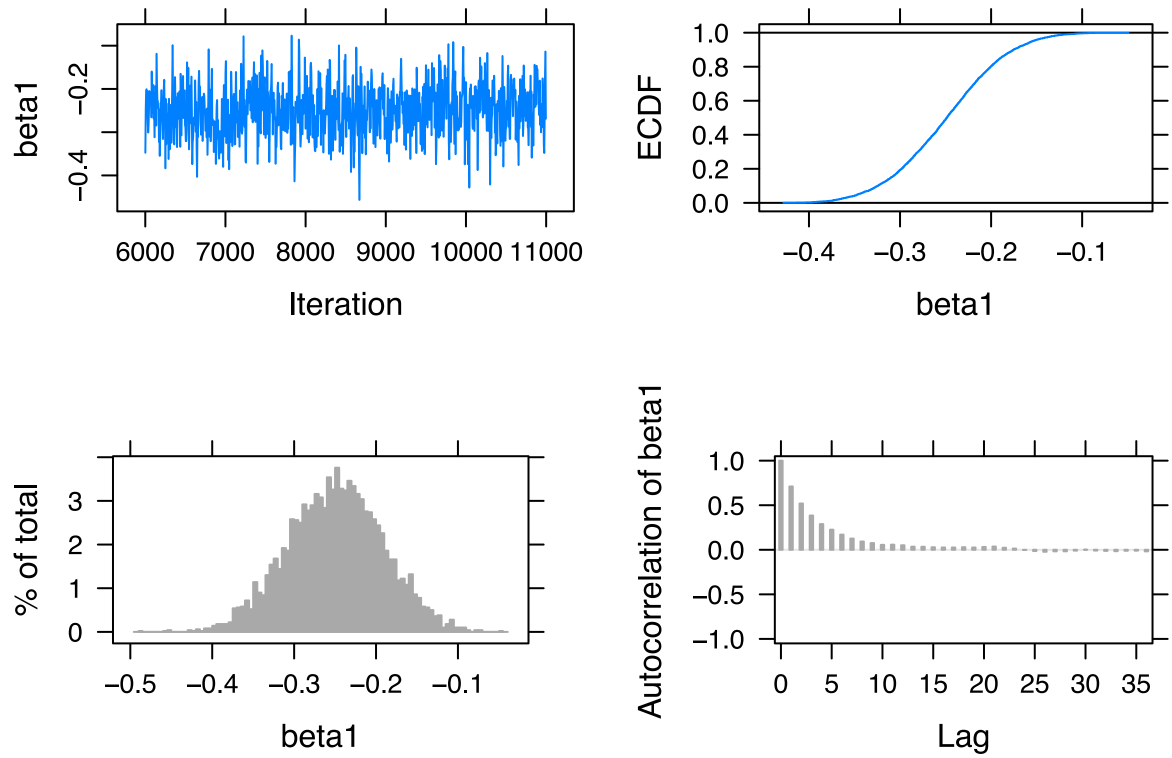
```r
plot(model.houseage, vars = "beta0")
```

```
## Generating plots...
```



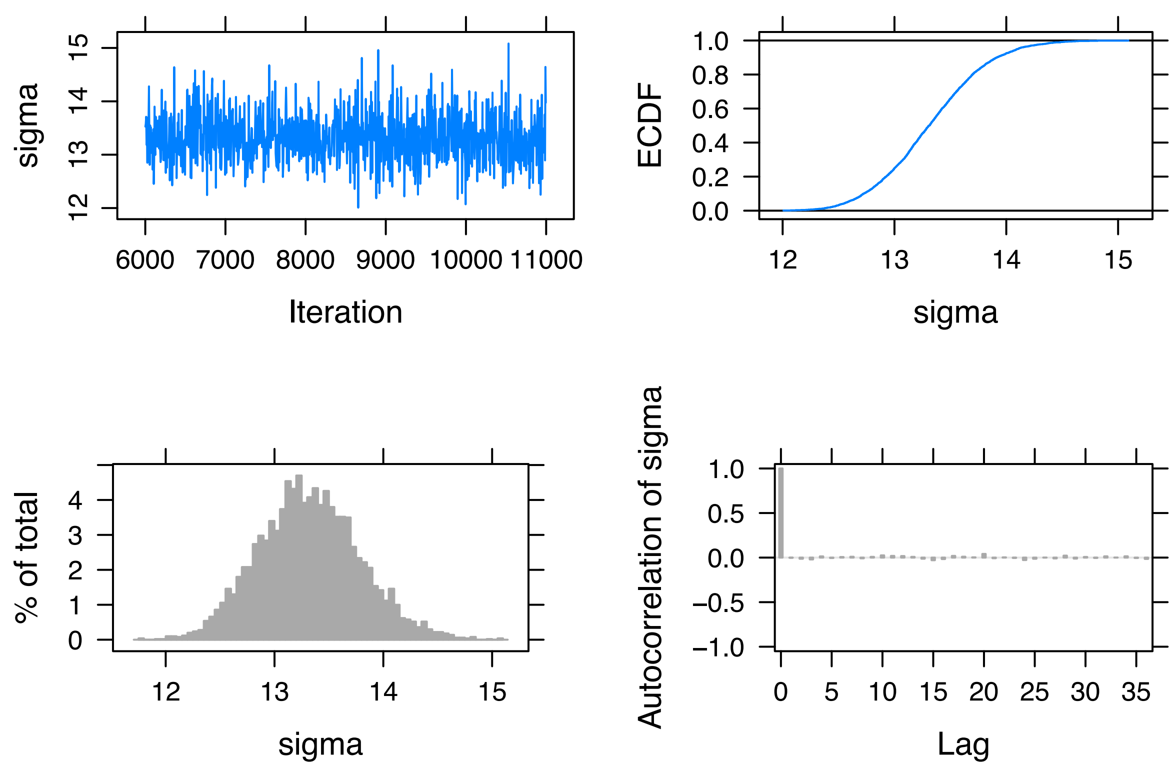

```r
plot(model.houseage, vars = "beta1")
```

```
## Generating plots...
```




```
plot(model.houseage, vars = "sigma")
```

```
## Generating plots...
```



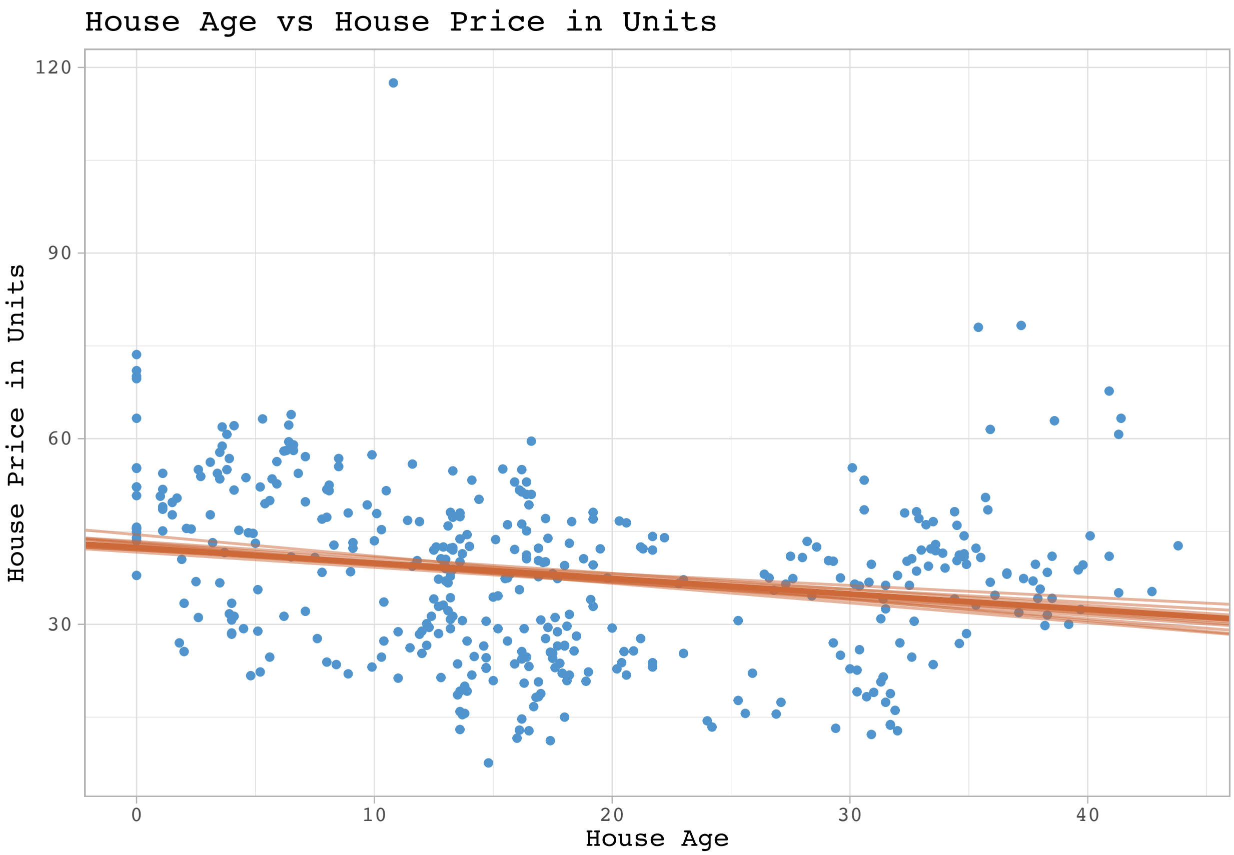
```r
summary(model.houseage)
```

```
##          Lower95   Median   Upper95       Mean         SD Mode       MCerr
## beta0 40.019000 42.37520 44.708500 42.3980190 1.21477396   NA 0.041560731
## beta1 -0.367924 -0.24883 -0.141715 -0.2495308 0.05783029   NA 0.002048539
## sigma 12.425800 13.30835 14.223300 13.3195186 0.46309185   NA 0.006549108
##       MC%ofSD SSeff      AC.10 psrf
## beta0     3.4   854 0.05667065   NA
## beta1     3.5   797 0.05551066   NA
## sigma     1.4  5000 0.02501681   NA
```

Simulate fits from the regression model

```r
post <- as.mcmc(model.houseage)
post_means <- apply(post, 2, mean)
post <- as.data.frame(post)
```

```r
ggplot(realestate1, aes(X2.house.age, Y.house.price.of.unit.area)) +
  geom_point(color="steelblue3", size=1) +
  geom_abline(data=post[1:10, ],
              aes(intercept=beta0, slope=beta1), color = "sienna3" ,alpha = 0.5) +
  geom_abline(intercept = post_means[1],
              slope = post_means[2], size = 1, color="sienna3") +
  theme_light(base_size = 10, base_family = "") +
  ylab("House Price in Units") +
  xlab("House Age") +
  ggtitle("House Age vs House Price in Units") + theme(text=element_text(family = "mono"))
```



After creating the JAGS model. We are able to use the posterior means to construct a linear regression line to show the relaionship between house age and house price in units. We can see a negative slope that indicates as a house gets older its cost value of the house decreases.

Single linear regression on Price per unit and distance to station

```
modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
}

## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

```
y <- as.vector(realestate1$Y.house.price.of.unit.area)
x <- as.vector(realestate1$X3.distance.to.the.nearest.MRT.station)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 0.0001,
                 "mu1" = 0, "g1" = 0.0001,
                 "a" = 1, "b" = 1)

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}
```

Pass the data and hyperparameter values to JAGS:

```
model.stationdist <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

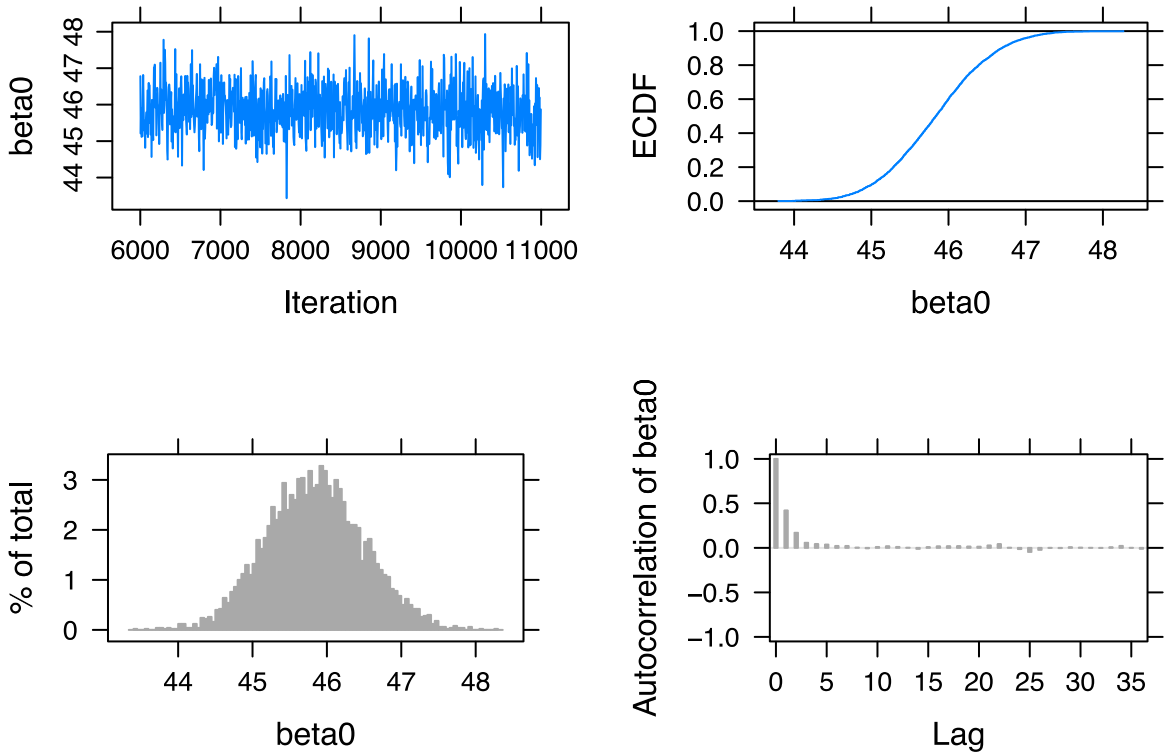
JAGS model for house price based on station distance

```
## Calling the simulation...
## Welcome to JAGS 4.3.1 (official binary) on Mon Dec 12 19:49:24 2022
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 414
##    Unobserved stochastic nodes: 3
##    Total graph size: 1360
## . Reading parameter file inits1.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 5000
```

```
## --------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Finished running the simulation
```

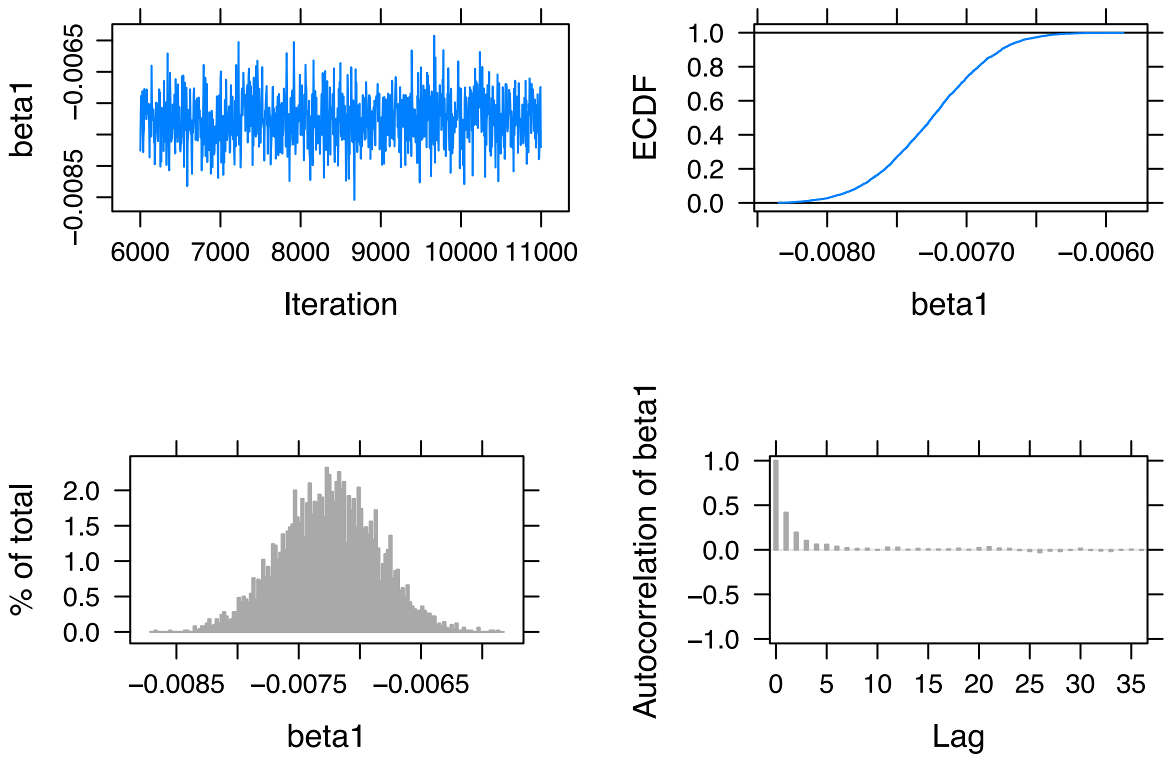
```
plot(model.stationdist, vars = "beta0")
```

```
## Generating plots...
```



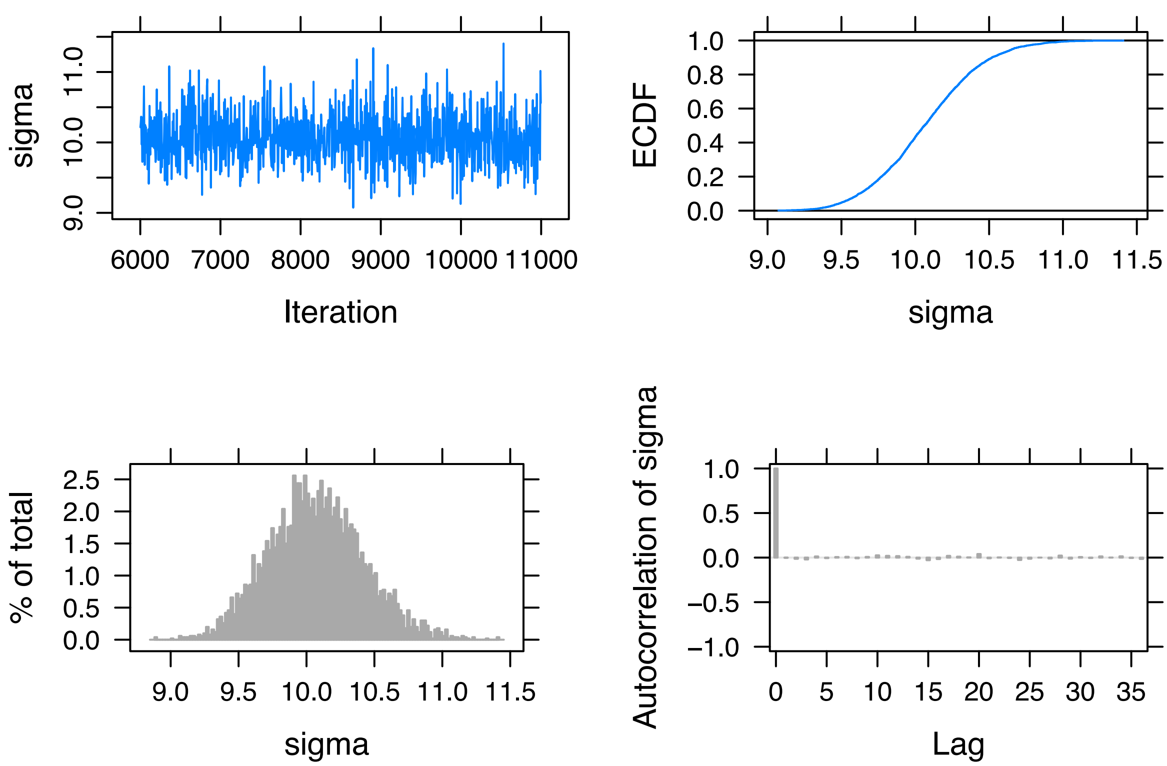

```
plot(model.stationdist, vars = "beta1")
```

```
## Generating plots...
```




```
plot(model.stationdist, vars = "sigma")
```

```
## Generating plots...
```



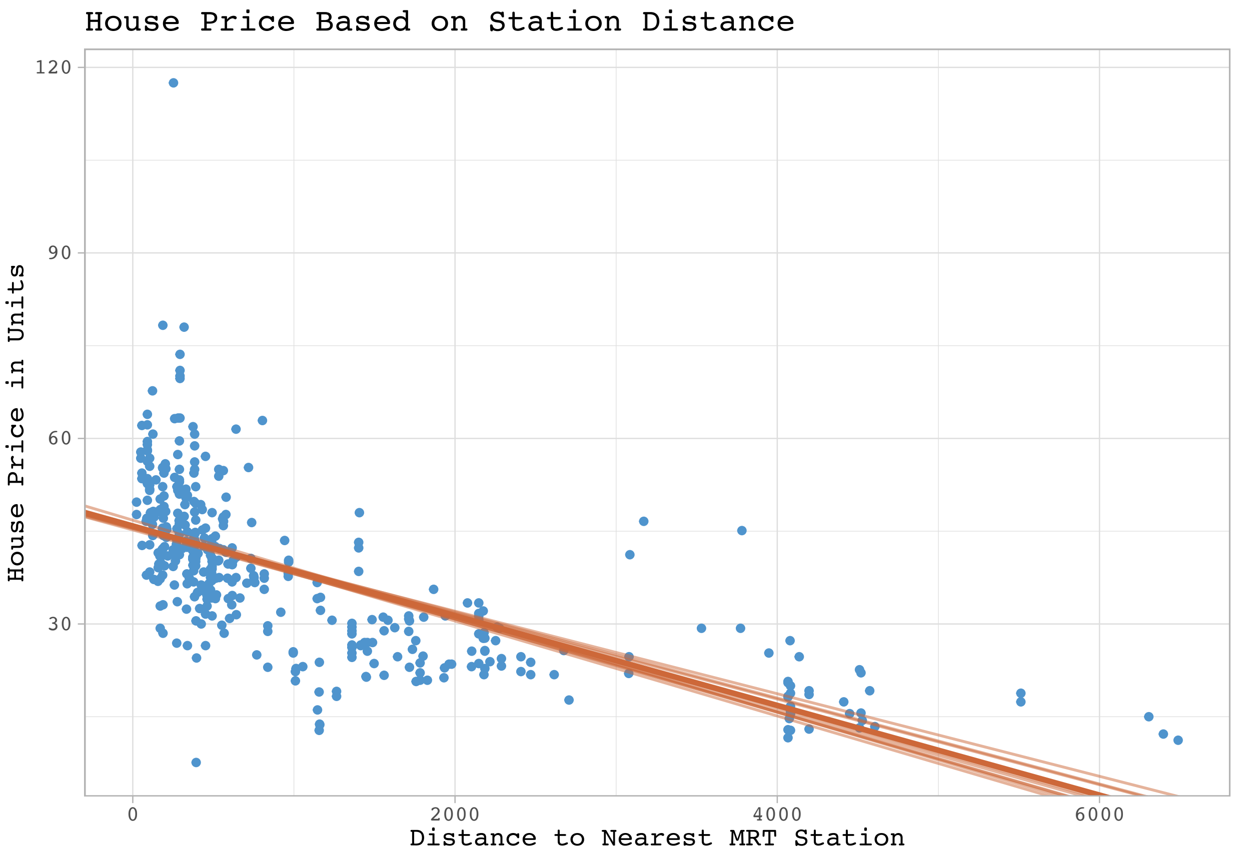
```
summary(model.stationdist)
```

```
##          Lower95      Median     Upper95        Mean           SD Mode
## beta0 44.58930000 45.836400000 47.11230000 45.840074900 0.6521706781   NA
## beta1 -0.00800519 -0.007248535 -0.00647906 -0.007252894 0.0003919241   NA
## sigma  9.39570000 10.061250000 10.75640000 10.070028870 0.3501931587   NA
##              MCerr MC%ofSD SSeff        AC.10 psrf
## beta0 1.443895e-02     2.2  2040  0.009084611   NA
## beta1 9.378371e-06     2.4  1746 -0.001024168   NA
## sigma 4.952479e-03     1.4  5000  0.025638965   NA
```

Simulate fits from the regression model

```
post <- as.mcmc(model.stationdist)
post_means <- apply(post, 2, mean)
post <- as.data.frame(post)
```

```
ggplot(realestate1, aes(X3.distance.to.the.nearest.MRT.station, Y.house.price.of.unit.area)) +
  geom_point(color="steelblue3",size=1) +
  geom_abline(data=post[1:10, ],
            aes(intercept=beta0, slope=beta1), alpha = 0.5, color="sienna3") +
  geom_abline(color="sienna3",intercept = post_means[1],
            slope = post_means[2], size = 1) +
  theme_light(base_size = 10, base_family = "") +
  ylab("House Price in Units") +
  xlab("Distance to Nearest MRT Station") +
  ggtitle("House Price Based on Station Distance") + theme(text=element_text(family = "mono"))
```



After creating the JAGS model. We are able to use the posterior means to construct a linear regression line to show the relaionship between the distance of the nearest MRT station and house price in units. Similar to the previous model we can see a negative slope that indicates the further the house is from the station the cost value of the house decreases.

Single linear regression on Price per unit and convinence stores

```
modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
}

## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

```r
y <- as.vector(realestate1$Y.house.price.of.unit.area)
x <- as.vector(realestate1$X4.number.of.convenience.stores)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 0.0001,
                 "mu1" = 0, "g1" = 0.0001,
                 "a" = 1, "b" = 1)

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}
```

Pass the data and hyperparameter values to JAGS:

```r
model.store <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

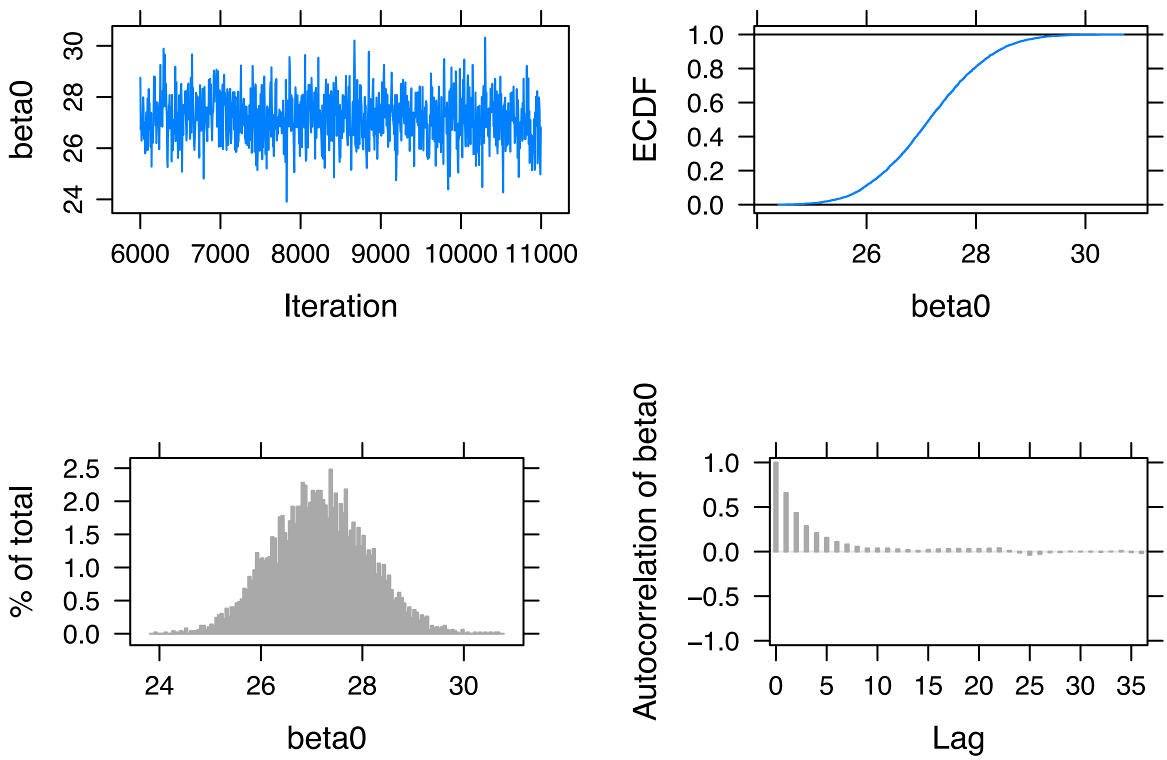
JAGS model for house price based on statin distance

```
## Calling the simulation...
## Welcome to JAGS 4.3.1 (official binary) on Mon Dec 12 19:49:30 2022
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 414
##    Unobserved stochastic nodes: 3
##    Total graph size: 864
## . Reading parameter file inits1.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
```

```
## . . . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Finished running the simulation
```

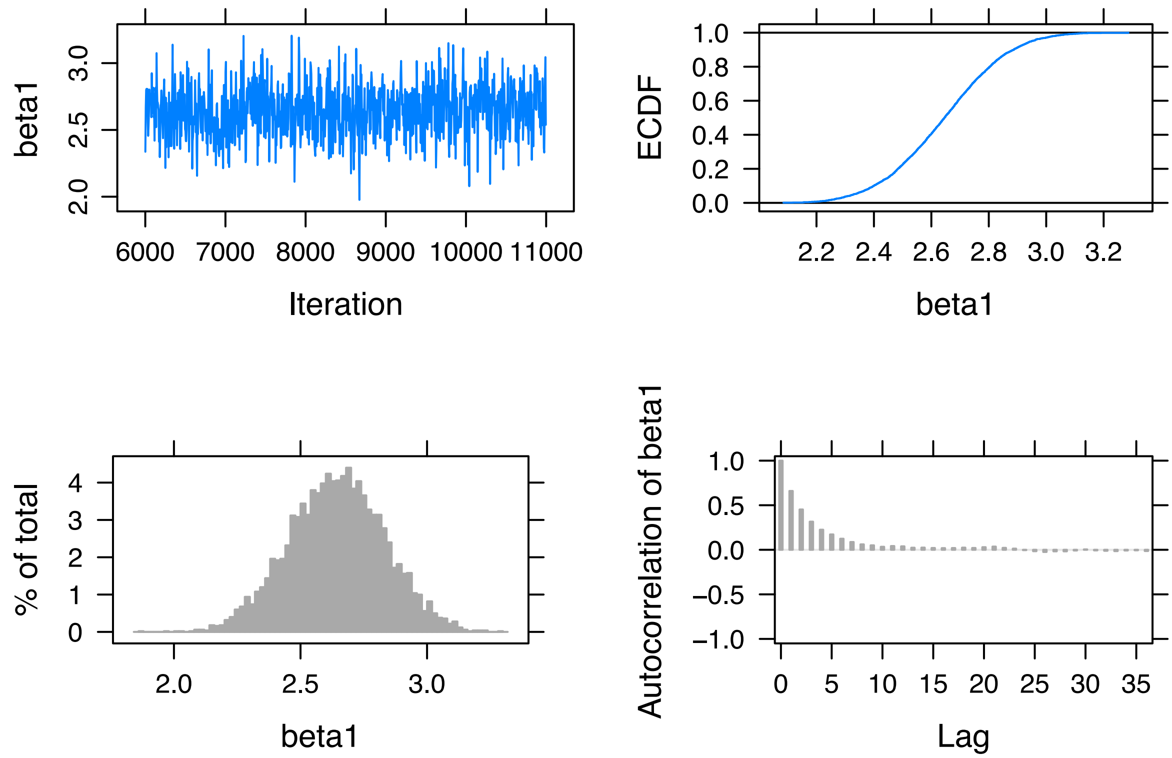
```r
plot(model.store, vars = "beta0")
```

```
## Generating plots...
```



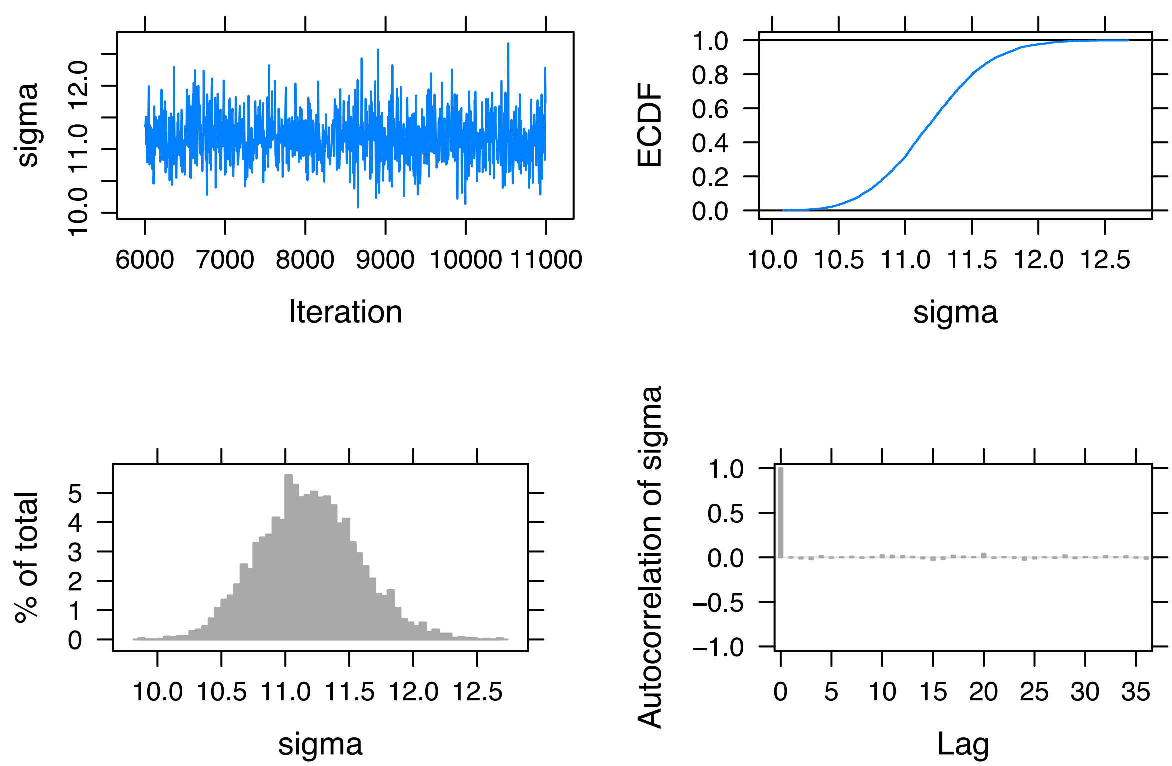

```r
plot(model.store, vars = "beta1")
```

```
## Generating plots...
```




```
plot(model.store, vars = "sigma")
```

```
## Generating plots...
```



```
summary(model.store)
```

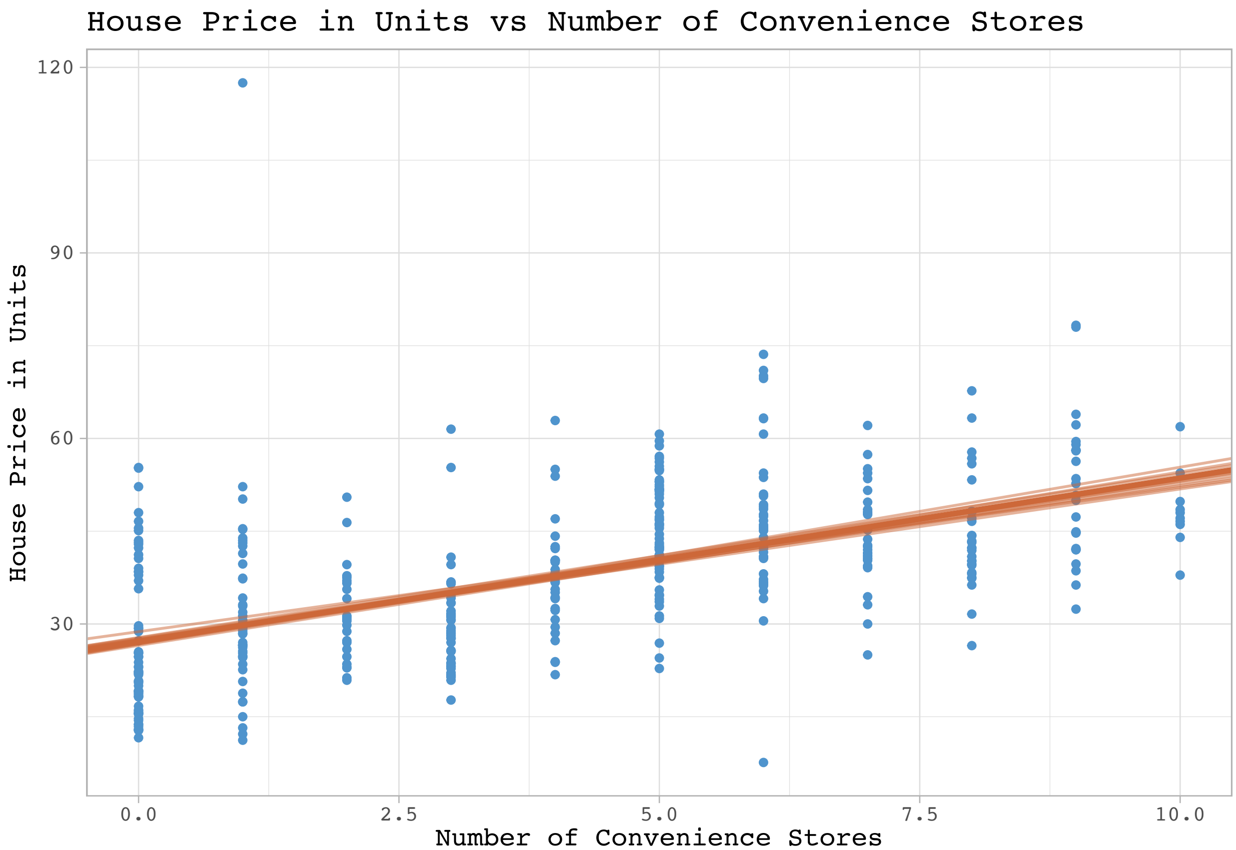
```
##       Lower95    Median  Upper95      Mean        SD Mode       MCerr MC%ofSD
## beta0 25.3089 27.144150 28.96290 27.157407 0.9430516   NA 0.029381239     3.1
## beta1  2.2830  2.645665  3.01633  2.643192 0.1873318   NA 0.006032710     3.2
## sigma 10.4362 11.176750 11.94370 11.185404 0.3888889   NA 0.005499719     1.4
##       SSeff      AC.10 psrf
## beta0  1030 0.03638949   NA
## beta1   964 0.03383639   NA
## sigma  5000 0.02504842   NA
```

Simulate fits from the regression model

```
post <- as.mcmc(model.store)
post_means <- apply(post, 2, mean)
post <- as.data.frame(post)
```

```
ggplot(realestate1, aes(X4.number.of.convenience.stores, Y.house.price.of.unit.area)) +
  geom_point(color="steelblue3",size=1) +
  geom_abline(color="sienna3",data=post[1:10, ],
              aes(intercept=beta0, slope=beta1), alpha = 0.5, color="sienna3") +
  geom_abline(color= "sienna3", intercept = post_means[1],
              slope = post_means[2], size = 1) +
  theme_light(base_size = 10, base_family = "") +
  ylab("House Price in Units") +
  xlab("Number of Convenience Stores") +
  ggtitle("House Price in Units vs Number of Convenience Stores") + theme(text=element_text(family = "m
```

```
## Warning: Duplicated aesthetics after name standardisation: colour
```



After creating the JAGS model. We are able to use the posterior means to construct a linear regression line to show the relaionship between the number of convenience stores and house price in units. We can see a positive slope that indicates as the number of convenience stores inscreases its cost value of the house increases.

A multiple linear regression, and MCMC simulation by JAGS

JAGS script for the MLR model

```
modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*x_house.age[i] + beta2*x_dist[i] +
beta3*x_store[i], invsigma2)
}
## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
beta2 ~ dnorm(mu2, g2)
beta3 ~ dnorm(mu3, g3)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

```r
y = as.vector(realestate1$Y.house.price.of.unit.area)
x_house.age = as.vector(realestate1$X2.house.age)
x_dist = as.vector(realestate1$X3.distance.to.the.nearest.MRT.station)
x_store = as.vector(realestate1$X4.number.of.convenience.stores)
N = length(y)  # Compute the number of observations
```

Pass the data and hyperparameter values to JAGS:

```r
the_data <- list("y" = y, "x_house.age" = x_house.age,
                 "x_dist" = x_dist, "x_store" = x_store,
                 "N" = N,
                 "mu0" = 0, "g0" = 1, "mu1" = 0, "g1" = 1,
                 "mu2" = 0, "g2" = 1, "mu3" = 0, "g3" = 1,
                 "a" = 1, "b" = 1)
```

Pass the data and hyperparameter values to JAGS:

```r
initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}
```

Pass the data and hyperparameter values to JAGS:

```r
posterior_MLR <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "beta2",
                                  "beta3", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

Run the JAGS code for this model:

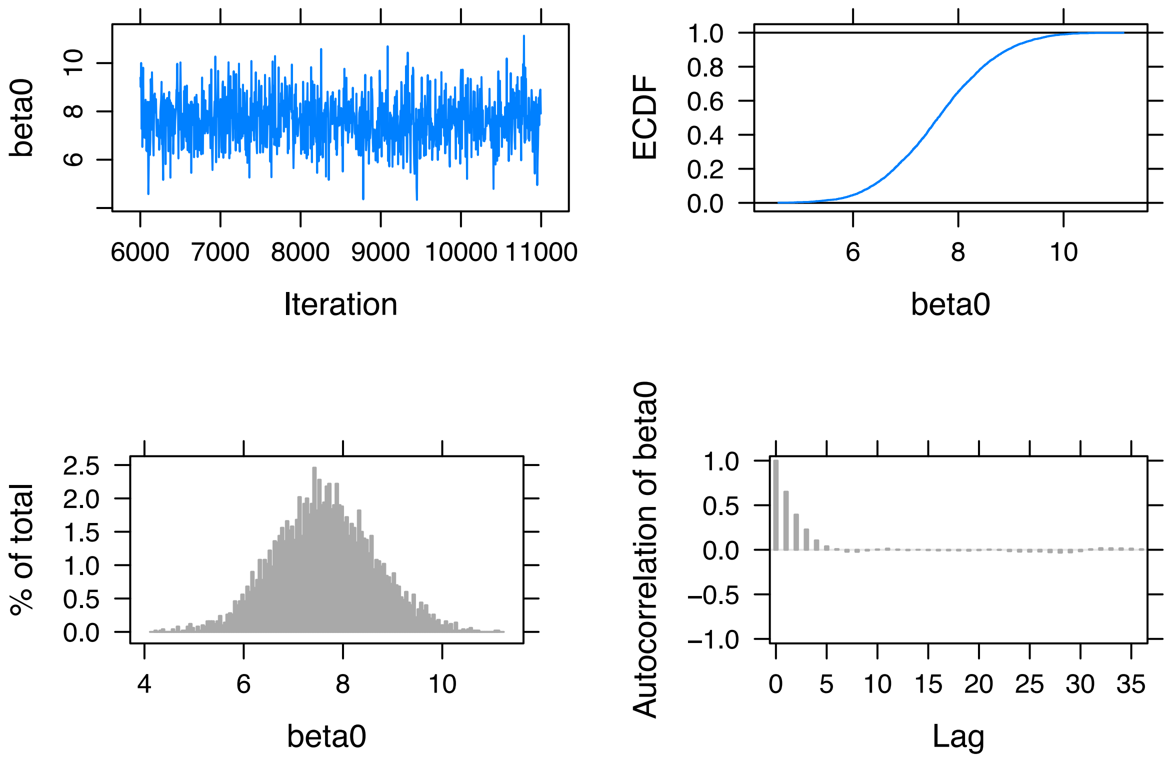
```
## Calling the simulation...
## Welcome to JAGS 4.3.1 (official binary) on Mon Dec 12 19:49:35 2022
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
```

```
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 414
##    Unobserved stochastic nodes: 5
##    Total graph size: 2551
## . Reading parameter file inits1.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...

## Warning: Convergence cannot be assessed with only 1 chain

## Finished running the simulation
```

```
plot(posterior_MLR, vars = "beta0")
```

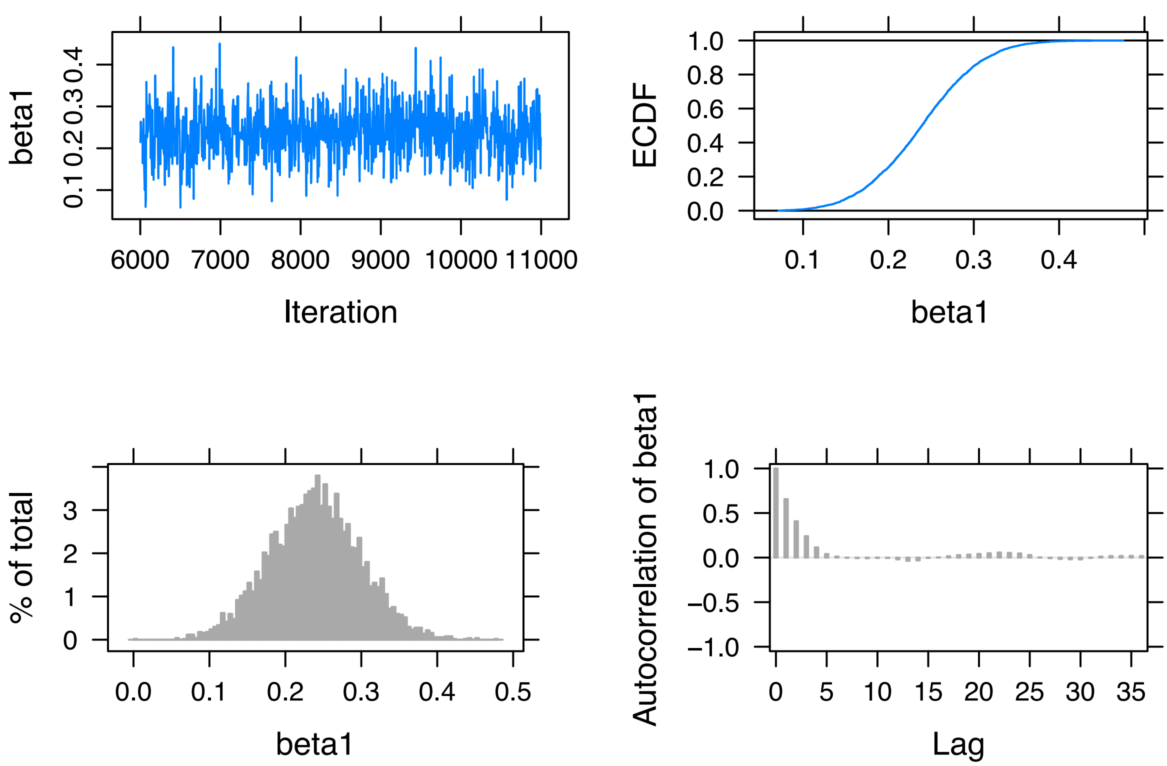
JAGS output for the MLR model

```
## Generating plots...
```



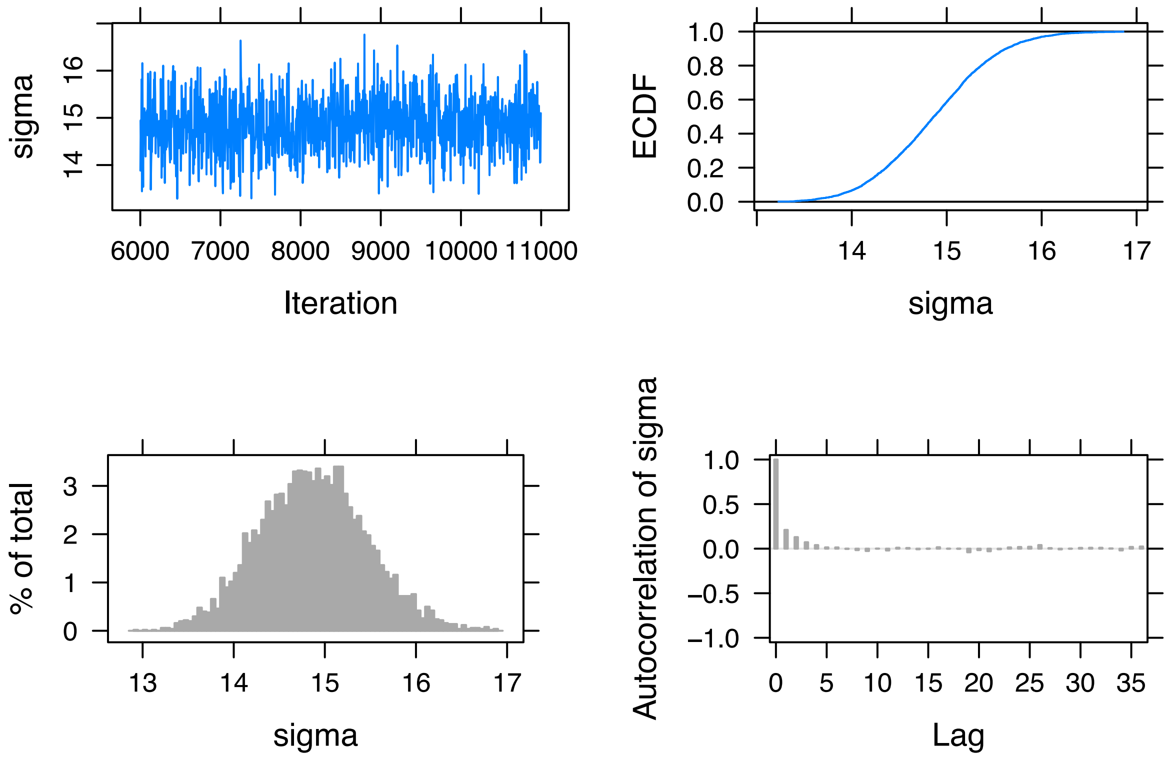

```
plot(posterior_MLR, vars = "beta1")
```

```
## Generating plots...
```



```r
plot(posterior_MLR, vars = "sigma")
```

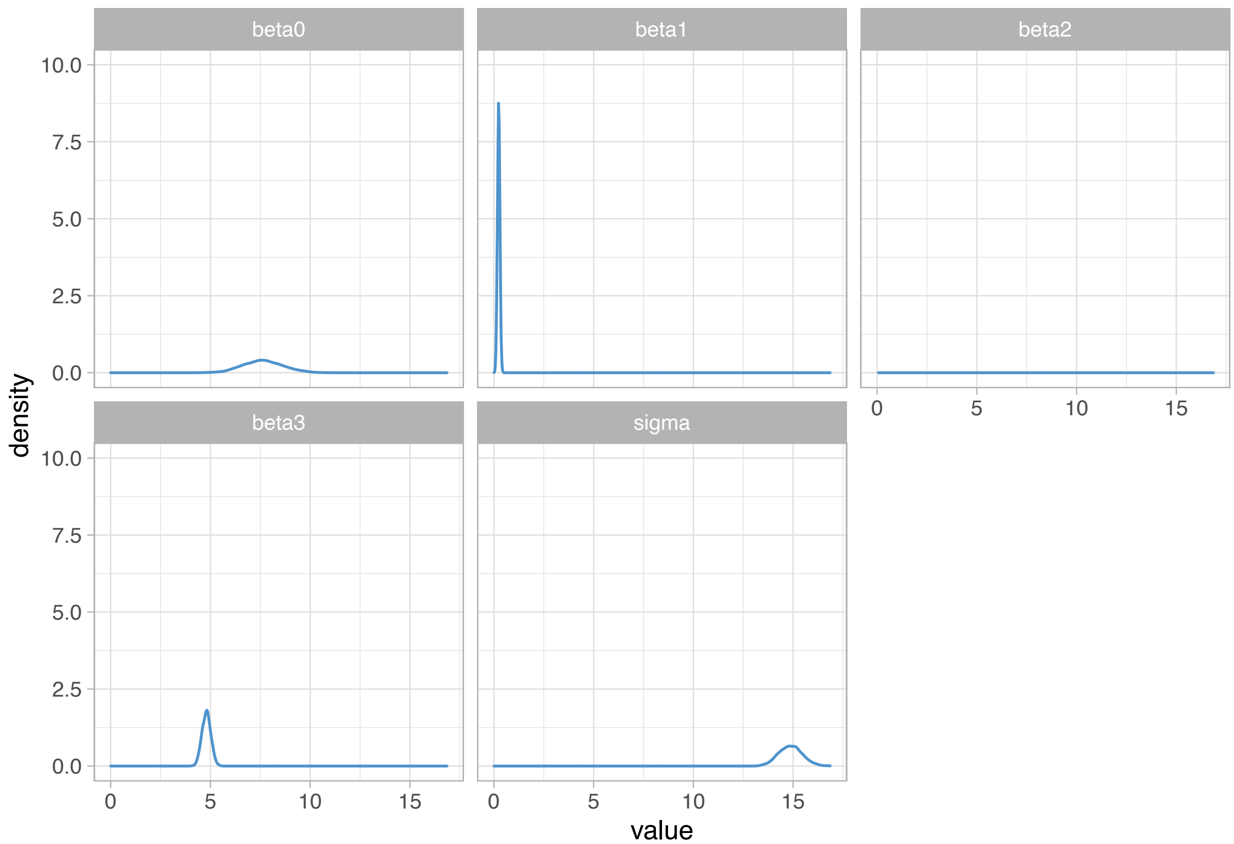
```
## Generating plots...
```

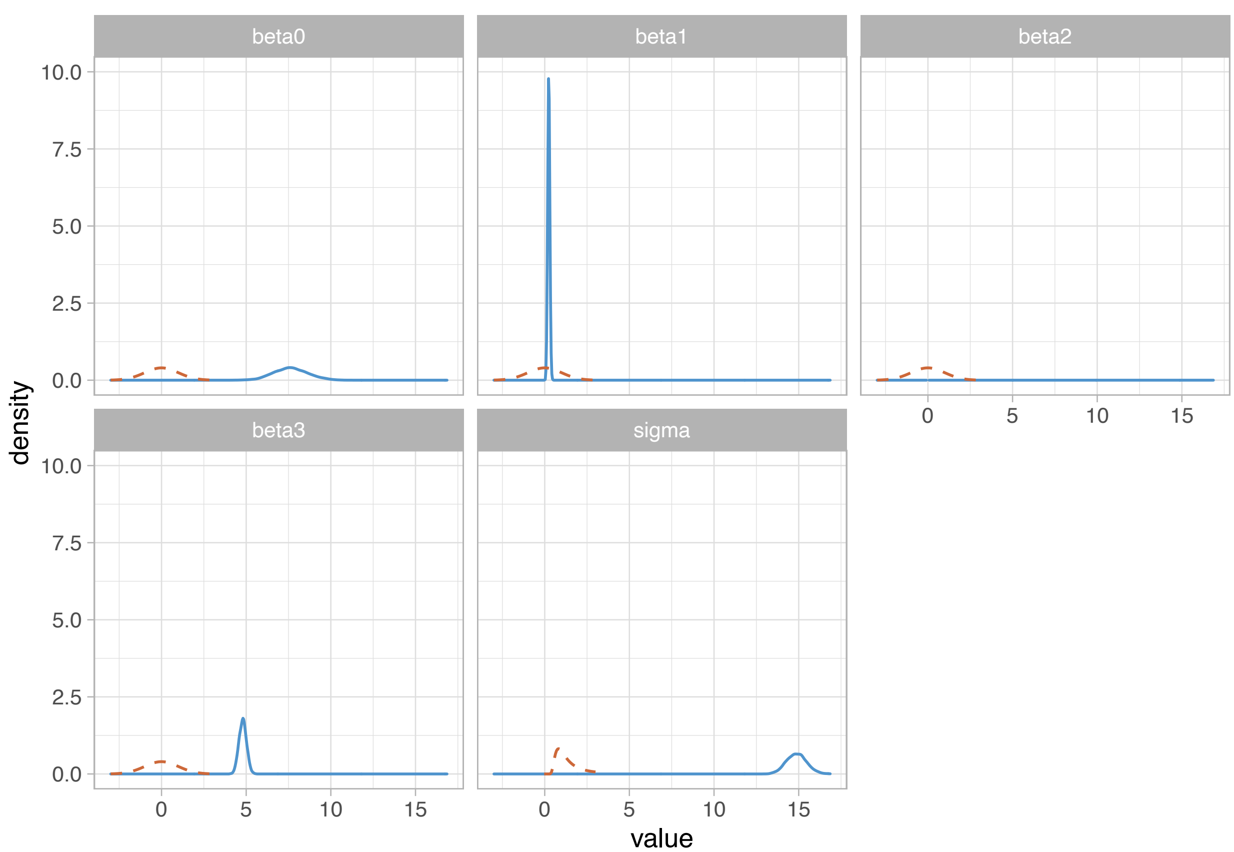



```r
summary(posterior_MLR)
```

```
##          Lower95      Median    Upper95        Mean           SD Mode
## beta0 5.80819000 7.611755000 9.68988000 7.631694720 0.9994540114   NA
## beta1 0.12552400 0.239208000 0.35947000 0.238950766 0.0597129873   NA
## beta2 0.00104491 0.002209215 0.00332674 0.002216513 0.0005813316   NA
## beta3 4.37546000 4.799070000 5.23731000 4.796362768 0.2221095687   NA
## sigma 13.72160000 14.861550000 16.03600000 14.869781040 0.5908957804   NA
##              MCerr MC%ofSD SSeff         AC.10 psrf
## beta0 2.758657e-02     2.8  1313  0.0041903158   NA
## beta1 1.669868e-03     2.8  1279 -0.0035750917   NA
## beta2 1.383778e-05     2.4  1765 -0.0334661500   NA
## beta3 6.082939e-03     2.7  1333  0.0062833835   NA
## sigma 1.167364e-02     2.0  2562  0.0007755715   NA
```

```r
post <- as.mcmc(posterior_MLR)
post %>% as.data.frame %>%
  gather(parameter, value) -> post2
ggplot(post2, aes(value)) +
  geom_density(color="steelblue3") + theme(text=element_text(family="mono"))+
  theme_light(base_size = 10, base_family = "") + facet_wrap(~ parameter, ncol = 3) + ylim(0,10)
```


beta0
beta1
beta2
beta3
sigma
density
value
10.0
7.5
5.0
2.5
0.0
0
5
10
15


beta0
beta1
beta2
beta3
sigma
density
value
10.0
7.5
5.0
2.5
0.0
0
5
10
15

In the above plots we can see a comparison between the beta values and how strongly the predictor variables impact our dependent variable. Please see the conclusion for an interpretation of the regression coefficients.

Beta Value Interpretation

Beta values are our regression coefficients and tell us how our independent/predictor variables impact our dependent variables. For our study, our dependent variable was the house cost per unit area. Our aim was to identify which predictor variables most impacted the cost of purchasing a home in New Taipei City, Taiwan. Since house sizes may differ dependent on location, the ability to use the house cost per unit area was valuable in making sound conclusions. The plots helped us to visualize the results while the posterior summaries clearly defined the explanation for the behavior of the graphs. With our beta values, we conclude that one unit increase in house age is associated an approximate 0.25 decrease in house cost per unit area. Additionally, one unit increase in distance to MRT Station denotes an approximate 0.00725 decrease in house cost per unit area. Finally, one unit increase in number of convenience stores denotes an approximate 2.6 increase in house cost per unit area, which supports our group's prediction prior to running the regression models. Therefore, we can deduce that people seeking to purchase homes in New Taipei City can anticipate an increase in cost as their distance to an MRT station decreases, as the convenience in the MRT station increases the value in the homes. After looking at the plots, we can also deduce that older houses to result in a decrease in their values, however there is not too strong of a regression. This is contrary to our original prediction as we all anticipated house age to have a stronger impact on the house cost. Therefore, house age may not play as high of a role in considerations for seeking homeowners. Finally, as expected, we see an increase in home value with higher numbers of convenience stores so New Taipei City residents and soon to be residents can anticipate higher costs of living as opposed to living in an area with fewer convenience stores.

Conclusions

Through our exploratory data analysis we were able to see correlation in house age, distance to MRT station, convenience stores, and no correlation in transaction date. When we began our simple linear regression, we were able to conclude that as a house gets older, its cost decreases. From our next simple linear regression model we were able to conclude that a houses value decreases the further it is from an MRT station. From our last simple linear regression model we were able to conclude that as the number of convenience stores near the house increase, the house value also increases. From these observations we are able to learn a lot about real estate and New Taipei City. From our results, it is clear that residents value convenience. Residents want to be near an MRT station and near convenience stores. Because of the demand for convenience, this drives home values up. This dataset also included variables for the longitude and latitude of the houses to provide their exact locations. If we were to conduct a further analysis on the cost of homes in New Taipei City, Taiwan, we would organize the longitude and latitude values to correspond to a region category (ex: Northeast, Northwest, Southeast, Southwest). With this information, we could perform another Bayesian Linear Regression to assess house location as an additional predictor variable when considering house cost per unit area in New Taipei City, Taiwan. Going forward, it would be interesting to perform these tests on a dataset from somewhere else in the world to see if they also had the same desire for convenience as the residents in this dataset did.

Posterior α Likelihood $\times$ Prior

Likelihood:

$p(Y|x_i, \beta, \sigma^2) = (2\pi\sigma^2) * \exp[(-\frac{1}{2\sigma^2}(Y - x_i\beta)^T(Y - x_i\beta))]$

Conditional prior on β:

$p(\beta|\sigma^2) = (2\pi\sigma^2)^{-P/2} * |\Lambda_0| * \exp[(-\frac{1}{2\sigma^2}(\beta - \mu_0)^T\Lambda_0(\beta - \mu_0))]$

Prior on σ^2:

$p(\sigma^2) = \frac{b_0^{a_0}}{\Gamma(a_0)} * (\sigma^2)^{-(a_0+1)} * \exp[-\frac{b_0}{\sigma^2}]$

Derivation:

When performing the calculation and multiplying all of the values, we can add the exponents. In doing so we can combine like terms and simplify:

$$
\begin{aligned}
&(Y - x_i\beta)^T(Y - x_i\beta)) + ((\beta - \mu_0)^T\Lambda_0(\beta - \mu_0)) \\
&= (Y - x_i\hat{\beta})^T(Y - x_i\hat{\beta}) + (\hat{\beta} - \beta)^T x_i^T x_i(\hat{\beta} - \beta) + (\beta - \mu_0)^T\Lambda_0(\beta - \mu_0) \\
&= Y^TY + \mu_0\Lambda_0\mu_0 - \mu_N^T\Lambda_N\mu_N + (\beta - \mu_N)^T\Lambda_N(\beta - \mu_N)
\end{aligned}
$$

Where:

$$
\begin{aligned}
\Lambda_N &= x_i^T x_i + \Lambda_0 \\
\mu_N &= \Lambda_0^- 1(\mu_0^T\Lambda_0 + x_i{}^TY)
\end{aligned}
$$

Then we can rewrite our posterior as:

$$
\begin{gathered}
\alpha\ (2\pi\sigma^2)^{-P/2}|\Lambda_0|^{1/2}\ \exp[-\frac{1}{2\sigma^2}[(\beta - \mu_N)^T\Lambda_N(\beta - \mu_N)]]\times \\
(2\pi\sigma^2)^{-N/2}\ \exp[-\frac{1}{2\sigma^2}[Y^TY + \mu_0\Lambda_0\mu_0 - \mu_N^T\Lambda\mu_N]]\times \\
\frac{b_0^{a_0}}{\Gamma(a_0)} * (\sigma^2)^{-(a_0+1)} * \exp[-\frac{b_0}{\sigma^2}]
\end{gathered}
$$

Since there is a P-variate normal distribution in the first line, we can ignore $(2\pi)^{-N/2}$ and the inverse-gamma prior normalizer and condense the bottom two lines to have:

$$
(\sigma^2)^{-(a_0+\frac{N}{2}+1)}\ \exp[-\frac{1}{\sigma^2}[b_0 + \frac{1}{2}\{Y^TY + \mu_0\Lambda_0\mu_0 - \mu_N^T\Lambda\mu_N\}]]
$$

Where:

$$
\begin{aligned}
a_N &= a_0 + \frac{N}{2} \\
b_n &= b_0 + \frac{1}{2}(Y^TY + \mu_0\Lambda_0\mu_0 - \mu_N^T\Lambda\mu_N)
\end{aligned}
$$

Therefore we can summarize our posterior distribution to be:

$$
\begin{gathered}
\boxed{p(\beta, \sigma^2|x_i, Y)\ \alpha\ p(\beta|x_i, Y, \sigma^2) \times p(\sigma^2|x_i, Y)} \\
\beta|x_i, Y, \sigma^2 \sim Normal(\mu_N, \sigma^2\Lambda_N^{-1}) \\
\sigma^2|x_i, Y \sim InverseGamma(a_N, b_N)
\end{gathered}
$$