Kc_house

ZhijianLiu

Data description

The dataset of interest contains house sale prices for King County. It includes homes sold data between May 2014 and May 2015. The description of the variables are listed as below:

variable.name	description
id	a notation for a house
date	Date house was sold
price	Price is prediction target
bedrooms	Number of Bedrooms/House
bathrooms	Number of bathrooms/bedrooms
sqft_living	square footage of the home
sqft_lot	square footage of the lot
floors	Total floors (levels) in house
waterfront	House which has a view to a waterfront
view	Has been viewed
condition	How good the condition is (Overall)
grade	overall grade given to the housing unit, based on King County grading system
sqft_above	square footage of house apart from basement
sqft_basement	square footage of the basement
yr_built	Built Year
yr_renovated	Year when house was renovated
zipcode	zip
lat	Latitude coordinate
long	Longitude coordinate
sqft_living15	Living room area in 2015(implies–some renovations). This might or might not have affected the lotsize are
sqft_lot15	lot Size area in 2015(implies– some renovations)

And a glimpse of data:

id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	conditi
7129300520	2014-10-13	221900	3	1.00	1180	5650	1	0	0	
6414100192	2014-12-09	538000	3	2.25	2570	7242	2	0	0	
5631500400	2015-02-25	180000	2	1.00	770	10000	1	0	0	
2487200875	2014-12-09	604000	4	3.00	1960	5000	1	0	0	
1954400510	2015-02-18	510000	3	2.00	1680	8080	1	0	0	
7237550310	2014-05-12	1225000	4	4.50	5420	101930	1	0	0	

I will fit a multilevel model to estimate the home sales price in King County. Before fitting the model, i take out some variables that seem not much relevant or redundant. Also some manipulation are applied on the variables of interest. The date that the house was sold and the year that the house was sold does not essentially influence the house price. Actually the age of the house affected the house price. So here i create a new variable call house_age by calculating the year between the sold date and built date.

```
df <- df %>%
  mutate(yr_sold = year(date)) %>%
```

```
  mutate(house_age = yr_sold - yr_built)
```

The year that the house was renovated should be relevant to house price. But this variable has so many 0 values and it makes no sense to be directly taken into the model, so i modify its value to 1 or 0, renovated or not.

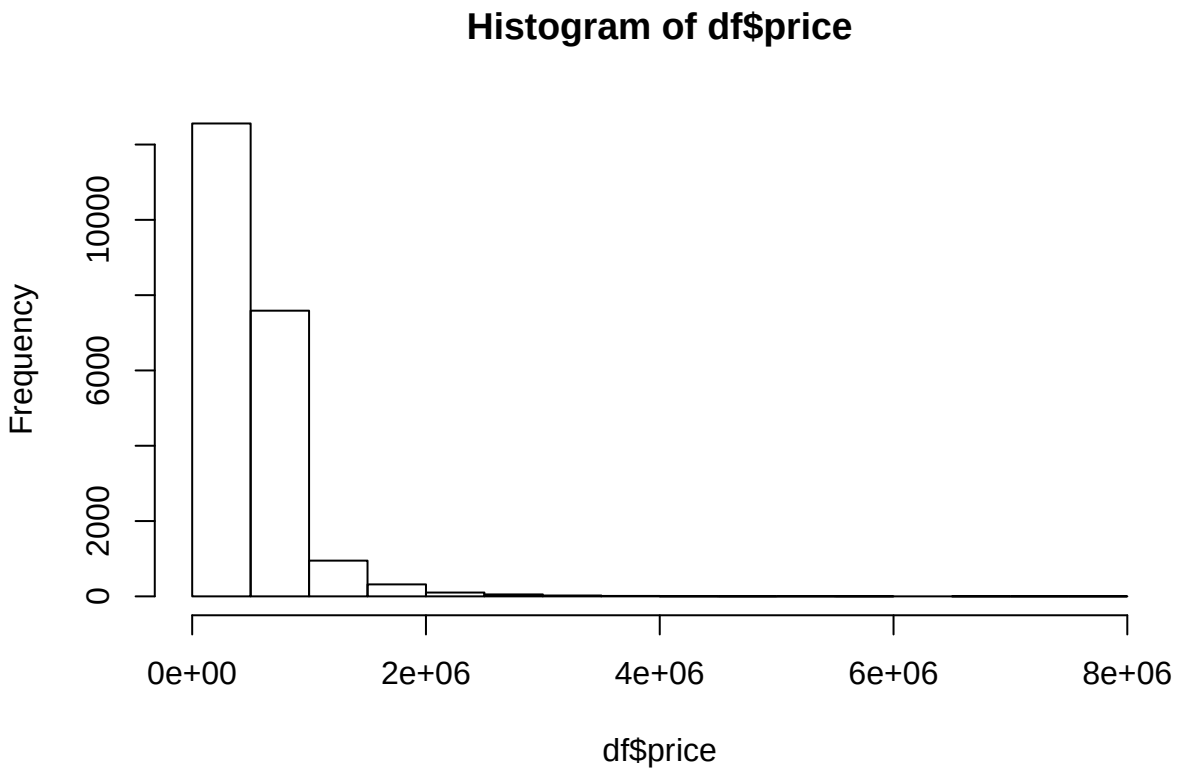
```
df <- df %>%
  mutate(renovated = ifelse(yr_renovated==0,0,1))
```

I do the same thing for basement. If the new variable basement equals 1, the house has a basement, and equals 0 otherwise.

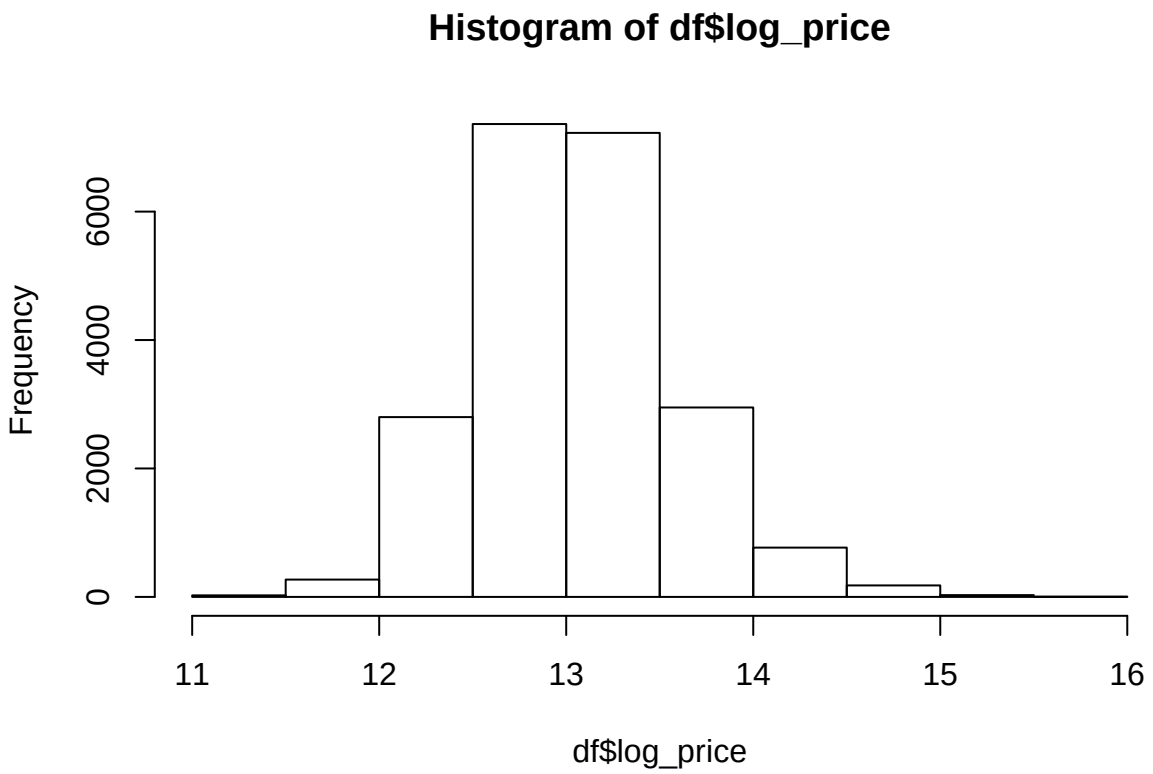
```
df <- df %>%
  mutate(basement = ifelse(sqft_basement==0,0,1))
```

The response variable price is highly skewed, to make the model closer to linear, a log-transformation on price is applicable:

```
hist(df$price)
```




```
df <- df %>%
  mutate(log_price = log(price))
hist(df$log_price)
```



Assumptions

There are some assumptions for the model:
1. The relationship between the response variable and predictors is linear and the additivity of the model is valid.
2. The errors from the prediction line are independent.
3. The variance of errors are equal.

Model

The model of interest would be:
$y_i \sim N(\alpha_{j[i]} + \beta X_i, \sigma_y^2)$ for i = 1,. . . ,n
$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2)$ for j = 1,. . . ,J
where y is the price of the house, x includes number of bedrooms, number of bathrooms, number of floors, has waterfront or not, condition of the house, grade of the house, living room area in 2015, lot area in 2015, age of the house, renovated or not, has basement or not. The intercept alpha differs among areas in King County, according to zipcodes.

```
attach(df)
y <- log_price
n <- nrow(df)
zip_uniq <- unique(zipcode)
J <- length(zip_uniq)
zip <- rep(NA, J)
for(i in 1:J){
  zip[zipcode == zip_uniq[i]] <- i
}
#model
```

```
kc.model <- function() {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- alpha[zip[i]] + beta[1]*bedrooms[i] + beta[2]*bathrooms[i] + beta[3]*floors[i] + beta[4
  }
  for (B in 1:10){
    beta[B] ~ dnorm (0, .0001)
  }
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    alpha[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- mu.a
    }
  mu.a ~ dnorm (0, .0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}
#write model
write.model(kc.model, "kc.model.rjags")
#set up
kc.data <- list ("n", "J", "y", "zip", "bedrooms", "bathrooms", "floors", "waterfront", "condition", "g
kc.inits <- function (){
  list (alpha=rnorm(J), beta=rnorm(10), mu.a=rnorm(1), sigma.y=runif(1), sigma.a=runif(1))}
kc.parameters <- c ("alpha", "beta", "mu.a", "sigma.y", "sigma.a")
#call to bugs
kc.out <- jags (kc.data, kc.inits, kc.parameters,
                model="kc.model.rjags", n.chains=3,n.iter = 10, DIC=TRUE)
```

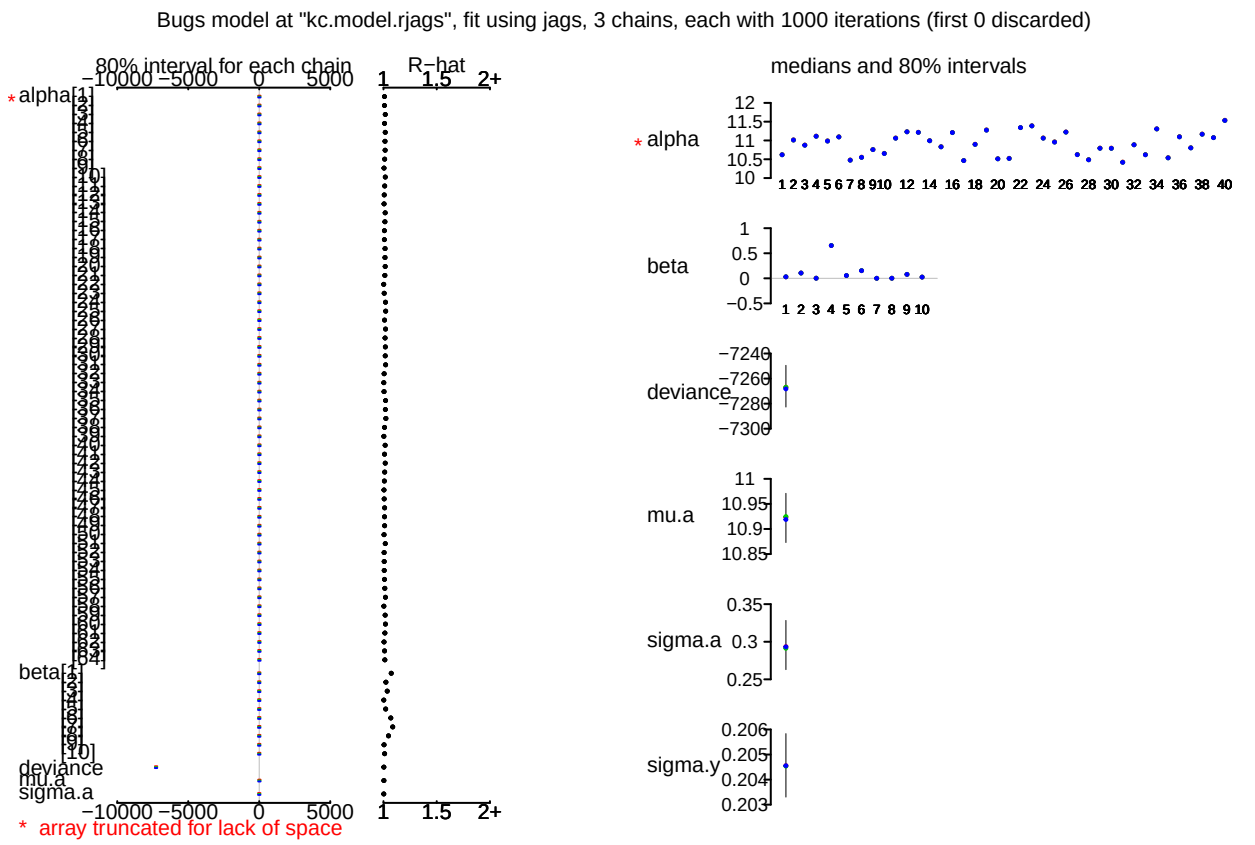
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 21613
##    Unobserved stochastic nodes: 83
##    Total graph size: 281500
##
## Initializing model
```

```
#RUN UNTIL G&R < 1.1
kc.out.ok <- autojags(kc.out)
#output
out <- kc.out.ok$BUGSoutput$summary
out[,c(1:8)] <- format(out[,c(1:8)], digits = 2, nsmall = 2, scientific = F)
knitr::kable(rbind(head(out,5), "..."="",tail(out, 20)))
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat
alpha[1]	10.621212	0.018264	10.586561	10.608980	10.620970	10.634611	10.655619	1.0095
alpha[2]	11.011898	0.016912	10.979348	10.999520	11.012528	11.024210	11.042868	1.0092
alpha[3]	10.874628	0.018525	10.838003	10.861366	10.874883	10.888309	10.909620	1.0169
alpha[4]	11.112822	0.018934	11.075449	11.099485	11.112881	11.126629	11.147717	1.0102
alpha[5]	10.985549	0.017644	10.951694	10.972687	10.985883	10.998702	11.018304	1.0168
...								

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat
alpha[65]	10.772903	0.017138	10.739100	10.760896	10.773286	10.784921	10.805044	1.0109
alpha[66]	10.948217	0.018502	10.912617	10.935047	10.948592	10.961341	10.983780	1.0125
alpha[67]	10.574507	0.022000	10.532442	10.559097	10.574823	10.589675	10.617097	1.0056
alpha[68]	10.881681	0.022396	10.837214	10.866858	10.882370	10.896731	10.924286	1.0065
alpha[69]	10.600515	0.018339	10.564738	10.587567	10.600711	10.613440	10.635093	1.0093
alpha[70]	11.682353	0.032827	11.617434	11.659984	11.683101	11.704969	11.744436	1.0062
beta[1]	0.030107	0.001771	0.026688	0.028921	0.030013	0.031262	0.033739	1.0730
beta[2]	0.105602	0.002933	0.099988	0.103547	0.105598	0.107629	0.111293	1.0229
beta[3]	0.002312	0.003759	-0.005121	0.000078	0.002235	0.004670	0.009459	1.0355
beta[4]	0.654593	0.016557	0.622309	0.643310	0.654615	0.665614	0.686681	1.0030
beta[5]	0.055516	0.002542	0.051039	0.053655	0.055279	0.057337	0.060384	1.0207
beta[6]	0.153115	0.001880	0.149322	0.151757	0.153227	0.154600	0.156531	1.0688
beta[7]	0.000176	0.000003	0.000171	0.000174	0.000176	0.000178	0.000183	1.0869
beta[8]	0.001644	0.000076	0.001495	0.001594	0.001645	0.001695	0.001790	1.0471
beta[9]	0.078775	0.007401	0.063953	0.073999	0.078602	0.083573	0.093734	1.0055
beta[10]	0.024731	0.003600	0.017450	0.022354	0.024803	0.027244	0.031637	1.0110
deviance	-7266.560204	13.079735	-7289.740742	-7275.695902	-7267.244022	-7258.263720	-7238.514089	1.0036
mu.a	10.922524	0.038144	10.848974	10.896583	10.922840	10.948344	10.998656	1.0041
sigma.a	0.294146	0.025345	0.247790	0.276188	0.292645	0.310585	0.348261	1.0029
sigma.y	0.204551	0.000986	0.202617	0.203892	0.204552	0.205219	0.206497	1.0005

```
plot(kc.out.ok)
```



The coefficient summaries provided in the table are the estimated marginal posterior means, standard errors, and credible intervals quantiles $\hat{R}$ and effective sample size for the coefficients of the model. These were

produced by summarizing the marginal draws from the last 1000 iterations of 3 iteration chains. We can see that the estimated mean of alpha is 10.9178819, with a 50% uncertainty interval of [10.8426177, 10.9950774] and a 95% interval of [10.9184067, 10.9440721], and it varies in 70 different areas in King County. The within-area standard deviation σ_y is estimated to be larger than the between-area area stardard deviation σ_α. The according estimated coefficients for predictors are also listed. β_1,β_2,...,β_{10} corresponds to coefficients of predictors, the number of bedrooms, the number of bathrooms, number of floors, has waterfront or not, condition of the house, grade of the house, living room area in 2015, lot area in 2015, age of the house, renovated or not, has basement or not.