Ufuk Secilmis

STAT 8670

# Manufactured Housing's Price Data Analysis using Bayesian Inference

## Introduction

Manufactured housing is a home unit constructed primarily or entirely off-site at factories prior to being moved to a piece of property where it is set. They are growing more popular in the US, so manufactured housing prices have been increasing. In this project, Markov Chain Monte Carlo Methods will be used to analyze whether manufactured housing prices increased between 2018-2021. The data is found on the US Census Bureau website. (https://www.census.gov/data/datasets/2021/econ/MHS/puf.html). The data includes 4686 observations in each year from 2018 to 2021. The questions below will be investigated.

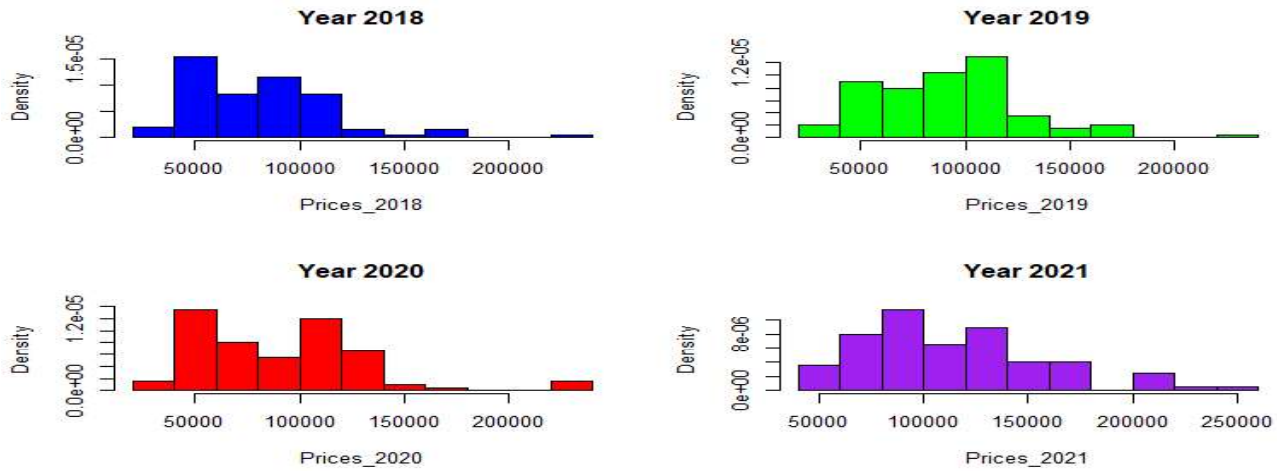
1. Is there a difference in mean number of manufactured housing prices from 2018 to 2019?
2. Is there a difference in mean number of manufactured housing prices from 2019 to 2020?
3. Is there a difference in mean number of manufactured housing prices from 2020 to 2021?

## Method

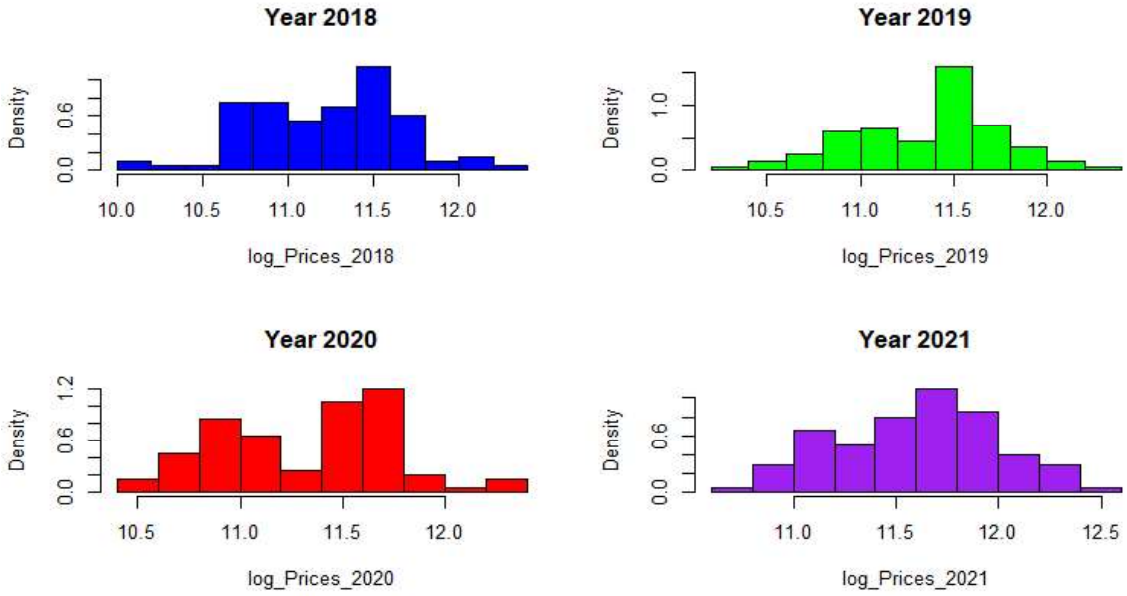
First, the number of missing values in each year will be detected, and missing values will be filled out with the mean of each year. Second, 100 data items will be randomly selected from each year to prevent the likelihood function from being so strong then histogram of randomly selected manufactured housing prices will be examined to determine distribution of the data. Transformations will be applied if necessary. Third, likelihood and prior functions will be developed based on assumed distribution then posterior distribution will be obtained by multiplying the likelihood and prior functions. Finally, Markov chain will be run using Metropolis algorithm to generate random variables from posterior distribution. Once random variables are obtained from posterior distribution, questions above will be investigated.

## Distribution of randomly selected data

Distribution of randomly selected data will be investigated to see common characteristics.




As seen, data in each year is right skewed, so log transformation will be applied to make data symmetric.




The data looks more symmetric after log transformation.

## 2018 - 2019 Manufactured Housing Prices Analysis

The log of manufactured housing prices is assumed to be normally distributed in the analysis. Therefore, the log(Prices_2018) will come from $N(\mu_1, \sigma_1^2)$ and the log(Prices_2019) will come from $N(\mu_2, \sigma_2^2)$. Thus, our parameter vector $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2)$. The likelihood function is given below.

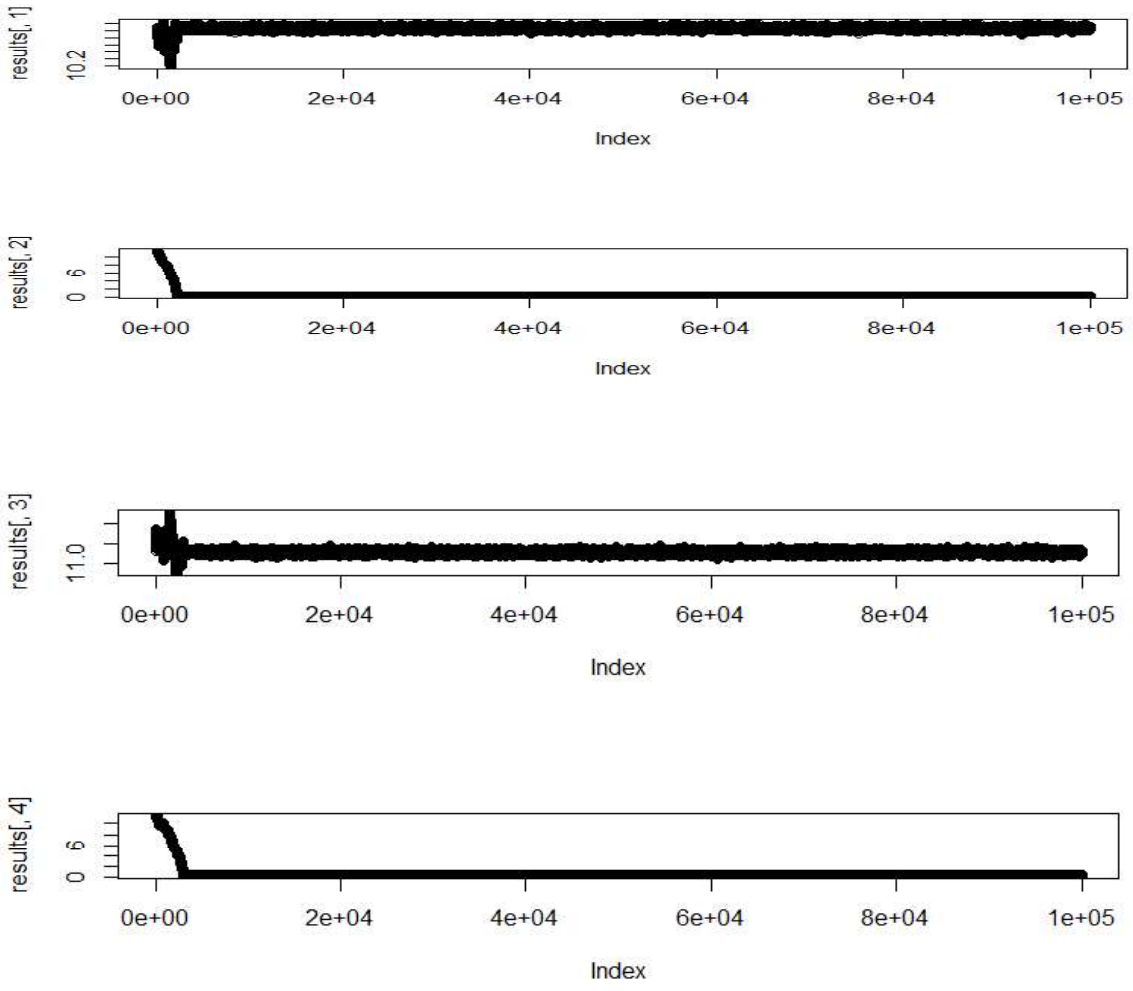
$$P(data|\theta) = P(\log 2018|\theta)P(log2019|\theta) = \prod_{i=1}^{100} N(\log 2018_i|\mu_1, \sigma_1^2)N(log2019_i|\mu_2, \sigma_2^2)$$

Thus, distribution of $\mu_1$ $is$ $N(11.27, 11.27^2)$ and distribution of $\mu_2$ $is$ $N(11.29, 11.29^2)$. Furthermore, prior distribution should be determined. Distribution of $\sigma_1$ is assumed to be exponentially distributed with mean 11.27 and distribution of $\sigma_2$ is assumed to be exponentially distributed with mean 11.29. As all parameters are independent, prior distribution can be developed as:

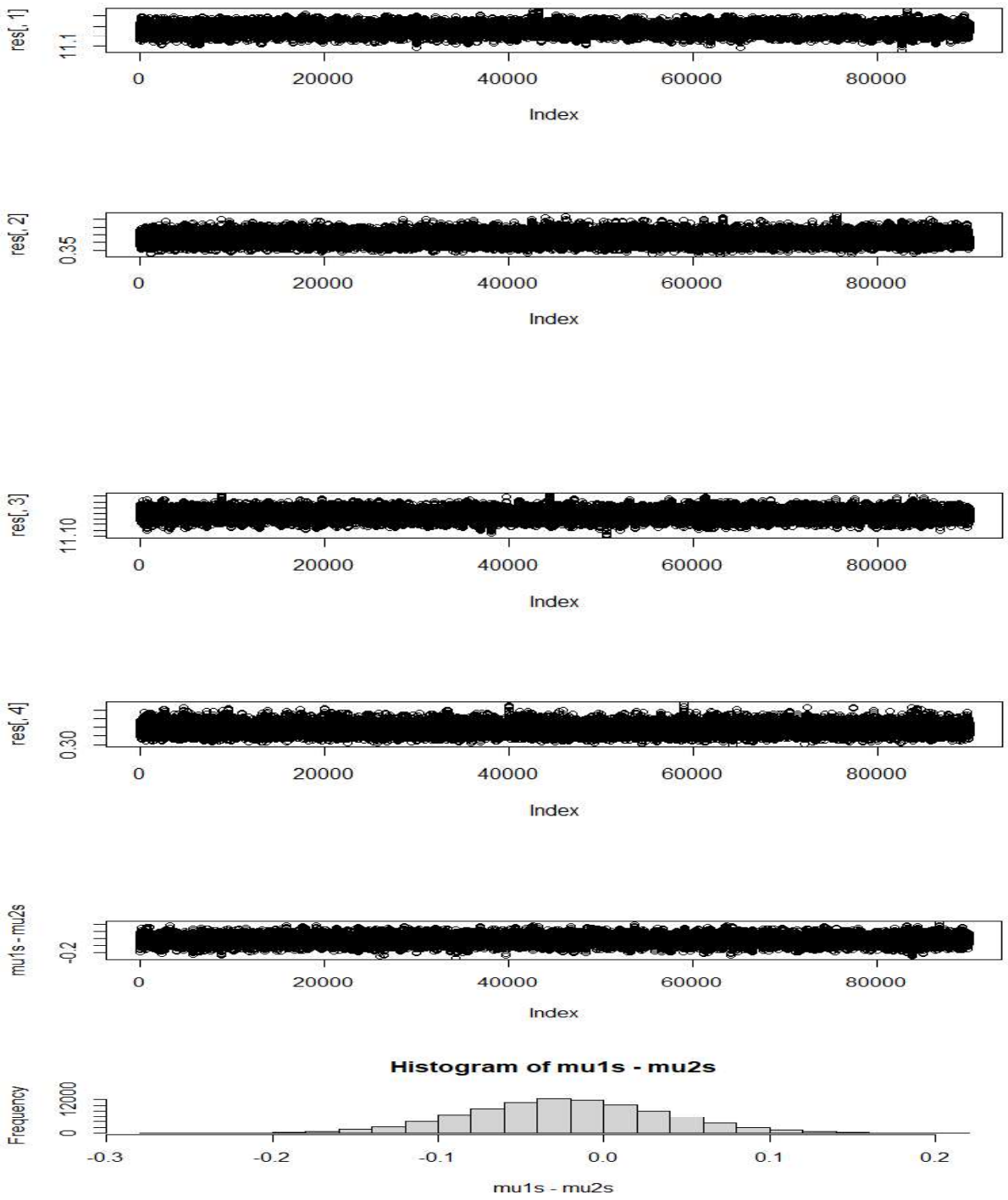
$$P(\theta) = P(\mu_1, \mu_2, \sigma_1, \sigma_2)$$

$$= \frac{1}{\sqrt{2\pi(11.27^2)}} e^{\frac{(\mu_1 - 11.27)}{2(11.27^2)}} \frac{1}{\sqrt{2\pi(11.29^2)}} e^{\frac{(\mu_2 - 11.29)}{2(11.29^2)}} \frac{1}{11.27} e^{\frac{-\sigma_1}{11.27}} \frac{1}{11.29} e^{\frac{-\sigma_2}{11.29}}$$

After prior distribution is developed, posterior distribution will be created by multiplying the likelihood and prior distributions. An initial vector is needed to start running algorithm. Initial vector is chosen to be $\theta_0 = (11.27, 11.27, 11.29, 11.29)$. Then Markov Chain will be run for 100,000 iterations using Metropolis algorithm to generate random variables from posterior distribution. Results are shown below.




As seen, random variables converge after some iterations. Burn-in will be removed. Remaining iterations will be used to analyze if there is a difference in mean prices between 2018 and 2019. Remaining iterations are shown below.




$P(\log(\mu_1) - \log(\mu_2) < 0) = 0.6526816$ that is the probability that manufactured housing prices in 2018 is less than in 2019. As sample means are not far away from each other, this probability is expected.

**2019 - 2020 Manufactured Housing Prices Analysis**

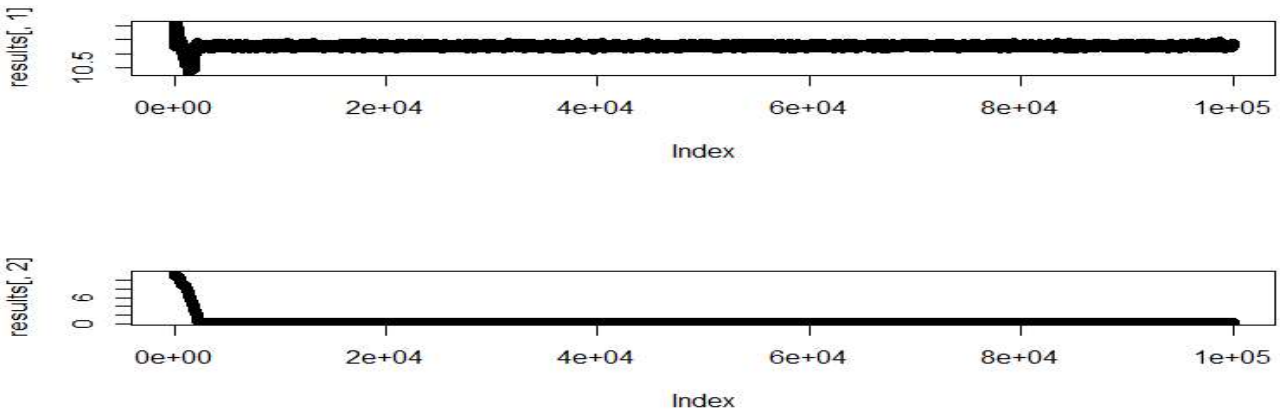
The log of manufactured housing prices is assumed to be normally distributed in the analysis. Therefore, the log(Prices_2019) will come from $N(\mu_1, \sigma_1^2)$ and the log(Prices_2020) will come from $N(\mu_2, \sigma_2^2)$. Thus, our parameter vector $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2)$. The likelihood function is given below.

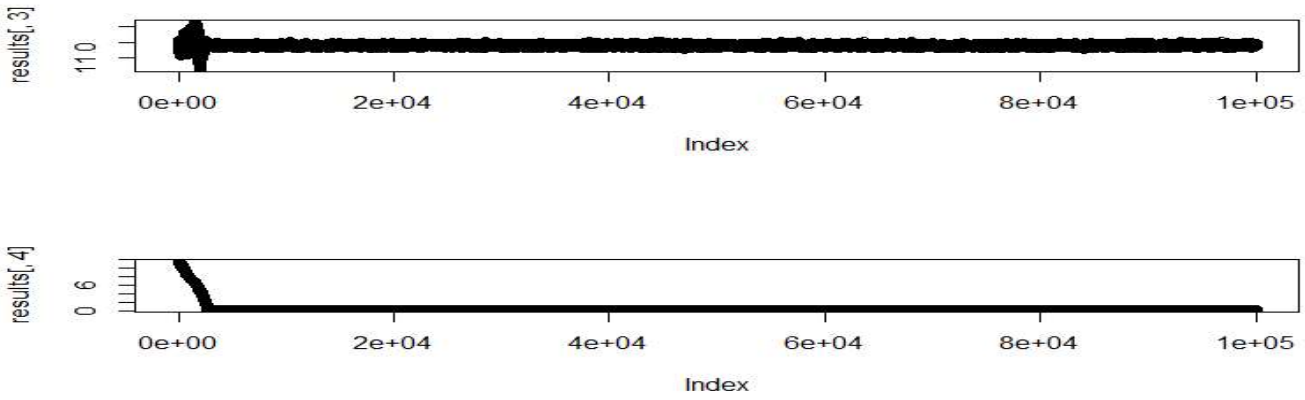
$$P(data|\theta) = P(\log2019|\theta)P(log2020|\theta) = \prod_{i=1}^{100} N(\log2019_i|\mu_1, \sigma_1^2)N(log2020_i|\mu_2, \sigma_2^2)$$

Thus, distribution of $\mu_1$ *is* $N(11.29, 11.29^2)$ and distribution of $\mu_2$ *is* $N(11.40, 11.40^2)$. Furthermore, prior distribution should be determined. Distribution of $\sigma_1$ is assumed to be exponentially distributed with mean 11.29 and distribution of $\sigma_2$ is assumed to be exponentially distributed with mean 11.40. As all parameters are independent, prior distribution can be developed as:

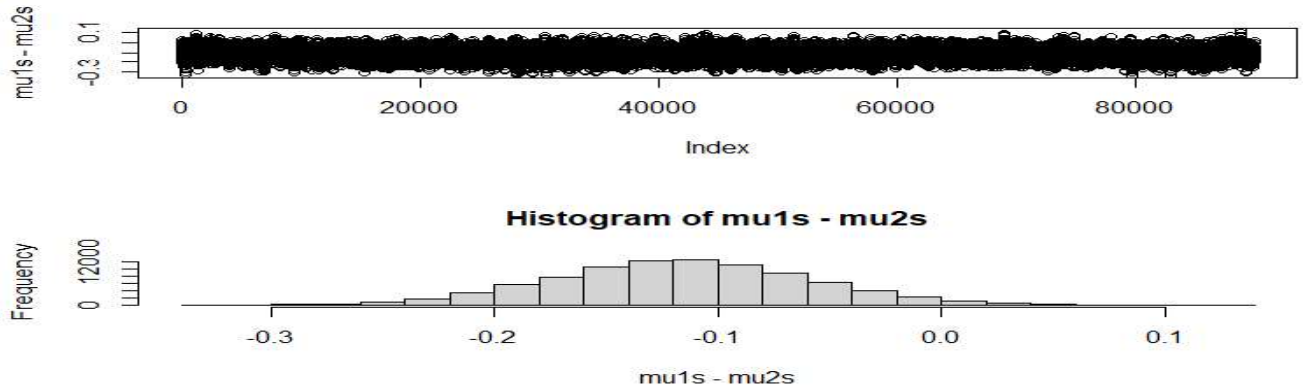
$$P(\theta) = P(\mu_1, \mu_2, \sigma_1, \sigma_2)$$
$$= \frac{1}{\sqrt{2\pi(11.29^2)}} e^{\frac{(\mu_1 - 11.29)}{2(11.29^2)}} \frac{1}{\sqrt{2\pi(11.40^2)}} e^{\frac{(\mu_2 - 11.40)}{2(11.40^2)}} \frac{1}{11.29} e^{\frac{-\sigma_1}{11.29}} \frac{1}{11.40} e^{\frac{-\sigma_2}{11.40}}$$

After prior distribution is developed, posterior distribution will be created by multiplying the likelihood and prior distributions. An initial vector is needed to start running algorithm. Initial vector is chosen to be $\theta_0 = (11.29, 11.29, 11.40, 11.40)$. Then Markov Chain will be run for 100,000 iterations using Metropolis algorithm to generate random variables from posterior distribution. Results are shown below.






As seen, random variables converge after some iterations. Burn-in will be removed. Remaining iterations will be used to analyze if there is a difference in mean prices between 2019 and 2020. Remaining iterations are shown below.




$P(\log(\mu_1) - \log(\mu_2) < 0) = 0.9804891$ that is the probability that manufactured housing prices in 2019 is less than in 2020. As sample means are not close to each other, this probability is expected.

**2020 - 2021 Manufactured Housing Prices Analysis**

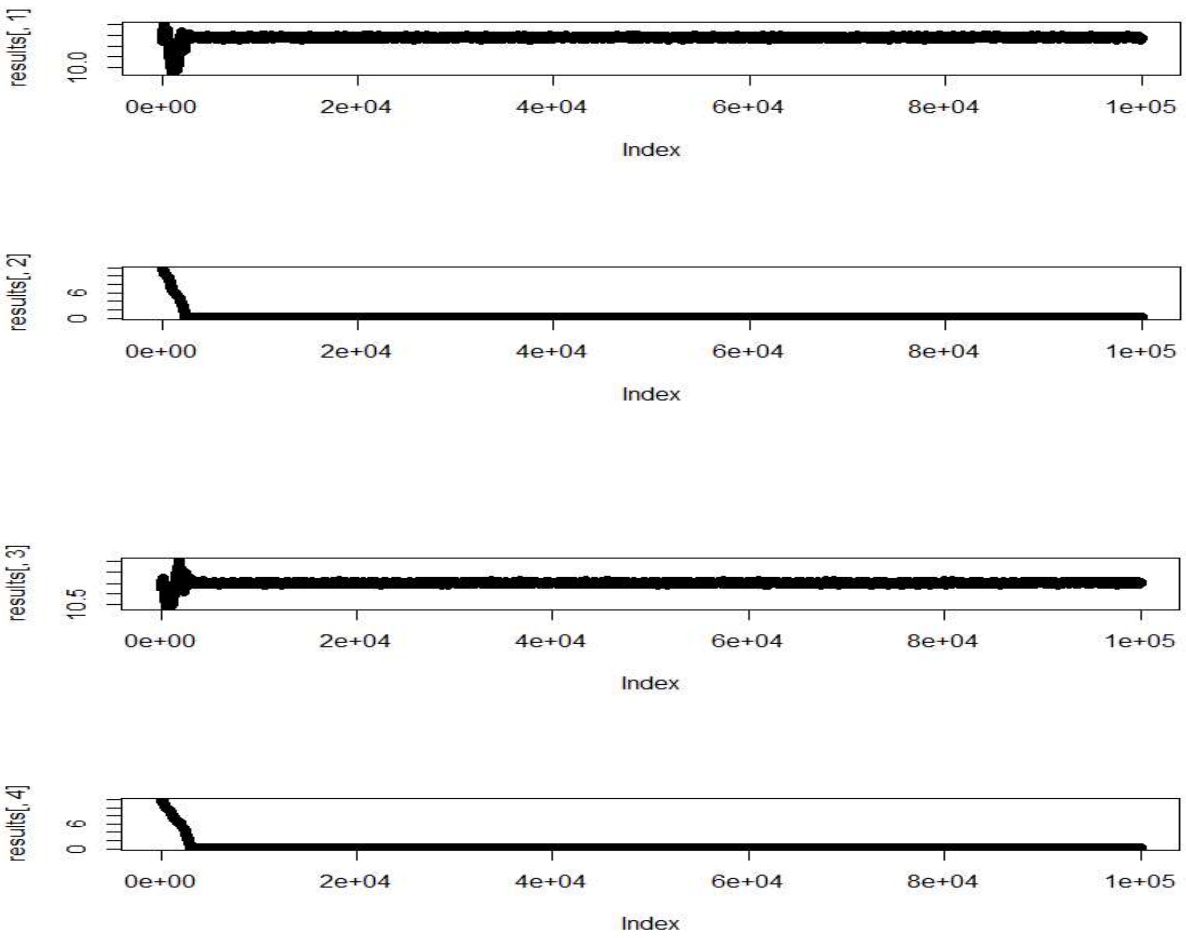
The log of manufactured housing prices is assumed to be normally distributed in the analysis. Therefore, the log(Prices_2020) will come from $N(\mu_1, \sigma_1^2)$ and the log(Prices_2021) will come from $N(\mu_2, \sigma_2^2)$. Thus, our parameter vector $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2)$. The likelihood function is given below.

$$P(data|\theta) = P(\log2020|\theta)P(log2021|\theta) = \prod_{i=1}^{100} N(\log2020_i|\mu_1, \sigma_1^2)N(log2021_i|\mu_2, \sigma_2^2)$$

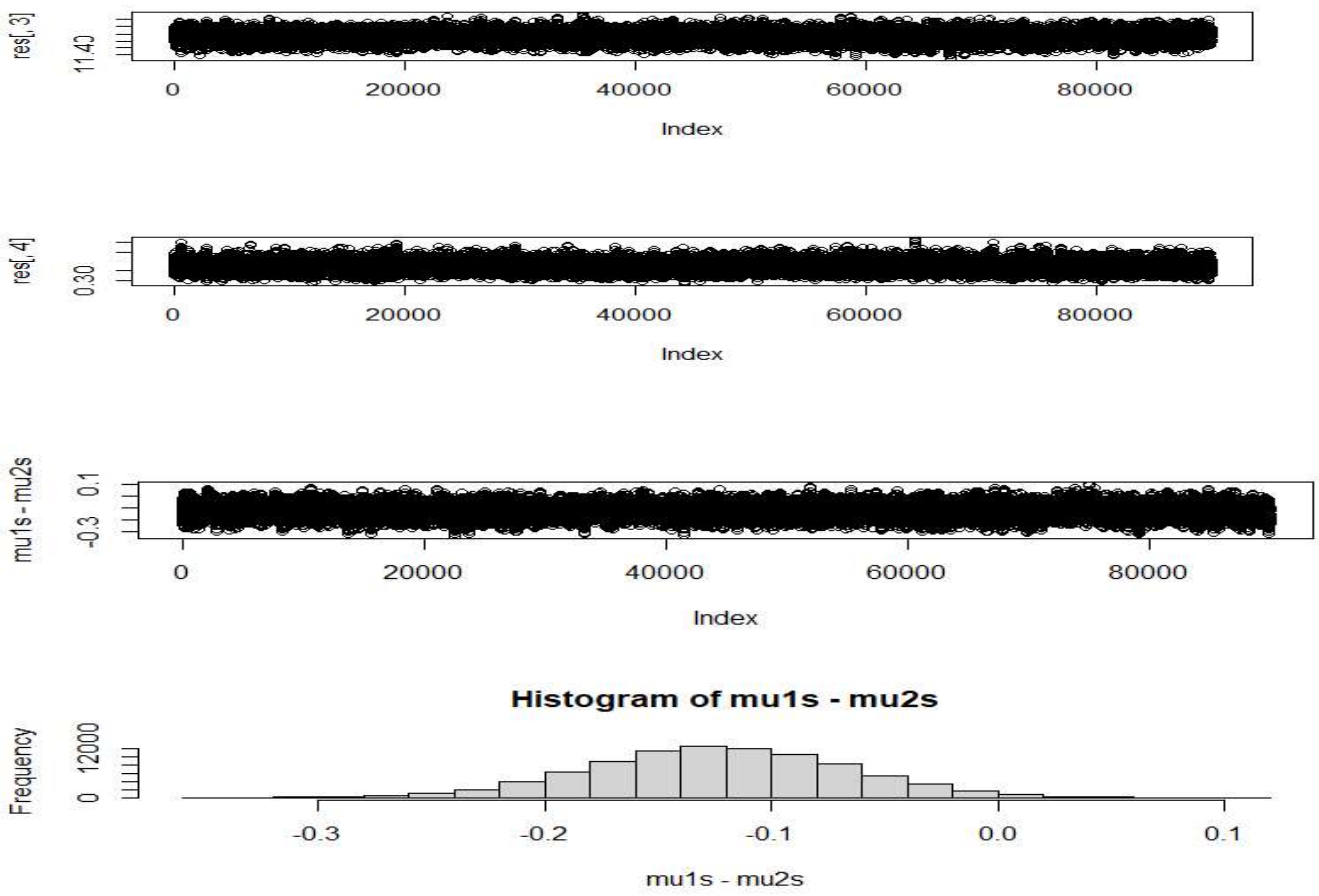
Thus, distribution of $\mu_1$ *is* $N(11.40, 11.40^2)$ and distribution of $\mu_2$ *is* $N(11.53, 11.53^2)$. Furthermore, prior distribution should be determined. Distribution of $\sigma_1$ is assumed to be exponentially distributed with mean 11.40 and distribution of $\sigma_2$ is assumed to be exponentially distributed with mean 11.53. As all parameters are independent, prior distribution can be developed as:

$$P(\theta) = P(\mu_1, \mu_2, \sigma_1, \sigma_2)$$
$$= \frac{1}{\sqrt{2\pi(11.40^2)}} e^{\frac{(\mu_1 - 11.40)}{2(11.40^2)}} \frac{1}{\sqrt{2\pi(11.53^2)}} e^{\frac{(\mu_2 - 11.53)}{2(11.53^2)}} \frac{1}{11.40} e^{\frac{-\sigma_1}{11.40}} \frac{1}{11.53} e^{\frac{-\sigma_2}{11.53}}$$

After prior distribution is developed, posterior distribution will be created by multiplying the likelihood and prior distributions. An initial vector is needed to start running algorithm. Initial vector is chosen to be $\theta_0 = (11.40, 11.40, 11.53, 11.53)$. Then Markov Chain will be run for 100,000 iterations using Metropolis algorithm to generate random variables from posterior distribution. Results are shown below.




As seen, random variables converge after some iterations. Burn-in will be removed. Remaining iterations will be used to analyze if there is a difference in mean prices between 2020 and 2021. Remaining iterations are shown below.




$P(\log(\mu_1) - \log(\mu_2) < 0) = 0.9865224$ that is the probability that manufactured housing prices in 2020 is less than in 2021. As sample means are not close to each other, this probability is expected.

## Conclusion

Based on the results above, it can be said that manufactured housing prices has been increasing. There is approximately 65% chance that manufactured housing prices in 2019 is greater than 2018. There is strong evidence that manufactured housing prices in 2020 is greater than 2019 and manufactured housing prices in 2021 is greater than 2020. The results obtained in this project can be confirmed using the articles in references. [1], [2]

References

[1] "Manufactured home sales are skyrocketing—here's why," *Stacker*. https://stacker.com/stories/44025/manufactured-home-sales-are-skyrocketing-heres-why (accessed Nov. 29, 2022).

[2] "Mobile homes are rising in value. But current residents can't cash out," *Georgia Public Broadcasting*. https://www.gpb.org/news/2022/10/27/mobile-homes-are-rising-in-value-current-residents-cant-cash-out (accessed Nov. 29, 2022).

Dr. Brian Pidgeon's Lecture Notes

**R code**

```
# Read Table
House_Prices = read.csv("House_Prices.csv",header=T)
head(House_Prices)
# Check if there is a missing value in each year
colSums(is.na(House_Prices))
# Fill out missing values with columns' means
for (i in 1:ncol(House_Prices)) {
  House_Prices[,i][is.na(House_Prices[,i])] = mean(House_Prices[,i],na.rm = T)
}
colSums(is.na(House_Prices))
# Take 100 samples
Prices_2018 = sample(House_Prices$Prices_2018,100,replace = F)
Prices_2019 = sample(House_Prices$Prices_2019,100,replace = F)
Prices_2020 = sample(House_Prices$Prices_2020,100,replace = F)
Prices_2021 = sample(House_Prices$Prices_2021,100,replace = F)
#Checking distribution of data
par(mfrow= c(2,2))
hist(Prices_2018,probability = T,col = "blue",main = "Year 2018")
hist(Prices_2019,probability = T,col = "green",main = "Year 2019")
hist(Prices_2020,probability = T,col = "red",main = "Year 2020")
hist(Prices_2021,probability = T,col = "purple",main = "Year 2021")
# Data is right skewed. Log transformation will be applied.
```

```
# Log Transformation
log_Prices_2018 = log(Prices_2018)
log_Prices_2019 = log(Prices_2019)
log_Prices_2020 = log(Prices_2020)
log_Prices_2021 = log(Prices_2021)
mean(log_Prices_2018)
mean(log_Prices_2019)
mean(log_Prices_2020)
mean(log_Prices_2021)
# Checking the distribution of the data after log transformation
hist(log_Prices_2018,probability = T,col = "blue",main = "Year 2018")
hist(log_Prices_2019,probability = T,col = "green",main = "Year 2019")
hist(log_Prices_2020,probability = T,col = "red",main = "Year 2020")
hist(log_Prices_2021,probability = T,col = "purple",main = "Year 2021")
# Data is looked symmetric after log transformation
# 2018 - 2019
mean(log_Prices_2018)
# 11.26665
mean(log_Prices_2019)
# 11.28928
#Likelihood Function
like=function(th){
  mu1=th[1]; sig1=th[2]; mu2=th[3]; sig2=th[4]
  prod(dnorm(log_Prices_2018, mean=mu1,sd=sig1))*prod(dnorm(log_Prices_2019,mean=mu2,sd=sig2))
}
#prior Distribution
Prior=function(th){
  mu1=th[1]; sig1=th[2]; mu2=th[3]; sig2=th[4]
  if (sig1<=0 | sig2<=0) return(0)
  dnorm(mu1,11.27,11.27)*dnorm(mu2,11.29,11.29)*dexp(sig1,rate=1/11.27)*dexp(sig2,rate=1/11.29)
```

```
}
#posterior
Posterior=function(th){Prior(th)*like(th)}
#starting
mu1=11.27; sig1=11.27; mu2=11.29; sig2=11.29
th0=c(mu1,sig1,mu2,sig2)
nit=100000
results=matrix(0,nrow=nit,ncol=4)
th=th0
results[1,]=th0
for (it in 2:nit){
  Cand=th + rnorm(4,sd=.03)
  ratio=Posterior(Cand)/Posterior(th)
  if (runif(1) < ratio) th=Cand
  results[it,]=th
}
edit(results)
par(mfrow=c(2,1))
plot(results[,1])
plot(results[,2])
plot(results[,3])
plot(results[,4])

res=results[1e+04:1e+05,]
par(mfrow=c(2,1))
plot(res[,1])
plot(res[,2])
plot(res[,3])
plot(res[,4])
mu1s=res[,1]
```

```
sig1s=res[,2]
mu2s=res[,3]
sig2s=res[,4]
par(mfrow=c(2,1))
plot(mu1s-mu2s)
hist(mu1s-mu2s)
mean(mu1s-mu2s<0)
# [1] 0.6526816
# 2019 - 2020
mean(log_Prices_2019)
# 11.28928
mean(log_Prices_2020)
# 11.40545
#Likelihood Function
like=function(th){
  mu1=th[1]; sig1=th[2]; mu2=th[3]; sig2=th[4]
  prod(dnorm(log_Prices_2019, mean=mu1,sd=sig1))*prod(dnorm(log_Prices_2020,mean=mu2,sd=sig2))
}
#prior Distribution
Prior=function(th){
  mu1=th[1]; sig1=th[2]; mu2=th[3]; sig2=th[4]
  if (sig1<=0 | sig2<=0) return(0)
  dnorm(mu1,11.29,11.29)*dnorm(mu2,11.40,11.40)*dexp(sig1,rate=1/11.29)*dexp(sig2,rate=1/11.40)
}
#posterior
Posterior=function(th){Prior(th)*like(th)}
#starting
mu1=11.29; sig1=11.29; mu2=11.40; sig2=11.40
th0=c(mu1,sig1,mu2,sig2)
nit=100000
```

```
results=matrix(0,nrow=nit,ncol=4)
th=th0
results[1,]=th0
for (it in 2:nit){
  Cand=th + rnorm(4,sd=.03)
  ratio=Posterior(Cand)/Posterior(th)
  if (runif(1) < ratio) th=Cand
  results[it,]=th
}
edit(results)
par(mfrow=c(2,1))
plot(results[,1])
plot(results[,2])
plot(results[,3])
plot(results[,4])
res=results[1e+04:1e+05,]
par(mfrow=c(2,1))
plot(res[,1])
plot(res[,2])
plot(res[,3])
plot(res[,4])
mu1s=res[,1]
sig1s=res[,2]
mu2s=res[,3]
sig2s=res[,4]
plot(mu1s-mu2s)
hist(mu1s-mu2s)
mean(mu1s-mu2s<0)
#[1] 0.9804891
```

**# 2020 - 2021**

```
mean(log_Prices_2020)
# [1] 11.40545
mean(log_Prices_2021)
# 11.52889
#Likelihood Function
like=function(th){
  mu1=th[1]; sig1=th[2]; mu2=th[3]; sig2=th[4]
  prod(dnorm(log_Prices_2020, mean=mu1,sd=sig1))*prod(dnorm(log_Prices_2021,mean=mu2,sd=sig2))
}
#prior Distribution
Prior=function(th){
  mu1=th[1]; sig1=th[2]; mu2=th[3]; sig2=th[4]
  if (sig1<=0 | sig2<=0) return(0)
  dnorm(mu1,11.40,11.40)*dnorm(mu2,11.53,11.53)*dexp(sig1,rate=1/11.40)*dexp(sig2,rate=1/11.53)
}
#posterior
Posterior=function(th){Prior(th)*like(th)}
#starting
mu1=11.40; sig1=11.40; mu2=11.53; sig2=11.53
th0=c(mu1,sig1,mu2,sig2)
nit=100000
results=matrix(0,nrow=nit,ncol=4)
th=th0
results[1,]=th0
for (it in 2:nit){
  Cand=th + rnorm(4,sd=.03)
  ratio=Posterior(Cand)/Posterior(th)
  if (runif(1) < ratio) th=Cand
  results[it,]=th
}
```

```
edit(results)
par(mfrow=c(2,1))
plot(results[,1])
plot(results[,2])
plot(results[,3])
plot(results[,4])
res=results[1e+04:1e+05,]
plot(res[,1])
plot(res[,2])
plot(res[,3])
plot(res[,4])
mu1s=res[,1]
sig1s=res[,2]
mu2s=res[,3]
sig2s=res[,4]
plot(mu1s-mu2s)
hist(mu1s-mu2s)
mean(mu1s-mu2s<0)
# 0.9865224
```