1. Introduction

The data is taken from Numbeo. It includes indices of cost of living, rent, cost of living and rent, grocery, restaurant, local purchasing power of 578 cities from 5 regions in 2022.

The main interest is to compare cost of living and rent (colnrent) versus local purchasing power (localpp) and to answer the following questions:

- Which model (equal mean/ separate mean/ hierarchical normal model) perform the best in finding the posterior mean and variance?
- Is colnrent higher or lower than localpp?
- Is the difference significant?

If we belief the mean and variance across 5 regions are the same, the prediction error of equal mean model should be lower. If the 5 regions share some information such as policy, prediction error in hierarchical normal model should be the lowest. Otherwise, separate mean model should be used which implies that 5 regions have different mean and variance and they are independent of each other. This will be done by cross-validation.

Therefore the motivation of this study is to assess if there is significant difference between the cost of living and rent and local purchasing power. If yes, what would be the mean difference? Which region has the largest difference and do they share some common information in posterior mean and variance.

Once concluded the best-performed model, diagnosis check will be performed on the model assumption, including the prior distribution, convergence and autocorrelation.

2. Methodology

3 models (equal mean, separate mean, hierarchical normal model) will be fitted for both colnrent and localpp. 3 model assumptions are the same between colnrent and localpp as the prior are set to be weakly informative and 2 indices use New York as the reference level at 100. For simplicity, only the 3 model assumptions of colnrent will be clearly stated.

To perform cross validation, the dataset is split into 2 with around 70% of training data and 30% of testing data.

Case 1: Equal Mean Model

a) Splitting data

70% of the total sample size would be 404.6 so 405 data points will be randomly sample as training set.

b) Model setup

This is equivalent as treating all data as 1 variable:

Sampling model:

$$Yi \sim Normal(\theta, \sigma^2)$$

Prior:

$$\theta \sim Normal(\mu_0, \tau_0^2)$$

$$\sigma^2 \sim InvGamma\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right))$$

ν_0: prior df $\nu_0\sigma_0^2$: prior sum of square

τ_0^2: prior variance of θ μ_0: prior of θ

Priors of each parameter are set to be:

mu0<-100 ; t20<-40^2 # for mu

s20<-40^2 ; nu0<-1 # for sigma2

This way 95% CI for y will be 100+/- 2*40 = (20,180) which is a wide range for the baseline of the index to be 100. $\sigma^2 \sim InvGamma(\frac{1}{2}, 50)$ which is a weakly informative prior.

Posterior:

$$\theta|\sigma^2, y_1, \dots, y_n \sim Normal(\frac{\frac{\mu_0}{\tau_0^2} + \frac{n\bar{y}}{\sigma^2}}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}}, \quad \left[\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}\right]^{-1})$$

$$\sigma^2|\theta, y1, \dots, y_n \sim InvGamma(\frac{\nu_n}{2}, \frac{\nu_n\sigma_n^2}{2})$$

$\nu_n = \nu_0 + n$ is the posterior df

$\sigma_n^2 = \frac{\nu_0\sigma_0^2 + (n-1)s^2 + n(\bar{y}-\theta)^2}{\nu_n}$

$\nu_n\sigma_n^2$ is the posterior sum of square

c) <u>MCMC</u>

MCMC steps:

i. Sample from $\theta^{s+1} \sim Normal(\frac{\frac{\mu_0}{\tau_0^2} + \frac{n\bar{y}}{\sigma^2}}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}}, \ \left[\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}\right]^{-1}$

ii. Sample from $\sigma^{2(s+1)} \sim InvGamma(\frac{\nu_n}{2}, \frac{\nu_n\sigma_n^2}{2})$

iii. Store the value into ϕ^{s+1}

The algorithm is run 10000 times. Since the result is already converging with low autocorrelation. No thinning or burn-in nor increasing the number of iterations were considered.

To initialize the algorithm, the starting value of θ, σ^2 are set to be the mean and variance of the observed training data.

d) <u>Compute prediction error</u>

Prediction error = mean((Y1.test - Y.pred)^2)

Where Y1.test is the test set defined in (a), and Y.pred is the average of the predictive draws of Y in the MCMC step.

Case 2: Separate Mean Model

Separate Mean Model is equivalent as fitting a normal model for each model

a) **Splitting data**

Sample size for each region is: 25, 164, 136, 237, 16. Each region will be split into 7:3 according to their sample size. The size of the training set will be:18, 115, 95, 166, 11

b) **Model setup**

The setup is similar to Case 1 only difference is each region is considered separately

Sampling model:

$$Y_{i,j} \sim Normal(\theta_j, \sigma_j^2)$$

Where j =1,2,3,4,5

Prior:

$$\theta_j \sim Normal(\mu_{0,j}, \tau_{0,j}^2)$$

$$\sigma_j^2 \sim InvGamma\left(\frac{\nu_{0,j}}{2}, \frac{\nu_{0,j}\sigma_{0,j}^2}{2}\right))$$

Meaning of each hyperprior are the same as Case 1. Value of the hyperprior is also set to be the same as Case 1.

Posterior:

$$\theta_j | \sigma_j^2, y_{1,j}, \dots, y_{n,j} \sim Normal(\frac{\frac{\mu_{0,j}}{\tau_{0,j}^2} + \frac{n_j \overline{y}_j}{\sigma_j^2}}{\frac{1}{\tau_{0,j}^2} + \frac{n_j}{\sigma_j^2}}, \quad \left[\frac{1}{\tau_{0,j}^2} + \frac{n_j}{\sigma_j^2}\right]^{-1})$$

$$\sigma_j^2 | \theta_j, y_{1,j}, \dots, y_{n,j} \sim InvGamma(\frac{\nu_{n,j}}{2}, \frac{\nu_{n,j}\sigma_{n,j}^2}{2})$$

c) **MCMC and Compute prediction error**

MCMC steps and prediction error are the same as Case 1 but repeat 5 times for each region.

Case 3: Hierarchical Normal Model

a) **Splitting data** (Same steps as Case 2)

b) **Model setup**

The observation of each city in group j (ie the sampling model):

$$Y_{1,j}, \dots, Y_{n_j,j} \overset{iid}{\sim} N(\theta_j, \sigma_j^2)$$

$i = 1, \dots, n_j$ where n_j is the sample size of each region and I is the ith observation of group j

$j = 1, \dots, m$ where m is the number of groups.

Both observations are assumed to follow a normal distribution. For each region, mean and variance are allowed to vary.

In this case, m = 5 and j in order is referring the data in Africa, America, Asia, Europe and Oceania. Each with a sample size of 18, 115, 95, 166, 11

Prior:

$$\frac{1}{\tau^2} \sim Gamma(\frac{\eta_0}{2}, \frac{\eta_0\tau_0^2}{2})$$
$$\mu \sim N(\mu_0 , \gamma_0^2)$$

η_0: prior df
τ_0^2: prior variance of τ^2
$\eta_0\tau_0^2$: prior sum of square
μ_0: prior of μ
γ_0^2: prior variance of μ

These 2 parameters will determine the variance and the mean of θ_j which is the mean of each group j.

$$\sigma_0^2 \sim Gamma(a, b)$$
$$\nu_0 \sim Geom(\alpha)$$

A,b are the rate and shape parameter of gamma distribution and a is the rate of geometric distribution.

These 2 parameters will determine the variance and the mean of σ_0^2 which is the variance of each group j.

With above 4 parameters, we can draw new pairs of θ_j, σ_j^2. That is updating the group mean and variance and produce an unobserved predictive draw.

Weakly informative priors were set up,

```
## weakly informative priors
  # tau ~ invGamma(1/2, 50)
eta0<-1 ; t20<-100
  # mu ~ normal(100, 40)
mu0<-100 ; g20<-40
  # for sigma2 ~ Gamma(1,1/100) - improper
a0<-1 ; b0<-1/100 ;
  # for nu0
alpha<-1
```

μ is set to center at 100 which is baseline of the index (representing New York) with a variance of 40^2. That way 95% of $\mu = (20{,}180)$ which is a wide range of mu and it is weakly informative.

Posterior:

$$\frac{1}{\tau^2}|\eta, \mu \sim Gamma(\frac{\eta_0 + m}{2}, \frac{\eta_0\tau_0^2 + \sum(\theta_j - \mu)^2}{2})$$

$$\mu|\tau^2, \gamma, \theta \sim N(\frac{\frac{m\bar{\theta}}{\tau^2} + \frac{\mu_0}{\nu_0^2}}{\frac{m}{\tau^2} + \frac{1}{\nu_0^2}}, \left[\frac{m}{\tau^2} + \frac{1}{\nu_0^2}\right]^{-1})$$

$$\left\{\theta_j \middle| Y_{1,j}, \dots, Y_{n_j,j}, \sigma_j^2\right\} \sim N\left(\frac{\frac{\bar{y}_j n_j}{\sigma_j^2} + \frac{\mu}{\tau^2}}{\frac{n_j}{\sigma^2} + \frac{1}{\tau^2}}, \left[\frac{n_j}{\sigma^2} + \frac{1}{\tau^2}\right]^{-1}\right)$$

$$\sigma_0^2 | \sigma_1^2, \dots, \sigma_m^2, \nu_0 \sim Gamma(a + \frac{1}{2} m\nu_0, b + \frac{\nu_0}{2}\sum_{j=1}^{m}\frac{1}{\sigma^2})$$

$$\nu_0 | \sigma_0^2, \sigma_1^2, \dots, \sigma_m^2 \propto \left(\frac{\left(\frac{\nu_0 \sigma_0^2}{2}\right)^{\frac{\nu_0}{2}}}{\Gamma\left(\frac{\nu_0}{2}\right)}\right)^m \times \exp\{-\nu_0(\alpha + \frac{1}{2}\sigma_0^2 \sum_{j=1}^{m}\frac{1}{\sigma_0^2})\}$$

$$\sigma_j^2 | Y_{1,j}, \dots, Y_{n_j,j}, \theta_j \sim invGamma(\frac{\nu_j + n_j}{2}, \sum_{i=1}^{n_j}\left(Y_{i,j} - \theta_j\right)^2$$

Since the posterior distribution of $\nu_0 | \sigma_0^2, \sigma_1^2, \dots, \sigma_m^2$ is not a standard distribution, it will be sampled using grid approximation.

c) <u>MCMC</u>

MCMC is run 50000 times with the following steps:

i. Initialize the parameters values
ii. Draw $\left\{\theta_j \middle| Y_{1,j}, \dots, Y_{n_j,j}, \sigma_j^2\right\}$ for j = 1,2,3,4,5
iii. Draw $\sigma_j^2 | Y_{1,j}, \dots, Y_{n_j,j}, \theta_j$ for j = 1,2,3,4,5
iv. Update $\sigma_0^2 | \sigma_1^2, \dots, \sigma_m^2, \nu_0$
v. Update $\nu_0 | \sigma_0^2, \sigma_1^2, \dots, \sigma_m^2$
vi. Update $\mu | \tau^2, \gamma, \theta$
vii. Update $\frac{1}{\tau^2} | \eta, \mu$
viii. Storing the updated parameters and draws

To initialize the value of each parameter:

```
theta<-ybar; sigma2<-sv2
mu<-mean(theta)
tau2<-var(theta)
s20<-1/mean(1/sv2)
nu0<-10
```

θ_j, σ_j^2 are set to be the mean and variance of each observation data of group j.
μ, τ^2 are set to be the mean and variance of θ_j
σ_0^2, ν_0 are set to be the inverse of mean of precision of data variance and 10.

After 50000 runs, effective size of MTSN = $\mu, \tau^2, \sigma_0^2, \nu_0$ are a bit low so the algorithm is thinned every 5 draws. Thus the total valid draws are 10000.

d) <u>Compute prediction error</u> <u>(same as above)</u>

2 libraries are used: coda and readxl. Coda is loaded to find the effective size of each parameter which will be used to assess the autocorrelation between each draw. Readxl is loaded to read excel file with 5 sheets containing data of 5 regions.

After deciding choice of model, diagnosis plot of convergence, autocorrelation will be generated. Effective size will also be found to assess if there is enough independent posterior draws.

Above model set-up are exactly the same for localpp, only using a different data.

Comparison

In this section, theta and the posterior predictive draws of colnrent and localpp will be compared where theta is the posterior mean of the indices. This would be done by averaging the posterior draws of theta and predictive y by columns to by mean of posterior draws of each region.

3. Results

A. Cost of living and rent

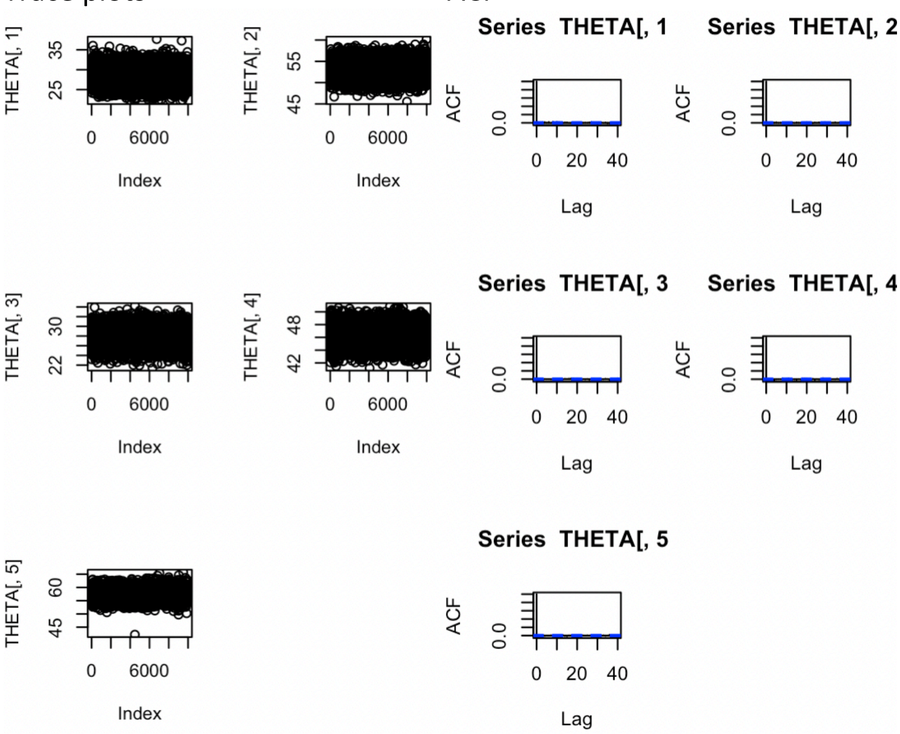
Model	Prediction error
Equal Mean	381.0604
Separate Mean (for 5 regions: Africa, America, Asia, Europe, Oceania)	22.09373 909.72000 271.91258 619.08923 776.33264
Hierarchical (for 5 regions: Africa, America, Asia, Europe, Oceania)	36.77522 335.53593 338.98960 215.17478 197.68126

Conclude from above table that the hierarchical model performs the best in modelling the posterior distribution of parameters.

Diagnosis check will be performed to check the model assumption:

For mean of each group: $\boldsymbol{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5}$

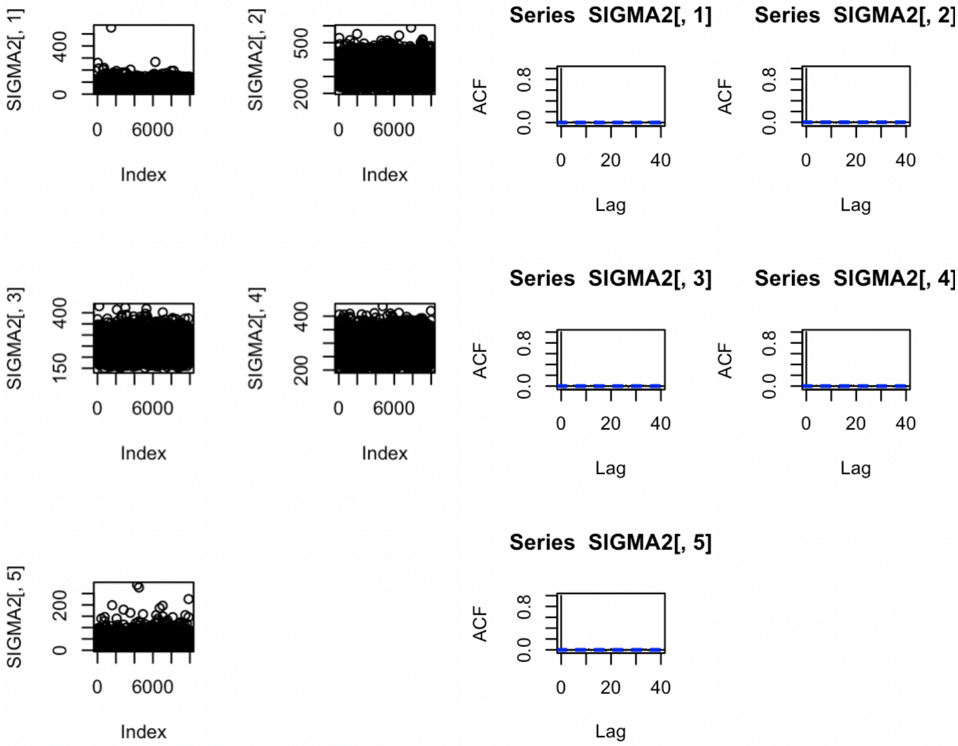
Trace plots ACF



```
> apply(THETA,2,effectiveSize)
[1] 8225.474 10000.000 10000.000 10000.000 10000.000
```

For variance of each group: $\sigma_1^2,\ \sigma_2^2,\ \sigma_3^2,\ \sigma_4^2,\ \sigma_5^2$

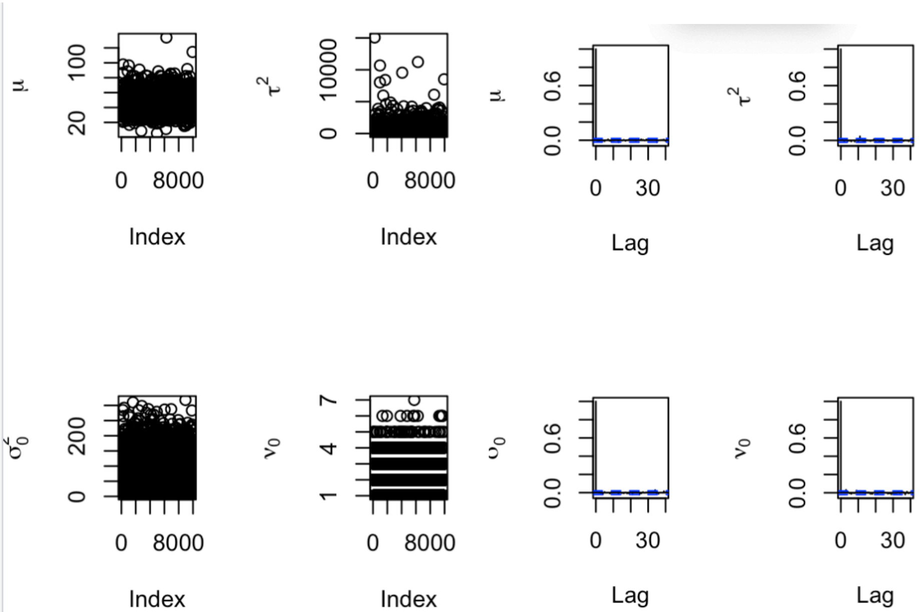
Traceplot ACF




```
> apply(SIGMA2,2,effectiveSize)
[1] 10000.00 10401.35 10492.55 10000.00 10000.00
```

For parameters: $(\mu,\ \tau^2,\ \sigma_0^2, \nu_0)$

Trace plots: ACF:

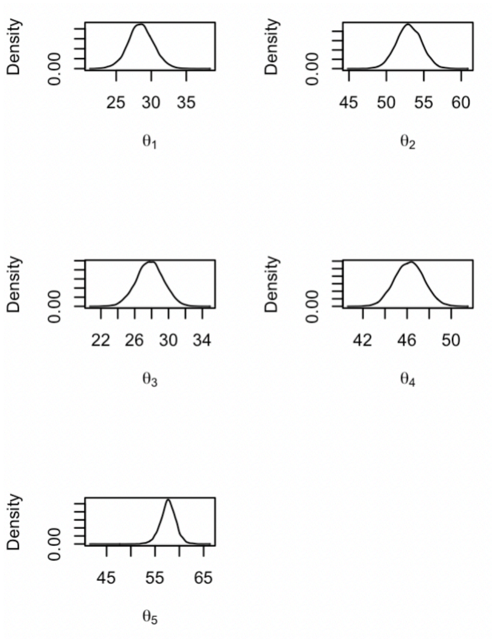



```
# MTSN[s,]<-c(mu,tau2,s20,nu0)
> apply(MTSN,2,effectiveSize)
```

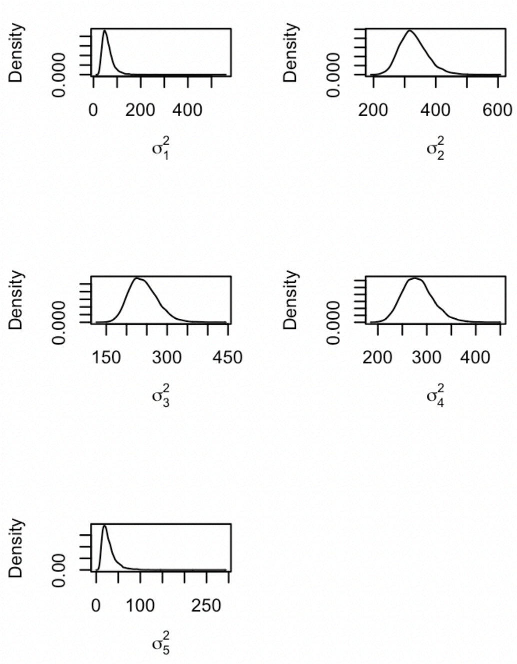
```
[1] 10000.000  9888.465 10000.000  9059.255
```

From above diagnosis, we are confident that the algorithm is converging in all THETAs, SIGMA2s and the parameter. They all have a low autocorrelation and a large enough effective size after thinning.

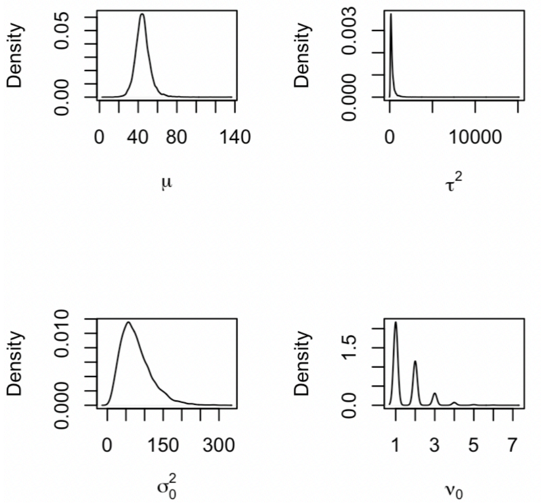
Density plot of THETA



Density plot of Sigma2



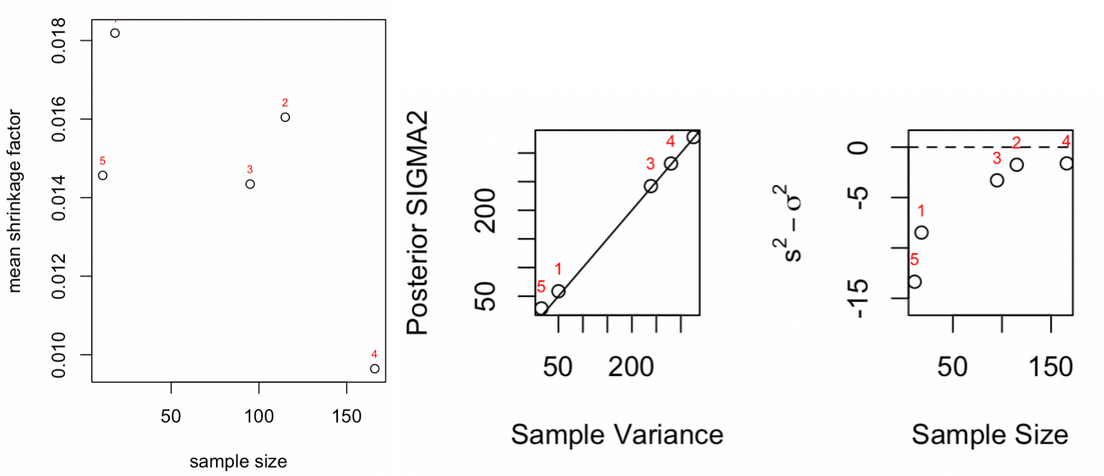
Density plots of MTSN




```
> apply(THETA,2,mean)
[1] 28.51943 53.00095 27.81326 46.19013 57.69560

> apply(SIGMA2,2,mean)
[1]  58.58098 327.87409 242.24005 281.62662  28.61796

> apply(MTSN,2,mean)
[1]  44.57125 298.72918  77.18211   1.56000
```

Density plots shows no problem is the draws. ν_0 is not drawn from a standard distribution so it was grid approximated to find the mean by maximizing the log likelihood so the result should be somewhere around 1 from its density. This is indeed the case after finding its mean.

Since Hierarchical model is chosen, shrinkage will also be assessed:



Region 5 and 1 have the smallest sample size so they shrunk the largest towards the overall mean. Although region 5 has a smallest sample size, region 1 shrunk the most. This is due to larger sample variation in region 1 (50.11) than region 5 (15.26). Posterior variance and sample variance are proportionate.

B. Local Purchasing Power

Model	Prediction error
Equal Mean	1291.827
Separate Mean (for 5 regions: Africa, America, Asia, Europe, Oceania)	538.6271 5241.2049 516.8833 2234.0655 4650.3766
Hierarchical (for 5 regions: Africa, America, Asia, Europe, Oceania)	263.3537 1419.0438 616.3927 630.2744 122.7762

Model choice for local purchasing power is not as obvious as colnrent. Prediction error is lower using hierarchical model for all 5 regions but prediction error of 2nd group (America) has a higher prediction error than the equal mean error. However, since other groups all have a lower error and the difference between the equal mean model and the 2nd group are not very large. Still hierarchical normal model will be used.

Main reason that causes a large prediction error in America is the large sample variance.

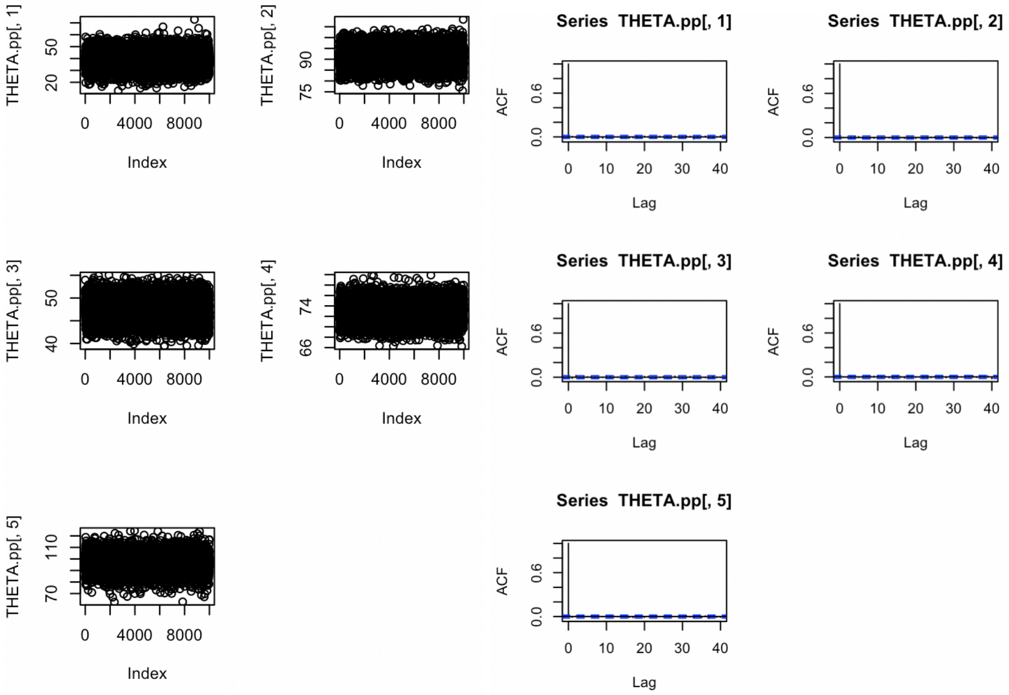
```
> sv2
[1]  647.8168 1710.1204  464.6968  526.1536  450.7217
```

The sample variance of America is 1710.1204 which is the largest among 5 regions. With such large variance in the sampled dataset, it may take more iterations or more data for America to lower the prediction error.

Diagnosis check will be performed to check the model assumption:

For mean of each group: $\boldsymbol{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5}$

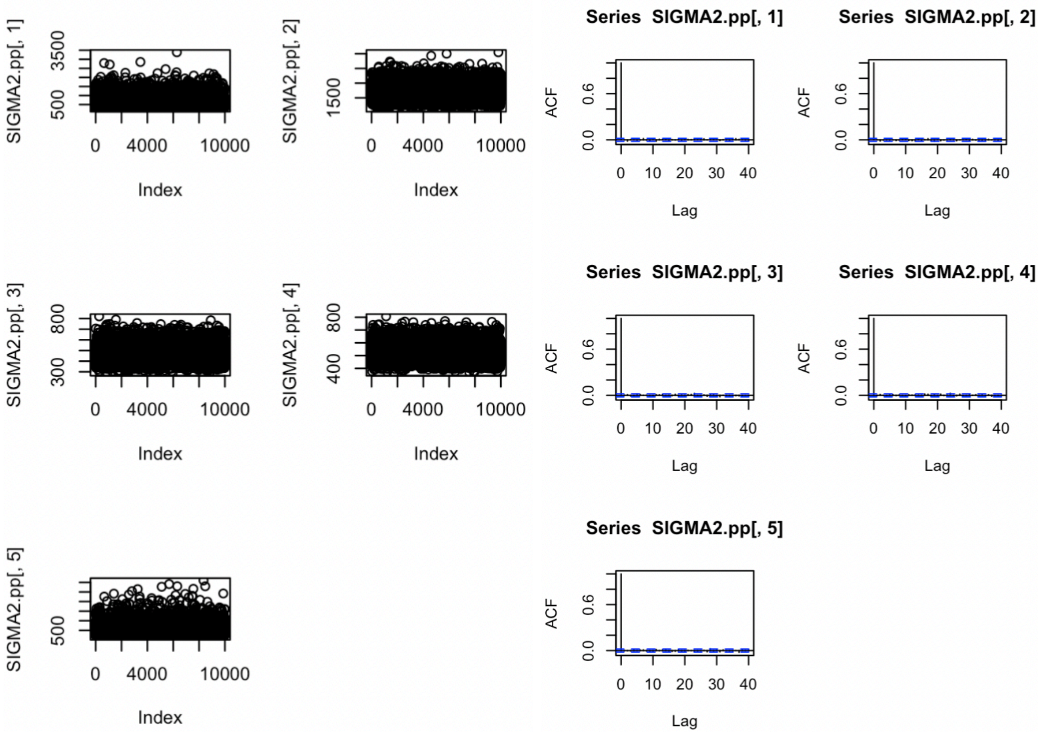
Trace plots ACF



> apply(THETA.pp,2,effectiveSize)
[1] 10000.00 10000.00 9842.34 10000.00 10000.00

For variance of each group: $\sigma_1^2,\ \sigma_2^2,\ \sigma_3^2,\ \sigma_4^2,\ \sigma_5^2$

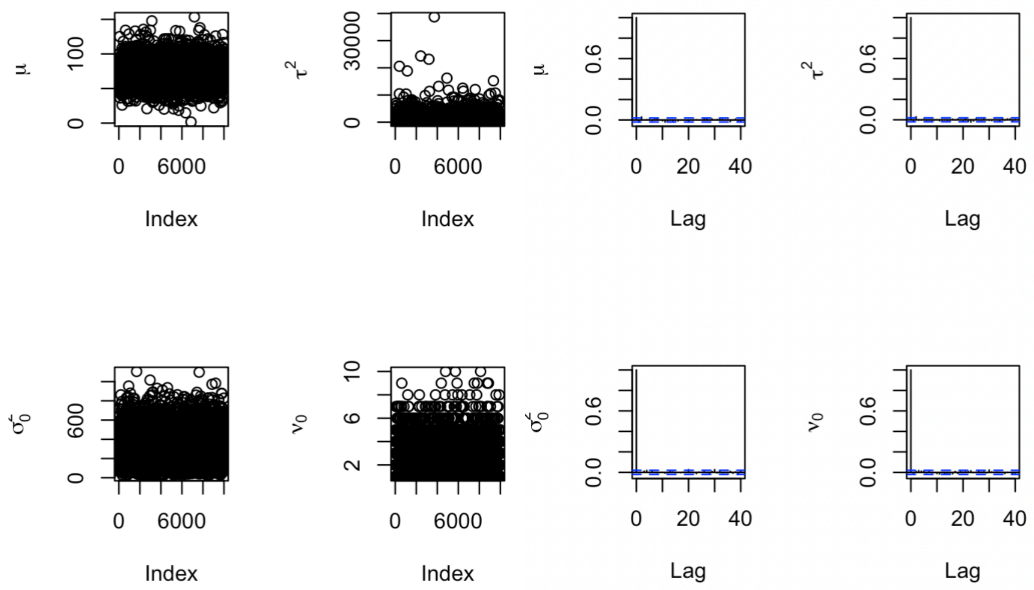
Traceplot ACF



> apply(SIGMA2.pp,2,effectiveSize)
[1] 10000 10000 10000 10000 10000

For parameters: $(\mu,\ \tau^2,\ \sigma_0^2, \nu_0)$

Trace plots: ACF:



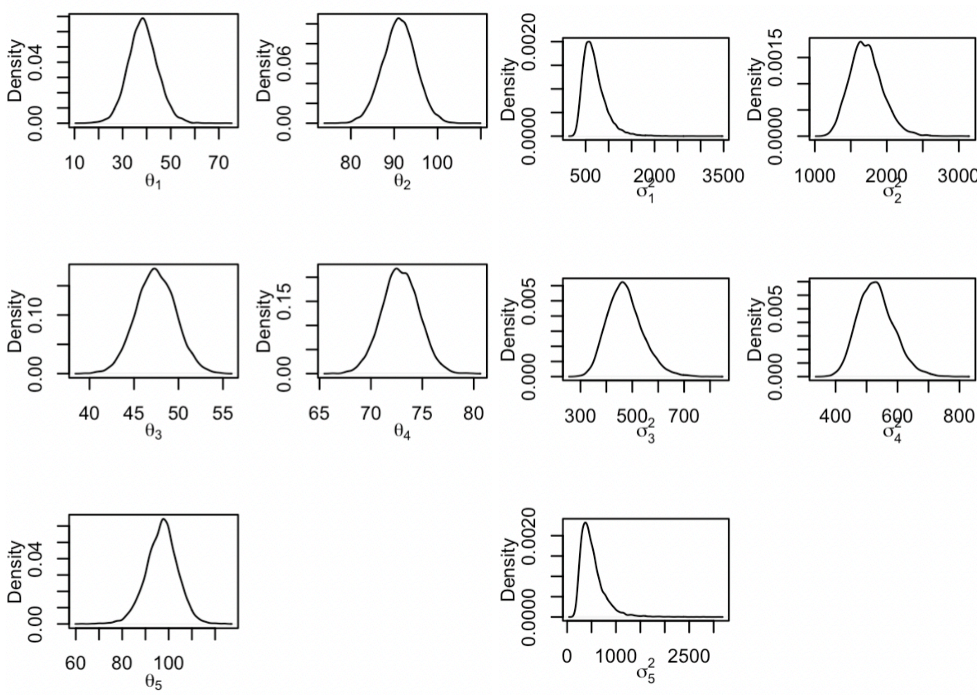
> apply(MTSN.pp,2,effectiveSize)
[1] 9801.13 9678.96 10000.00 10000.00

All trace plots above show convergence as the parameter space are explored evenly. No serious problem in acf plot meaning there is no autocorrelation problem in theta, sigma2 and the parameters. Effective size of draws are also large enough to perform further analysis.

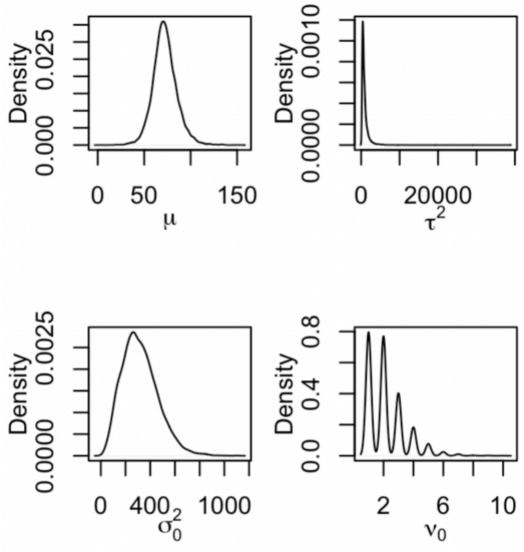
Thus conclude that both colnrent and localpp will be compared against with each other using hierarchical model.

Density plot

THETA SIGMA2

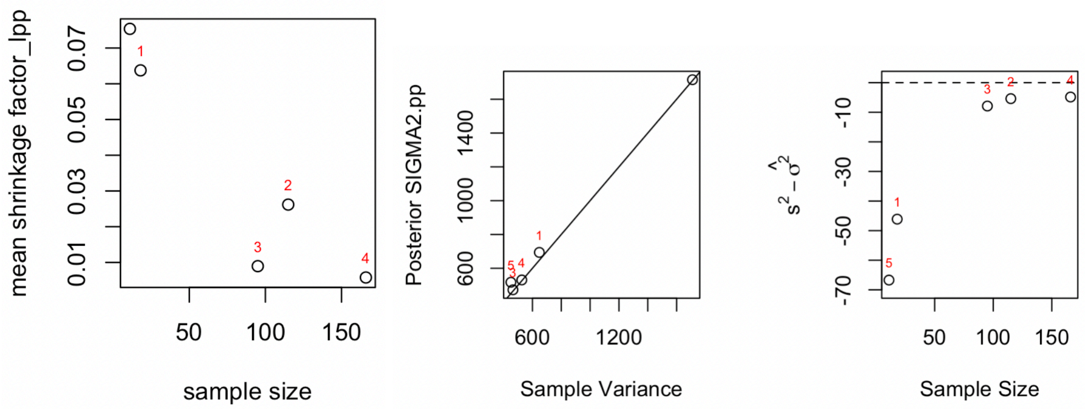


MTSN



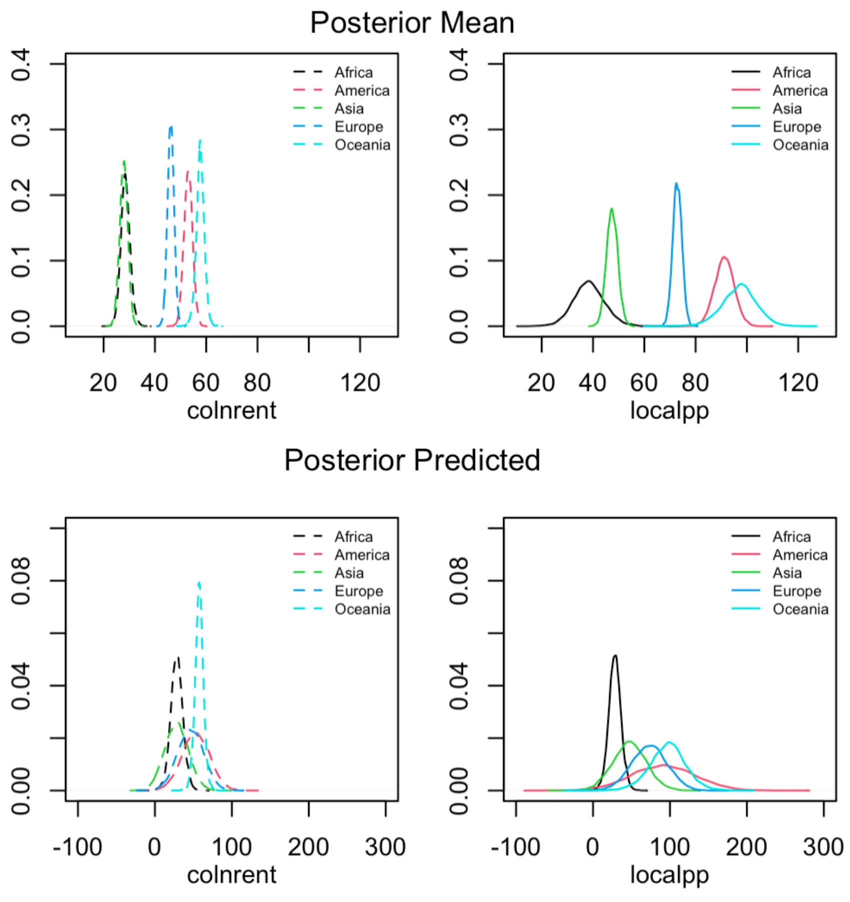

```
> apply(THETA.pp,2,mean)
[1]  38.43116 91.17847 47.44920 72.87207 96.99121

> apply(SIGMA2.pp,2,mean)
[1]  693.9381 1715.5199  472.5908  531.0008  517.4287

> apply(MTSN.pp,2,mean)
[1]  72.24849 934.64211 317.37011   2.17080
```

Shrinkage:



Region 5 shrunk the most towards the overall mean as it has the smallest sample size. Region 2 has a large sample size than 3 but it shrunk more towards the overall mean as it has a very large sample variance (1710.1204). Overall, it is indeed the case the smaller sample size shrinks more towards the overall mean.

C. Comparison



R output:

```
> ###### africa ######
> mean(THETA.pp[,1]>THETA[,1])
[1] 0.9425
> mean(THETA.pp[,1]-THETA[,1])
[1] 9.911729

> ###### america ######
> mean(THETA.pp[,2]>THETA[,2])
[1] 1
> mean(THETA.pp[,2]-THETA[,2])
[1] 38.17752

> ###### asia ######
> mean(THETA.pp[,3]>THETA[,3])
[1] 1
> mean(THETA.pp[,3]-THETA[,3])
[1] 19.63594

> ###### eu ######
> mean(THETA.pp[,4]>THETA[,4])
[1] 1
> mean(THETA.pp[,4]-THETA[,4])
[1] 26.68194

> ####### oceania ######
> mean(THETA.pp[,5]>THETA[,5])
[1] 1
> mean(THETA.pp[,5]-THETA[,5])
[1] 39.29561
```

Posterior mean of localpp are 100% higher than colnrent for 4 regions except Africa with only 95.25%. The difference between localpp and colrent is the largest in Oceania (39.30) and the smallest in Africa (9.91). The difference ranges from 10 to 40.

```
> ###### africa ######
> mean(YS2.h.pp[,1]>YS1.h[,1])
[1] 0.6438
> mean(YS2.h.pp[,1]-YS1.h[,1])
[1] 10.12581

> ###### america ######
> mean(YS2.h.pp[,2]>YS1.h[,2])
[1] 0.8004
> mean(YS2.h.pp[,2]-YS1.h[,2])
[1] 37.93745
```

```
> ###### asia ######
> mean(YS2.h.pp[,3]>YS1.h[,3])
[1] 0.765
> mean(YS2.h.pp[,3]-YS1.h[,3])
[1] 19.43804

> ###### eu ######
> mean(YS2.h.pp[,4]>YS1.h[,4])
[1] 0.8283
> mean(YS2.h.pp[,4]-YS1.h[,4])
[1] 26.7654

> ####### oceania ######
> mean(YS2.h.pp[,5]>YS1.h[,5])
[1] 0.9471
> mean(YS2.h.pp[,5]-YS1.h[,5])
[1] 39.42027
```

For posterior predictive draws of indices, it is less certain if localpp would be greater than colnrent since it incorporates extra uncertainty in sampling. America and Africa have the highest and lowest probability of having a higher localpp than colrent while Oceania and Africa have the largest and smallest difference. The difference ranges from 10 to 39.

4. <u>Conclusions</u>

After Bayesian analysis, it is concluded that:

- **Hierarchical model was chosen** for both cost of living & rent and local purchasing power.
- Looking at the posterior draws of theta which is the mean of the indices, all cities in **America, Asia, Europe, Oceanica have a higher local purchasing power than cost of living & rent.** Only **94.25% cities in Africa** have a higher local purchasing power. The difference between 2 indices ranges from 10 to 40.
- **Posterior predictive draws include extra sampling uncertainty** so the probability of observing local purchasing power higher than cost of living & rent among all 5 cities are lower. Otherwise, conclusion 2 still hold.
- For both indices, if the **sample size is smaller**, the posterior draws will **shrink more towards the overall mean,** unless the **sample variance is large**. Then draws will shrink extra more to the overall mean to compensate such sample variance. Sample cases are Region 1's cost of living & rent and Region 2's local purchasing power.

Choosing hierarchical model over the other 2 models implies that it is true that 5 regions share some information in their mean and variance. Intuitively, less developed countries' cost of living & rent and local purchasing power should be lower and this should be true for all less developed countries.

One limitation is that the sample variance in American local purchasing power is too large that it affected the performance of the model. This is why the prediction error of American local purchasing power is higher than the equal mean model when all other 4 regions are not. This is the main reason why the variance of the posterior predictive draws is so wide, ranging from around -100 to 300 which is practically impossible for the index must be positive.

Although the benefit of hierarchical model is the sharing of information if the sample size is small, the result can still be improved if more sample is available. Sample size of region

1 and 5 are 25 and 16 only. If more samples are available, the prediction error can be further reduced. The limitation can also be alleviated.

Another limitation would be no comparison in posterior draws of sigma2 was shown. This is mainly due to the page limit. Given comparison between posterior mean and posterior predictive draws, it is enough to answer the proposed question but it is more comprehensive to compare sigma2 as well.

Given the complex structure of this report, the code to perform above study is also very complicated and lengthy.

Possible extension of this study is to fit a hierarchical linear model to find what component best explain the cost of living & rent index which allows a more accurate prediction of the index. In this study, ν_0, which is the prior belief on variance of sigma2, is not a standard distribution. Therefore, when running the algorithm, it was approximated by grid search that maximize its likelihood. It might be possible to run a MH algorithm instead to produce posterior draws. Last potential improvement is to try derived and visualize 95% posterior CI of each parameter and predictive draws.

5. Appendices (R code)

```
library(readxl)
africa <-
read_xlsx('./data/all_regions.xlsx',sheet
=1)
america <-
read_xlsx('./data/all_regions.xlsx',sheet
=2)
asia <-
read_xlsx('./data/all_regions.xlsx',sheet
=3)
eu <-
read_xlsx('./data/all_regions.xlsx',sheet
=4)
oceania <-
read_xlsx('./data/all_regions.xlsx',sheet
=5)

#### Cost of living and rent ####

#### Equal Mean ####
Y1.e <- NULL
Y1.e <- rbind(Y1.e,
          c(africa$colnrent
           ,america$colnrent
           ,asia$colnrent
           ,eu$colnrent
           ,oceania$colnrent))
length(Y1.e)

##### split train and test #####
# train index
set.seed(111)
index <- sample(seq(1,578,1),405)

Y1.e.train <- NULL
Y1.e.train <- Y1.e[index]
Y1.e.test <- Y1.e[-index]

n = length(Y1.e.train)
ybar = mean(Y1.e.train)
sv2 = var(Y1.e.train)

n.test = length(Y1.e.test)
ybar.test = mean(Y1.e.test)
sv2.test = var(Y1.e.test)


##### overall mean #####
#Gibbs sampler code
S<-10000

y = Y1.e.train

mu0<-100 ; t20<-40^2  # for mu
s20<-40^2 ; nu0<-1  # for sigma2
# 95% CI for y = 100+ 2*40 = 20
# sigma2 ~ invGamma(1/2, 50)

PHI.e<-matrix(nrow=S,ncol=2)
phi<-c( ybar, sv2)
YS1.e <- NULL

set.seed(112)
### Gibbs sampling
for(s in 1:S) {
 # new theta
 t2n<- 1/( 1/t20 + n/phi[2] )
 mun<-  ( mu0/t20 + n*ybar/phi[2] )*
t2n
 phi[1]<-rnorm(1, mun, sqrt(t2n) )

 # new 1/sigma^2
 nun<- nu0+n
 s2n<- (nu0*s20 + (n-1)*sv2 + n*(ybar-
phi[1])^2 ) /nun
 phi[2]<- 1/rgamma(1, nun/2,
nun*s2n/2)

 ys1.e <- rnorm(1,phi[1],sqrt(phi[2]))


 YS1.e <- rbind(YS1.e,ys1.e)
 PHI.e[s,]<-phi        }
###


##### compare error #####
# general predictive draws
Y.e.pred <- apply(YS1.e,2,mean)
length(Y.e.pred)
mean( (Y1.e.test - Y.e.pred)^2)   # [1]
381.0604


#### Separate Mean ####
Y1.s<- NULL
Y1.s[[1]] <- africa$colnrent
Y1.s[[2]] <- america$colnrent
Y1.s[[3]] <- asia$colnrent
Y1.s[[4]] <- eu$colnrent
Y1.s[[5]] <- oceania$colnrent


##### split train and test #####
# train index
set.seed(121)
index_africa <- sample(seq(1,25,1),18)
## 18/25
index_america <-
sample(seq(1,164,1),115) ## 115/164
index_asia <- sample(seq(1,136,1),95)
## 95/136
index_eu <- sample(seq(1,237,1),166)
## 166/237
index_oceania <-
sample(seq(1,16,1),11) ## 11/16

Y1.s.train <- NULL
Y1.s.train[[1]] <-
africa$colnrent[index_africa]
Y1.s.train[[2]] <-
america$colnrent[index_america]
Y1.s.train[[3]] <-
asia$colnrent[index_asia]
Y1.s.train[[4]] <- eu$colnrent[index_eu]
Y1.s.train[[5]] <-
oceania$colnrent[index_oceania]

Y1.s.test <- NULL
Y1.s.test[[1]] <- africa$colnrent[-
index_africa]
Y1.s.test[[2]] <- america$colnrent[-
index_america]
Y1.s.test[[3]] <- asia$colnrent[-
index_asia]
Y1.s.test[[4]] <- eu$colnrent[-index_eu]
Y1.s.test[[5]] <- oceania$colnrent[-
index_oceania]

J=length(Y1.s)
n <- ybar <- sv2 <- rep(0,J)     # stat for
each regions
for(j in 1:J) {
  ybar[j]<-mean(Y1.s.train[[j]])
  n[j]<-length(Y1.s.train[[j]])
  sv2[j]<-var(Y1.s.train[[j]])
}


#####
S<-10000
mu0<-100 ; t20<-40^2  # for mu
s20<-20^2 ; nu0<-1  # for sigma2
# 95% CI for y = 100+ 2*40 = 20
# sigma2 ~ invGamma(1/2, 50)

YS.s1 <- YS.s2 <- YS.s3 <- YS.s4 <- YS.s5
<- NULL
ys.s1 <- ys.s2 <- ys.s3 <- ys.s4 <- ys.s5 <-
NULL

##### africa ######
Mean1<- Sigma21<-NULL
mean <- ybar[1]; var <- sv2[1]
set.seed(122)
### Gibbs sampling
for(s in 1:S) {
```

```
  # generate a new theta value from its full conditional
  t2n<- 1/( 1/t20 + n[1]/var)
  mun<-  ( mu0/t20 + n[1]*ybar/var)*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value from its full conditional
  nun<- nu0+n[1]
  s2n<- (nu0*s20 + (n[1]-1)*sv2[1] + n[1]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys.s1 <- rnorm(1,mean,sqrt(var))

  YS.s1 <- rbind(YS.s1, ys.s1)
  Mean1 <- rbind(Mean1,mean)
  Sigma21 <- rbind(Sigma21,var)
}


##### america #####
Mean2<- Sigma22<-NULL
mean <- ybar[2]; var <- sv2[2]
set.seed(123)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its full conditional
  t2n<- 1/( 1/t20 + n[2]/var)
  mun<-  ( mu0/t20 + n[2]*ybar/var)*t2n
  mean <- rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value from its full conditional
  nun<- nu0+n[2]
  s2n<- (nu0*s20 + (n[2]-1)*sv2[2] + n[2]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys.s2 <- rnorm(1,mean,sqrt(var))

  YS.s2 <- rbind(YS.s2, ys.s2)
  Mean2 <- rbind(Mean2,mean)
  Sigma22 <- rbind(Sigma22,var)
}


##### asia #####
Mean3<- Sigma23<-NULL
mean <- ybar[3]; var <- sv2[3]
set.seed(124)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its full conditional
  t2n<- 1/( 1/t20 + n[3]/var)
  mun<-  ( mu0/t20 + n[3]*ybar/var)*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value from its full conditional
  nun<- nu0+n[3]
  s2n<- (nu0*s20 + (n[3]-1)*sv2[3] + n[3]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys.s3 <- rnorm(1,mean,sqrt(var))

  YS.s3 <- rbind(YS.s3, ys.s3)
  Mean3 <- rbind(Mean3,mean)
  Sigma23 <- rbind(Sigma23,var)
}


##### eu #####
Mean4<- Sigma24<-NULL
mean <- ybar[4]; var <- sv2[4]
set.seed(125)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its full conditional
  t2n<- 1/( 1/t20 + n[4]/var)
  mun<-  ( mu0/t20 + n[4]*ybar/var)*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value from its full conditional
  nun<- nu0+n[4]
  s2n<- (nu0*s20 + (n[4]-1)*sv2[4] + n[4]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys.s4 <- rnorm(1,mean,sqrt(var))

  YS.s4 <- rbind(YS.s4, ys.s4)
  Mean4 <- rbind(Mean4,mean)
  Sigma24 <- rbind(Sigma24,var)
}


##### oceania #####
Mean5<- Sigma25<-NULL
mean <- ybar[5]; var <- sv2[5]
set.seed(126)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its full conditional
  t2n<- 1/( 1/t20 + n[5]/var)
  mun<-  ( mu0/t20 + n[5]*ybar/var)*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value from its full conditional
  nun<- nu0+n[5]
  s2n<- (nu0*s20 + (n[5]-1)*sv2[5] + n[5]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys.s5 <- rnorm(1,mean,sqrt(var))

  YS.s5 <- rbind(YS.s5, ys.s5)
  Mean5 <- rbind(Mean5,mean)
  Sigma25 <- rbind(Sigma25,var)
}

Mean.s = cbind(Mean1, Mean2, Mean3, Mean4, Mean5)
Sigma2.s = cbind(Sigma21, Sigma22, Sigma23, Sigma24, Sigma25)


###

library(coda)
mean_sep = apply(Mean.s,2,mean)
mean_sep
# [1] 28.66123 28.58981 28.54824 28.51357 29.06416
sig_sep = apply(Sigma2.s,2,mean)
sig_sep
# [1] 85.59536 356.57544 274.22999 262.54768 161.03640


##### compare error #####
# general predictive draws
Y.pred_sep <- NULL
mean_err_sep <- NULL
YS1.s <- cbind(YS.s1,YS.s2,YS.s3,YS.s4,YS.s5)

for(j in 1:J){
  Y.pred_sep <- apply(YS1.s,2,mean)
  mean_err_sep[j] <- mean( (Y1.s.test[[j]] - Y.pred_sep[j])^2)
}
mean_err_sep
# [1]  22.09373 909.72000 271.91258 619.08923 776.33264


#### Hier ####
Y1.h<- NULL
Y1.h[[1]] <- africa$colnrent
```

```
Y1.h[[2]] <- america$colnrent
Y1.h[[3]] <- asia$colnrent
Y1.h[[4]] <- eu$colnrent
Y1.h[[5]] <- oceania$colnrent


##### split train and test #####
# train index
set.seed(131)
index_africa <- sample(seq(1,25,1),18) ## 18/25
index_america <- sample(seq(1,164,1),115) ## 115/164
index_asia <- sample(seq(1,136,1),95) ## 95/136
index_eu <- sample(seq(1,237,1),166) ## 166/237
index_oceania <- sample(seq(1,16,1),11) ## 11/16

Y1.h.train <- NULL
Y1.h.train[[1]] <- africa$colnrent[index_africa]
Y1.h.train[[2]] <- america$colnrent[index_america]
Y1.h.train[[3]] <- asia$colnrent[index_asia]
Y1.h.train[[4]] <- eu$colnrent[index_eu]
Y1.h.train[[5]] <- oceania$colnrent[index_oceania]

Y1.h.test <- NULL
Y1.h.test[[1]] <- africa$colnrent[-index_africa]
Y1.h.test[[2]] <- america$colnrent[-index_america]
Y1.h.test[[3]] <- asia$colnrent[-index_asia]
Y1.h.test[[4]] <- eu$colnrent[-index_eu]
Y1.h.test[[5]] <- oceania$colnrent[-index_oceania]

J=length(Y1.h)
n <- ybar <- sv2 <- rep(0,J)     # stat for each regions
for(j in 1:J) {
  ybar[j]<-mean(Y1.h.train[[j]])
  n[j]<-length(Y1.h.train[[j]])
  sv2[j]<-var(Y1.h.train[[j]])
}


##### mean and variance  vary #####
m<-5

## weakly informative priors
# tau ~ invGamma(1/2, 50)
eta0<-1 ; t20<-100

# mu ~ normal(100, 40)
mu0<-100 ; g20<-40^2

# for sigma2 ~ Gamma(1,1/100) - improper
a0<-1 ; b0<-1/100 ;

# for nu0
alpha<-1


## starting values
theta<-ybar; sigma2<-sv2
mu<-mean(theta)
tau2<-var(theta)
s20<-1/mean(1/sv2)
nu0<-10


##### MCMC #####
S<-50000; odens<-S/10000
## setup MCMC
SIGMA2<-THETA<- YS1.h <- NULL
MTSN<-NULL
nu0s<-1:5000
ys1.h <- mtsn <- NULL

set.seed(132)
for(s in 1:S){
  # thetas - mean of Y
  for(j in 1:5){
    vtheta<-1/(n[j]/sigma2[j]+1/tau2)
    etheta<-vtheta*(ybar[j]*n[j]/sigma2[j] + mu/tau2)
    theta[j]<-rnorm(1,etheta,sqrt(vtheta))
  }

  # sigma2s - var of Y
  for(j in 1:5) {
    nun<-nu0+n[j]
    ss<-nu0*s20+ sum((Y1.h.train[[j]]-theta[j])^2)
    sigma2[j]<-1/rgamma(1,nun/2,ss/2)
  }

  # s20 - var of sigma2s
  s20<-rgamma(1, a0+m*nu0/2, b0+nu0*sum(1/sigma2)/2)

  # nu0s - mean of sigma2s
  lpnu0<- .5*nu0s*m*log(s20*nu0s/2)-m*lgamma(nu0s/2)+(nu0s/2-1)*sum(log(1/sigma2)) -
    nu0s*s20*sum(1/sigma2)/2   - alpha*nu0s

  nu0<-sample(nu0s,1,prob=exp( lpnu0-max(lpnu0)) )

  # mu - mean of thetas
  vmu<- 1/(m/tau2+1/g20)
  emu<- vmu*(m*mean(theta)/tau2 + mu0/g20)
  mu<-rnorm(1,emu,sqrt(vmu))

  # tau2 - var of thetas
  etam<-eta0+m
  ss<- eta0*t20 + sum( (theta-mu)^2 )
  tau2<-1/rgamma(1,etam/2,ss/2)

  for(j in 1:5){
    ys1.h[j] <- rnorm(1,theta[j],sqrt(sigma2[j]))
  }

  mtsn <- c(mu,tau2,s20,nu0)
  if(s%%odens==0){
    #store results
    YS1.h <- rbind(YS1.h, ys1.h)
    THETA <- rbind(THETA, theta)
    SIGMA2<-rbind(SIGMA2, sigma2)
    MTSN <- rbind(MTSN, mtsn)
  }
}


mean_hier = apply(THETA,2,mean)
mean_hier
# [1] 28.52344 53.01722 27.82402 46.20123 57.69117

sig_hier = apply(SIGMA2,2,mean)
sig_hier
# [1]  58.93735 328.32395 241.66792 281.74305  28.57518


# > n
# [1]  18 115  95 166  11

##### compare error #####
# general predictive draws
Y.pred_hier <- NULL
mean_err_hier <- NULL

Y.pred.hier <- apply(YS1.h,2,mean)
Y1.h.err <- NULL
for(j in 1:J){
  Y1.h.err[j] = mean((Y1.h.test[[j]] - Y.pred.hier[j])^2)
```

```
}
Y1.h.err
# [1] 36.77522 335.53593 338.98960
215.17478 197.68126


##### diagnosis #####
par(mfrow=c(3,2))
plot(THETA[,1])
plot(THETA[,2])
plot(THETA[,3])
plot(THETA[,4])
plot(THETA[,5])

par(mfrow=c(3,2))
acf(THETA[,1])
acf(THETA[,2])
acf(THETA[,3])
acf(THETA[,4])
acf(THETA[,5])

par(mfrow=c(3,2))
plot(density(THETA[,1]),main='',
xlab=expression(theta[1]))
plot(density(THETA[,2]),main='',
xlab=expression(theta[2]))
plot(density(THETA[,3]),main='',
xlab=expression(theta[3]))
plot(density(THETA[,4]),main='',
xlab=expression(theta[4]))
plot(density(THETA[,5]),main='',
xlab=expression(theta[5]))

apply(THETA,2,effectiveSize)
apply(THETA,2,mean)

par(mfrow=c(3,2))
plot(SIGMA2[,1])
plot(SIGMA2[,2])
plot(SIGMA2[,3])
plot(SIGMA2[,4])
plot(SIGMA2[,5])

par(mfrow=c(3,2))
acf(SIGMA2[,1])
acf(SIGMA2[,2])
acf(SIGMA2[,3])
acf(SIGMA2[,4])
acf(SIGMA2[,5])

par(mfrow=c(3,2))
plot(density(SIGMA2[,1]),main='',
xlab=expression(sigma[1]^2))
plot(density(SIGMA2[,2]),main='',
xlab=expression(sigma[2]^2))
plot(density(SIGMA2[,3]),main='',
xlab=expression(sigma[3]^2))
plot(density(SIGMA2[,4]),main='',
xlab=expression(sigma[4]^2))
plot(density(SIGMA2[,5]),main='',
xlab=expression(sigma[5]^2))

apply(SIGMA2,2,effectiveSize)
apply(SIGMA2,2,mean)


# MTSN[s,]<-c(mu,tau2,s20,nu0)
par(mfrow=c(2,2))
plot(MTSN[,1],ylab=expression(mu))
plot(MTSN[,2],ylab=expression(tau^2))
plot(MTSN[,3],ylab=expression(sigma[0]
^2))
plot(MTSN[,4],ylab=expression(nu[0]))

par(mfrow=c(2,2))
acf(MTSN[,1],ylab=expression(mu),main
='')
acf(MTSN[,2],ylab=expression(tau^2),m
ain='')
acf(MTSN[,3],ylab=expression(sigma[0]
^2),main='')
acf(MTSN[,4],ylab=expression(nu[0]),m
ain='')

par(mfrow=c(2,2))
plot(density(MTSN[,1]), main='',
xlab=expression(mu))
plot(density(MTSN[,2]), main='',
xlab=expression(tau^2))
plot(density(MTSN[,3]), main='',
xlab=expression(sigma[0]^2))
plot(density(MTSN[,4]), main='',
xlab=expression(nu[0]))

apply(MTSN,2,effectiveSize)
apply(MTSN,2,mean)

##### shrinkage #####
shrink <- factor <- matrix(ncol=S,
nrow=m)

# MTSN[s,]<-c(mu,tau2,s20,nu0)
for(j in 1:m){
  factor[j,] = (SIGMA2[,j]/n[j]) /
(SIGMA2[,j]/n[j] + MTSN[,2])

  shrink[j,] <- (1-factor[j,])*ybar[j] +
factor[j]*MTSN[,1]
}
shrink

mean_factor = apply(factor,1,mean)
par(mfrow=c(1,1))
plot(n,mean_factor,
   xlab = 'sample size', ylab='mean
shrinkage factor')
text(n,mean_factor, 1:5, cex=0.6,
pos=3, col="red")


##### plots (param shrinkage) #####
par(mfrow=c(1,2))
apply(SIGMA2,2,mean) -> sigma2.hat
plot(sv2,sigma2.hat ,
   ylab='Posterior SIGMA2',
   xlab='Sample Variance')
abline(0,1)
text(sv2, sigma2.hat, 1:5, cex=0.6,
pos=3, col="red")

# par(mar = c(5.1, 5, 2, 2.1))
plot(n, sv2-sigma2.hat,ylim=c(-16,1),
   xlab = 'Sample Size',
   ylab=expression(s^2-hat(sigma^2)))
abline(h=0, lty=2)
text(n, sv2-sigma2.hat, 1:5, cex=0.6,
pos=3, col="red")

dev.off()


##################################
########################
#### LPP ####

#### Equal Mean ####
Y2.e <- NULL
Y2.e <- rbind(Y2.e,
         c(africa$localpp
          ,america$localpp
          ,asia$localpp
          ,eu$localpp
          ,oceania$localpp))
length(Y2.e)

##### split train and test #####
# train index
set.seed(212)
index <- sample(seq(1,578,1),405)

Y2.e.train <- NULL
Y2.e.train <- Y2.e[index]
Y2.e.test <- Y2.e[-index]

n = length(Y2.e.train)
ybar = mean(Y2.e.train)
sv2 = var(Y2.e.train)


##### overall mean #####
#Gibbs sampler code
```

```
S<-10000

y = Y2.e.train

mu0<-100 ; t20<-40^2  # for mu
s20<-40^2 ; nu0<-1  # for sigma2
# 95% CI for y = 100+ 2*40 = 20
# sigma2 ~ invGamma(1/2, 50)

PHI1.e<-matrix(nrow=S,ncol=2)
phi1<-c( ybar, sv2)
YS2.e <- ys2.e <- NULL

set.seed(211)
### Gibbs sampling
for(s in 1:S) {
 # new theta
 t2n<- 1/( 1/t20 + n/phi1[2] )
 mun<-  ( mu0/t20 + n*ybar/phi1[2] )*
t2n
 phi1[1]<-rnorm(1, mun, sqrt(t2n) )

 # new 1/sigma^2
 nun<- nu0+n
 s2n<- (nu0*s20 + (n-1)*sv2 + n*(ybar-
phi1[1])^2 ) /nun
 phi1[2]<- 1/rgamma(1, nun/2,
nun*s2n/2)

 ys2.e <- rnorm(1,phi1[1],sqrt(phi1[2]))

 YS2.e <- rbind(YS2.e,ys2.e)
 PHI1.e[s,]<-phi1        }
###


##### compare error #####
# general predictive draws
Y.e.pred <- apply(YS2.e,2,mean)
length(Y.e.pred)
mean( (Y2.e.test - Y.e.pred)^2)   # [1]
1291.827

#### Separate Mean2 ####
Y2.s<- NULL
Y2.s[[1]] <- africa$localpp
Y2.s[[2]] <- america$localpp
Y2.s[[3]] <- asia$localpp
Y2.s[[4]] <- eu$localpp
Y2.s[[5]] <- oceania$localpp


##### split train and test #####
# train index
set.seed(221)
index_africa <- sample(seq(1,25,1),18)
## 18/25
index_america <-
sample(seq(1,164,1),115) ## 115/164
index_asia <- sample(seq(1,136,1),95)
## 95/136
index_eu <- sample(seq(1,237,1),166)
## 166/237
index_oceania <-
sample(seq(1,16,1),11) ## 11/16

Y2.s.train <- NULL
Y2.s.train[[1]] <-
africa$localpp[index_africa]
Y2.s.train[[2]] <-
america$localpp[index_america]
Y2.s.train[[3]] <-
asia$localpp[index_asia]
Y2.s.train[[4]] <- eu$localpp[index_eu]
Y2.s.train[[5]] <-
oceania$localpp[index_oceania]

Y2.s.test <- NULL
Y2.s.test[[1]] <- africa$localpp[-
index_africa]
Y2.s.test[[2]] <- america$localpp[-
index_america]
Y2.s.test[[3]] <- asia$localpp[-
index_asia]
Y2.s.test[[4]] <- eu$localpp[-index_eu]
Y2.s.test[[5]] <- oceania$localpp[-
index_oceania]

J=length(Y2.s)
n <- ybar <- sv2 <- rep(0,J)     # stat for
each regions
for(j in 1:J) {
  ybar[j]<-mean(Y2.s.train[[j]])
  n[j]<-length(Y2.s.train[[j]])
  sv2[j]<-var(Y2.s.train[[j]])
}

#####
S<-10000
mu0<-100 ; t20<-40^2  # for mu
s20<-20^2 ; nu0<-1  # for sigma2
# 95% CI for y = 100+ 2*40 = 20
# sigma2 ~ invGamma(1/2, 50)

YS2.s1 <- YS2.s2 <- YS2.s3 <- YS2.s4 <-
YS2.s5 <- NULL
ys2.s1 <- ys2.s2 <- ys2.s3 <- ys2.s4 <-
ys2.s5 <- NULL

##### africa ######
Mean21<- Sigma2_21<-NULL
mean <- ybar[1]; var <- sv2[1]
set.seed(222)
for(s in 1:S) {
  # generate a new theta value from its
full conditional
  t2n<- 1/( 1/t20 + n[1]/var)
  mun<-  ( mu0/t20 + n[1]*ybar/var)
*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value
from its full conditional
  nun<- nu0+n[1]
  s2n<- (nu0*s20 + (n[1]-1)*sv2[1] +
n[1]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys2.s1 <- rnorm(1,mean,sqrt(var))

  YS2.s1 <- rbind(YS2.s1, ys2.s1)
  Mean21 <- rbind(Mean21,mean)
  Sigma2_21 <- rbind(Sigma2_21,var)
}


##### america #####
Mean22<- Sigma2_22<-NULL
mean <- ybar[2]; var <- sv2[2]
set.seed(223)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its
full conditional
  t2n<- 1/( 1/t20 + n[2]/var)
  mun<-  ( mu0/t20 + n[2]*ybar/var)*t2n
  mean <- rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value
from its full conditional
  nun<- nu0+n[2]
  s2n<- (nu0*s20 + (n[2]-1)*sv2[2] +
n[2]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys2.s2 <- rnorm(1,mean,sqrt(var))

  YS2.s2 <- rbind(YS2.s2, ys2.s2)
  Mean22 <- rbind(Mean22,mean)
  Sigma2_22 <- rbind(Sigma2_22,var)
}


##### asia #####
Mean23<- Sigma2_23<-NULL
mean <- ybar[3]; var <- sv2[3]
set.seed(224)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its
full conditional
```

```
  t2n<- 1/( 1/t20 + n[3]/var)
  mun<-  ( mu0/t20 + n[3]*ybar/var)
*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value
from its full conditional
  nun<- nu0+n[3]
  s2n<- (nu0*s20 + (n[3]-1)*sv2[3] +
n[3]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys2.s3 <- rnorm(1,mean,sqrt(var))

  YS2.s3 <- rbind(YS2.s3, ys2.s3)
  Mean23 <- rbind(Mean23,mean)
  Sigma2_23 <- rbind(Sigma2_23,var)
}


##### eu #####
Mean24<- Sigma2_24<-NULL
mean <- ybar[4]; var <- sv2[4]
set.seed(225)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its
full conditional
  t2n<- 1/( 1/t20 + n[4]/var)
  mun<-  ( mu0/t20 + n[4]*ybar/var)*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value
from its full conditional
  nun<- nu0+n[4]
  s2n<- (nu0*s20 + (n[4]-1)*sv2[4] +
n[4]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys2.s4 <- rnorm(1,mean,sqrt(var))

  YS2.s4 <- rbind(YS2.s4, ys2.s4)
  Mean24 <- rbind(Mean24,mean)
  Sigma2_24 <- rbind(Sigma2_24,var)
}


##### oceania #####
Mean25<- Sigma2_25<-NULL
mean <- ybar[5]; var <- sv2[5]
set.seed(226)
### Gibbs sampling
for(s in 1:S) {
  # generate a new theta value from its
full conditional
  t2n<- 1/( 1/t20 + n[5]/var)
  mun<-  ( mu0/t20 + n[5]*ybar/var)
*t2n
  mean<-rnorm(1, mun, sqrt(t2n) )

  # generate a new 1/sigma^2 value
from its full conditional
  nun<- nu0+n[5]
  s2n<- (nu0*s20 + (n[5]-1)*sv2[5] +
n[5]*(ybar-mean)^2 ) /nun
  var<- 1/rgamma(1, nun/2, nun*s2n/2)

  ys2.s5 <- rnorm(1,mean,sqrt(var))

  YS2.s5 <- rbind(YS2.s5, ys2.s5)
  Mean25 <- rbind(Mean25,mean)
  Sigma2_25 <- rbind(Sigma2_25,var)
}

Mean2.s = cbind(Mean21, Mean22,
Mean23, Mean24, Mean25)
Sigma2_2.s = cbind(Sigma2_21,
Sigma2_22, Sigma2_23, Sigma2_24,
Sigma2_25)


###
library(coda)
mean_sep = apply(Mean2.s,2,mean)
mean_sep
# [1] 36.63033 35.87715 35.54366
35.45861 37.12423
sig_sep = apply(Sigma2_2.s,2,mean)
sig_sep
# [1] 569.3414 1726.0820  567.1149
530.9891  520.3140


##### compare error #####
# general predictive draws
Y.pred_sep <- NULL
mean_err_sep <- NULL
YS2.s <-
cbind(YS2.s1,YS2.s2,YS2.s3,YS2.s4,YS2.s
5)

for(j in 1:J){
  Y.pred_sep <- apply(YS2.s,2,mean)
  mean_err_sep[j] <-
mean( (Y2.s.test[[j]] - Y.pred_sep[j])^2)
}
mean_err_sep
# [1]  538.6271 5241.2049  516.8833
2234.0655 4650.3766


#### Hier ####
Y2.h<- NULL
Y2.h[[1]] <- africa$localpp
Y2.h[[2]] <- america$localpp
Y2.h[[3]] <- asia$localpp
Y2.h[[4]] <- eu$localpp
Y2.h[[5]] <- oceania$localpp


##### split train and test #####
# train index
set.seed(231)
index_africa <- sample(seq(1,25,1),18)
## 18/25
index_america <-
sample(seq(1,164,1),115) ## 115/164
index_asia <- sample(seq(1,136,1),95)
## 95/136
index_eu <- sample(seq(1,237,1),166)
## 166/237
index_oceania <-
sample(seq(1,16,1),11) ## 11/16

Y2.h.train <- NULL
Y2.h.train[[1]] <-
africa$localpp[index_africa]
Y2.h.train[[2]] <-
america$localpp[index_america]
Y2.h.train[[3]] <-
asia$localpp[index_asia]
Y2.h.train[[4]] <- eu$localpp[index_eu]
Y2.h.train[[5]] <-
oceania$localpp[index_oceania]

Y2.h.test <- NULL
Y2.h.test[[1]] <- africa$localpp[-
index_africa]
Y2.h.test[[2]] <- america$localpp[-
index_america]
Y2.h.test[[3]] <- asia$localpp[-
index_asia]
Y2.h.test[[4]] <- eu$localpp[-index_eu]
Y2.h.test[[5]] <- oceania$localpp[-
index_oceania]

J=length(Y2.h)
n <- ybar <- sv2 <- rep(0,J)     # stat for
each regions
for(j in 1:J) {
  ybar[j]<-mean(Y2.h.train[[j]])
  n[j]<-length(Y2.h.train[[j]])
  sv2[j]<-var(Y2.h.train[[j]])
}


##### mean and variance  vary #####
m<-5

## weakly informative priors
```

```
# tau ~ invGamma(1/2, 50)
eta0<-1 ; t20<-100

# mu ~ normal(100, 40)
mu0<-100 ; g20<-40^2

# for sigma2 ~ Gamma(1,1/100) - improper
a0<-1 ; b0<-1/100 ;

# for nu0
alpha<-1


## starting values
theta<-ybar; sigma2<-sv2
mu<-mean(theta)
tau2<-var(theta)
s20<-1/mean(1/sv2)
nu0<-10


##### MCMC #####
set.seed(232)
S<-50000; odens<-S/10000
## setup MCMC
SIGMA2.pp<-THETA.pp<- YS2.h.pp <- NULL
MTSN.pp<-NULL
nu0s<-1:5000
ys2.h <- mtsn <- NULL

for(s in 1:S){
 # thetas - mean of Y
 for(j in 1:5){
  vtheta<-1/(n[j]/sigma2[j]+1/tau2)
  etheta<-vtheta*(ybar[j]*n[j]/sigma2[j] + mu/tau2)
  theta[j]<-rnorm(1,etheta,sqrt(vtheta))
 }

 # sigma2s - var of Y
 for(j in 1:5) {
  nun<-nu0+n[j]
  ss<-nu0*s20+ sum((Y2.h.train[[j]]-theta[j])^2)
  sigma2[j]<-1/rgamma(1,nun/2,ss/2)
 }

 # s20 - var of sigma2s
 s20<-rgamma(1, a0+m*nu0/2, b0+nu0*sum(1/sigma2)/2)

 # nu0s - mean of sigma2s
 lpnu0<- .5*nu0s*m*log(s20*nu0s/2)-m*lgamma(nu0s/2)+(nu0s/2-1)*sum(log(1/sigma2)) -
  nu0s*s20*sum(1/sigma2)/2  - alpha*nu0s

 nu0<-sample(nu0s,1,prob=exp( lpnu0-max(lpnu0)) )

 # mu - mean of thetas
 vmu<- 1/(m/tau2+1/g20)
 emu<- vmu*(m*mean(theta)/tau2 + mu0/g20)
 mu<-rnorm(1,emu,sqrt(vmu))

 # tau2 - var of thetas
 etam<-eta0+m
 ss<- eta0*t20 + sum( (theta-mu)^2 )
 tau2<-1/rgamma(1,etam/2,ss/2)

 for(j in 1:5){
  ys2.h[j] <- rnorm(1,theta[j],sqrt(sigma2[j]))
 }

 mtsn <- c(mu,tau2,s20,nu0)
 if(s%%odens==0){
  #store results
  YS2.h.pp <- rbind(YS2.h.pp, ys2.h)
  THETA.pp <- rbind(THETA.pp, theta)
  SIGMA2.pp<-rbind(SIGMA2.pp, sigma2)
  MTSN.pp <- rbind(MTSN.pp, mtsn)
 }
}


mean_hier = apply(THETA.pp,2,mean)
mean_hier
# [1] 38.43116 91.17847 47.44920 72.87207 96.99121

sig_hier = apply(SIGMA2.pp,2,mean)
sig_hier
# [1] 693.9381 1715.5199  472.5908 531.0008  517.4287


# > n
# [1]  18 115  95 166  11

##### compare error #####
# general predictive draws
Y.pred_hier <- NULL
mean_err_hier <- NULL

Y.pred.hier <- apply(YS2.h.pp,2,mean)
Y2.h.err <- NULL
for(j in 1:J){
 Y2.h.err[j] = mean((Y2.h.test[[j]] - Y.pred.hier[j])^2)
}
Y2.h.err
# [1] 263.3537 1419.0438  616.3927 630.2744  122.7762


##### diagnosis #####
par(mfrow=c(3,2))
plot(THETA.pp[,1])
plot(THETA.pp[,2])
plot(THETA.pp[,3])
plot(THETA.pp[,4])
plot(THETA.pp[,5])

par(mfrow=c(3,2))
acf(THETA.pp[,1])
acf(THETA.pp[,2])
acf(THETA.pp[,3])
acf(THETA.pp[,4])
acf(THETA.pp[,5])

par(mfrow=c(3,2))
plot(density(THETA.pp[,1]),main='', xlab=expression(theta[1]))
plot(density(THETA.pp[,2]),main='', xlab=expression(theta[2]))
plot(density(THETA.pp[,3]),main='', xlab=expression(theta[3]))
plot(density(THETA.pp[,4]),main='', xlab=expression(theta[4]))
plot(density(THETA.pp[,5]),main='', xlab=expression(theta[5]))

apply(THETA.pp,2,effectiveSize)
apply(THETA.pp,2,mean)

par(mfrow=c(3,2))
plot(SIGMA2.pp[,1])
plot(SIGMA2.pp[,2])
plot(SIGMA2.pp[,3])
plot(SIGMA2.pp[,4])
plot(SIGMA2.pp[,5])

par(mfrow=c(3,2))
acf(SIGMA2.pp[,1])
acf(SIGMA2.pp[,2])
acf(SIGMA2.pp[,3])
acf(SIGMA2.pp[,4])
acf(SIGMA2.pp[,5])

par(mfrow=c(3,2))
plot(density(SIGMA2.pp[,1]),main='', xlab=expression(sigma[1]^2))
```

```
plot(density(SIGMA2.pp[,2]),main='',
xlab=expression(sigma[2]^2))
plot(density(SIGMA2.pp[,3]),main='',
xlab=expression(sigma[3]^2))
plot(density(SIGMA2.pp[,4]),main='',
xlab=expression(sigma[4]^2))
plot(density(SIGMA2.pp[,5]),main='',
xlab=expression(sigma[5]^2))

apply(SIGMA2.pp,2,effectiveSize)
apply(SIGMA2.pp,2,mean)


# MTSN.pp[s,]<-c(mu,tau2,s20,nu0)
par(mfrow=c(2,2))
plot(MTSN.pp[,1],ylab=expression(mu))
plot(MTSN.pp[,2],ylab=expression(tau^
2))
plot(MTSN.pp[,3],ylab=expression(sigm
a[0]^2))
plot(MTSN.pp[,4],ylab=expression(nu[0]
))

par(mfrow=c(2,2))
acf(MTSN.pp[,1],ylab=expression(mu),
main='')
acf(MTSN.pp[,2],ylab=expression(tau^2)
,main='')
acf(MTSN.pp[,3],ylab=expression(sigma
[0]^2),main='')
acf(MTSN.pp[,4],ylab=expression(nu[0])
,main='')

par(mfrow=c(2,2))
plot(density(MTSN.pp[,1]), main='',
xlab=expression(mu))
plot(density(MTSN.pp[,2]), main='',
xlab=expression(tau^2))
plot(density(MTSN.pp[,3]), main='',
xlab=expression(sigma[0]^2))
plot(density(MTSN.pp[,4]), main='',
xlab=expression(nu[0]))

apply(MTSN.pp,2,effectiveSize)
apply(MTSN.pp,2,mean)


##### shrinkage #####
shrink_lpp <- factor_lpp <-
matrix(ncol=S, nrow=m)

# MTSN_lpp[s,]<-c(mu,tau2,s20,nu0)

for(j in 1:m){
 factor_lpp[j,] = (SIGMA2.pp[,j]/n[j]) /
(SIGMA2.pp[,j]/n[j] + MTSN.pp[,2])

 shrink_lpp[j,] <- (1-
factor_lpp[j,])*ybar[j] +
factor_lpp[j]*MTSN.pp[,1]
}
shrink_lpp

mean_factor =
apply(factor_lpp,1,mean)
par(mfrow=c(1,1))
plot(n,mean_factor,
   xlab = 'sample size', ylab='mean
shrinkage factor_lpp')
text(n,mean_factor, 1:5, cex=0.6,
pos=3, col="red")


##### plots (param shrinkage) #####
par(mfrow=c(1,2))
apply(SIGMA2.pp,2,mean) ->
sigma2_lpp.hat
plot(sv2,sigma2_lpp.hat ,
   ylab='Posterior SIGMA2.pp',
   xlab='Sample Variance')
abline(0,1)
text(sv2, sigma2_lpp.hat, 1:5, cex=0.6,
pos=3, col="red")


par(mar = c(5.1, 5, 2, 2.1))
plot(n, sv2-sigma2_lpp.hat,ylim=c(-
70,1),
   xlab = 'Sample Size',
   ylab=expression(s^2-hat(sigma^2)))
abline(h=0, lty=2)
text(n, sv2-sigma2_lpp.hat, 1:5, cex=0.6,
pos=3, col="red")


##################################
##############################
#### Comparison ####
# colnrent
  # #store results
  # YS1.h <- rbind(YS1.h, ys1.h)
  # THETA <- rbind(THETA, theta)
  # SIGMA2<-rbind(SIGMA2, sigma2)
  # MTSN <- rbind(MTSN, mtsn)
# lpp
  # #store results
  # YS2.h.pp <- rbind(YS2.h.pp, ys2.h)
  # THETA.pp <- rbind(THETA.pp, theta)
  # SIGMA2.pp<-rbind(SIGMA2.pp,
sigma2)
  # MTSN.pp <- rbind(MTSN.pp, mtsn)

##### THETA #####
###### africa ######
mean(THETA.pp[,1]>THETA[,1])
mean(THETA.pp[,1]-THETA[,1])
# 94.25% cities in africa have a higher
pp than colnrent by mean of 9.911729

###### america ######
mean(THETA.pp[,2]>THETA[,2])
mean(THETA.pp[,2]-THETA[,2])
# all cities have a higher pp by mean of
38.1775

###### asia ######
mean(THETA.pp[,3]>THETA[,3])
mean(THETA.pp[,3]-THETA[,3])
# all cities have a higher pp by mean of
19.63594

###### eu ######
mean(THETA.pp[,4]>THETA[,4])
mean(THETA.pp[,4]-THETA[,4])
# all cities have a higher pp by mean of
26.68194

####### oceania ######
mean(THETA.pp[,5]>THETA[,5])
mean(THETA.pp[,5]-THETA[,5])
# all cities have a higher pp by mean of
39.29561

###### all colnrent ######
plot(density(THETA[,1]),xlim=c(15,70),c
ol=1,ylim=c(0,0.4),
    main='Densities of colnrent')
lines(density(THETA[,2]),col=2)
lines(density(THETA[,3]),col=3)
lines(density(THETA[,4]),col=4)
lines(density(THETA[,5]),col=5)
legend('topleft',legend=c('Africa','Ameri
ca','Asia','Europe','Oceania'),
     lty=1:1, col=1:5, cex=0.7)

###### all localpp ######
plot(density(THETA.pp[,1]),xlim=c(10,13
0),col=1,ylim=c(0,0.3),
    main='Densities of localpp')
lines(density(THETA.pp[,2]),col=2)
lines(density(THETA.pp[,3]),col=3)
lines(density(THETA.pp[,4]),col=4)
lines(density(THETA.pp[,5]),col=5)
legend('topleft',legend=c('Africa','Ameri
ca','Asia','Europe','Oceania'),
     lty=1:1, col=1:5, cex=0.7)
```

```
##### YS #####
###### africa ######
mean(YS2.h.pp[,1]>YS1.h[,1])
mean(YS2.h.pp[,1]-YS1.h[,1])
# 64.28% cities in africa have a higher predicted pp
# than colnrent by mean of 10.12581

###### america ######
mean(YS2.h.pp[,2]>YS1.h[,2])
mean(YS2.h.pp[,2]-YS1.h[,2])
# 80.04% cities have a higher predicted pp by mean of 37.93745

###### asia ######
mean(YS2.h.pp[,3]>YS1.h[,3])
mean(YS2.h.pp[,3]-YS1.h[,3])
# 76.5% cities have a higher predicted pp by mean of 19.43804

###### eu ######
mean(YS2.h.pp[,4]>YS1.h[,4])
mean(YS2.h.pp[,4]-YS1.h[,4])
# 82.83% cities have a higher predicted pp by mean of 26.7654

####### oceania ######
mean(YS2.h.pp[,5]>YS1.h[,5])
mean(YS2.h.pp[,5]-YS1.h[,5])
# 94.71% cities have a higher predicted pp by mean of 39.42027

###### all colnrent ######
plot(density(YS1.h[,1]),xlim=c(-50,140),
    col='black',ylim=c(0,0.1),
    main='Densities of \nposterior predicted colnrent')
lines(density(YS1.h[,2]),col='red')
lines(density(YS1.h[,3]),col=3)
lines(density(YS1.h[,4]),col=4)
lines(density(YS1.h[,5]),col=5)
legend('topleft',legend=c('Africa','America','Asia','Europe','Oceania'),
     lty=1:1, col=1:5, cex=0.7)

###### all localpp ######
plot(density(YS2.h.pp[,1]),xlim=c(-100,300),col=1,ylim=c(0,0.03),
    main='Densities of \nposterior predicted localpp')
lines(density(YS2.h.pp[,2]),col=2)
lines(density(YS2.h.pp[,3]),col=3)
lines(density(YS2.h.pp[,4]),col=4)
lines(density(YS2.h.pp[,5]),col=5)
legend('topleft',legend=c('Africa','America','Asia','Europe','Oceania'),
     lty=1:1, col=1:5, cex=0.7)


###### in 1 plot  ######
par(mfrow=c(2,2),mar=c(4,3,2,0.5),mgp=c(1.9,1,0))
plot(density(THETA[,1]),xlim=c(10,130),
    col=1, ylim=c(0,0.4),lty=2,main=''
    ,xlab='colnrent',ylab='')
lines(density(THETA[,2]),col=2,lty=2)
lines(density(THETA[,3]),col=3,lty=2)
lines(density(THETA[,4]),col=4,lty=2)
lines(density(THETA[,5]),col=5,lty=2)

legend('topright',bty='n', xpd=NA,

legend=c('Africa','America','Asia','Europe','Oceania'),
     col=1:5, lty=2:2, cex=0.6)

mtext("Posterior Mean", side = 3, line = -1.5, outer = TRUE)


plot(density(THETA.pp[,1]),xlim=c(10,130),
    col=1, ylim=c(0,0.4),
    main='',xlab='localpp',ylab='')
lines(density(THETA.pp[,2]),col=2)
lines(density(THETA.pp[,3]),col=3)
lines(density(THETA.pp[,4]),col=4)
lines(density(THETA.pp[,5]),col=5)
legend('topright',bty='n', xpd=NA,

legend=c('Africa','America','Asia','Europe','Oceania'),
      col=1:5, lty=1:1, cex=0.6)


plot(density(YS1.h[,1]),xlim=c(-100,300),ylim=c(0,0.1),
    col=1, xlab='colnrent',ylab='',
    main='',lty=2)
lines(density(YS1.h[,2]),col=2,lty=2)
lines(density(YS1.h[,3]),col=3,lty=2)
lines(density(YS1.h[,4]),col=4,lty=2)
lines(density(YS1.h[,5]),col=5,lty=2)

legend('topright',bty='n', xpd=NA,

legend=c('Africa','America','Asia','Europe','Oceania'),
      col=1:5,lty=2:2, cex=0.6)


mtext("Posterior Predicted", line = -16, outer = TRUE)


plot(density(YS1.h[,1]),xlim=c(-100,300),ylim=c(0,0.1),
    col=1, xlab='localpp',
    main='',ylab='')
# lines(density(YS2.h.pp[,1]),col='grey3')
lines(density(YS2.h.pp[,2]),col=2)
lines(density(YS2.h.pp[,3]),col=3)
lines(density(YS2.h.pp[,4]),col=4)
lines(density(YS2.h.pp[,5]),col=5)

legend('topright',bty='n', xpd=NA,

legend=c('Africa','America','Asia','Europe','Oceania'),
      col=1:5, lty=1:1, cex=0.6)
```

6. **Reference**

Cost of Living Index by City 2022, *Numbeo*, Retrieved from: https://www.numbeo.com/cost-of-living/rankings.jsp?title=2022