---

title: "version2\_parallel"

author: "Linyi Guo"

date: "2020/1/11"

output: html\_document

---

```{r}

# install.packages("parallel")

library(parallel)

library(forecast)

```

\*\*This part is to define funtions\*\*

```{r}

simulation1 <- function(length){

model <- Arima(ts(rnorm(120),start=c(1980,01),frequency =12), order=c(1,1,1),

seasonal=c(0,1,1), fixed=c(phi=runif(1), theta=runif(1),

Theta=runif(1))

)

data <- simulate(model, nsim=length)

# because if we need to take log later, data must be positive

if(min(data) <= 0) data <- data - min(data) + runif(1)

else data <- data

return(data)

}

simlist1 <- function(n,length) {

Datalist <- list()

for (i in 1:n) Datalist[[i]] <- simulation1(length)

return(Datalist)

}

fun1 <- function(x){

library(seasonal)

seas(x, x11='')

}

preprocess <- function(x11) {

if(transformfunction(x11) == 'log')

data <- log(series(x11, 'b1'))

else

data <- series(x11, 'b1')

return(data)

}

# put previous functions 'exhaustion1' and 'Dif1' together

exhaustion1 <- function(data){

Difference <- c()

index <- c()

x11 <- seas(data, x11='')

for (i in 1:100) {

for (j in 1:100) {

ssmm <- SSModel(data ~ SSMtrend(1, Q=list(j\*0.2)) +

SSMseasonal(12, sea.type = 'dummy', Q = 1),

H = i\*0.2)

ssm <- KFS(ssmm)

sigma <- c(i\*0.2, j\*0.2, 1)

### difference ###

x11\_trend <- series(x11, 'd12')

x11\_seasonal <- series(x11, 'd10')

x11\_irregular <- series(x11, 'd13')

ssm\_trend <- coef(ssm, states = 'trend')

ssm\_seasonal <- -rowSums(coef(ssm, states='seasonal'))

ssm\_irregular <- data[-1] - ssm\_trend[-1] - ssm\_seasonal[-length(data)]

D <- sum((x11\_irregular[-1]-ssm\_irregular)^2)/sigma[1] +

sum((x11\_trend-ssm\_trend)^2)/sigma[2] +

sum((x11\_seasonal[-1]-ssm\_seasonal[-length(data)])^2)/sigma[3]

### end ###

Difference <- c(Difference, D)

index <- rbind(index, sigma)

}

}

df <- data.frame(variance=index, difference = Difference)

return(df)

}

```

\*\*Simulation\*\*

```{r}

set.seed(1)

# 400 is the number of datasets and 180 is the length for each one

datalist2 <- simlist1(400, 180)

```

\*\*Parallel Processing\*\*

```{r}

# I put 4 cores here

# you could check the cores of your PC and set up this number by yourself

# detectCores()

cl <- makeCluster(4)

# Build the package environment for each core

clusterEvalQ(cl,{

library(seasonal)

library(KFAS)

}

)

# Running

x11list2 <- parLapply(cl, datalist2, fun1 )

Datalist2 <- parLapply(cl, x11list2, preprocess )

idevallist2 <- parLapply(cl, Datalist2, exhaustion1)

idevalmat2 <- c()

for (i in 1:100){

ideval <- idevallist2[[i]][which.min(idevallist2[[i]]$difference),c(1,2)]

idevalmat2 <- rbind(idevalmat2, ideval)

}

rownames(idevalmat2) <- c(1:100)

write.csv(idevalmat2, "... ...//idevalmat2.csv")

# Stop parallel processing

stopCluster(cl)

```