Spatio Temporal Data Analysis

Hyeonki Seo

Due: Nov 1st 2021

1 Fit a linear model relating rent per RPM2 to the covariates

Fit a linear model without Locations and Room1 indicator. Because Location parameters are treated later in spatial models, and Room 1 is the baseline. If Room1 indicator is included, Room indicator columns become dependent. Codes and coefficients are below.

```
linmod <- lm(RentPerM2~.-Location-Room1, rents)
summary(linmod)
```

Listing 1: codes for linear model

	Intercept	Year	NoHotWater	NoCentralHeat	NoBathTiles	SpecialBathroom
Coefficient	-16.01	0.01	-1.87	-1.23	-0.73	0.66
	SpecialKitchen	Room2	Room3	Room4	Room5	Room6
Coefficient	1.46	-1.32	-1.89	-2.46	-2.38	-2.48

Table 1: Coefficient for simple linear model

2 Create an nb object with neighbors, add the parks shaded a different color

Make neighbor list by poly2nb functions and plot them

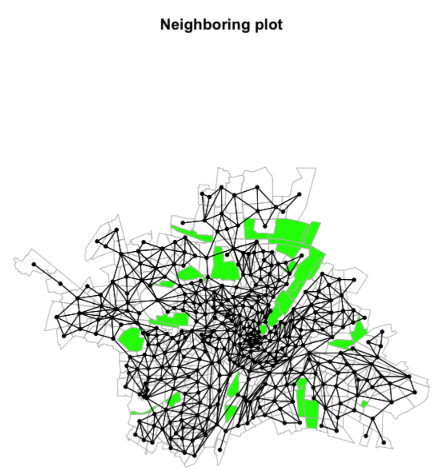

Figure 1: Neighbor plot

3 plot number of observations in each district

Sum each column of H matrix means the number of observations of each district.

```r
pal <- two.colors(n=5, start="white", end="black", middle="grey", alpha=1.0)
q <- classIntervals(colSums(H), n = 5, style = "quantile")
col <- findColours(q, pal)
plot(districts.sp, col = col)
legend("topright", fill = attr(col, "palette"),
       legend = names(attr(col, "table")),
       bty="n", cex = 0.8, y.intersp = 1.5)
title(main = "number of apartment for each district")
```

Listing 2: codes for linear model

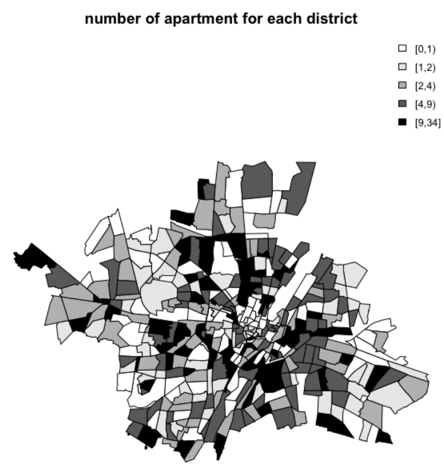

Figure 2: Number of observations

4

4.1 Trace plots and ACF plots for σ^2 and τ^2

I give 100 burn-in in this MCMC samples. σ^2 converged well and its ACF is good. However, τ^2 did not converge well and have large ACF plot.

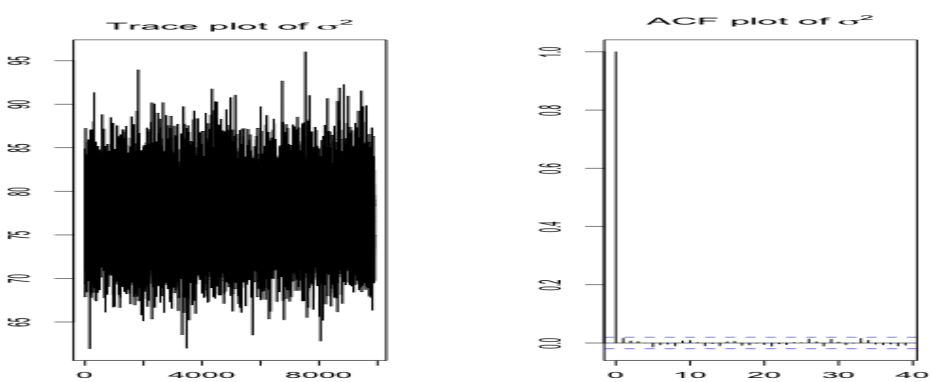

Figure 3: Trace plot and ACF plot for σ^2

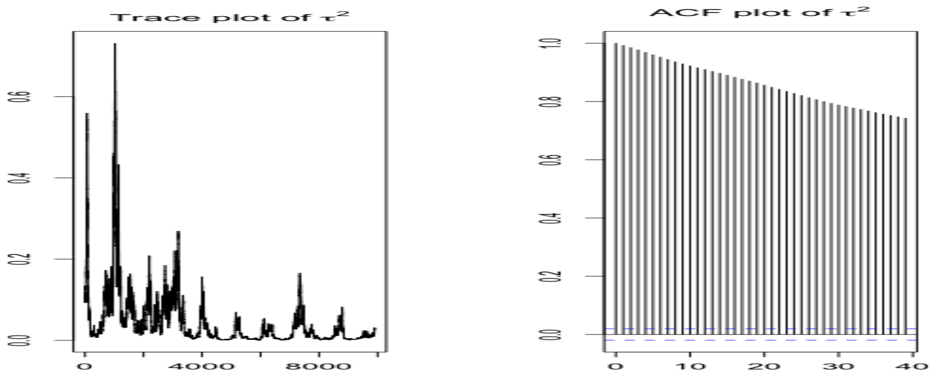

Figure 4: Trace plot and ACF plot for σ^2

4.2 posterior means and credible interval of β's

most of beta coefficients are close to zero. but its 95% credible intervals are different. NoHotWater, Room5 and Room6 indicator have high variances.

```
                     mean      2.5%   97.5%
(Intercept)        0.3031  -32.4196 33.1704
Year              -0.0002   -0.0168  0.0166
NoHotWater        -0.0191   -2.4924  2.4087
NoCentralHeat     -0.0009   -1.6486  1.6357
NoBathTiles       -0.0036   -0.9939  0.9924
SpecialBathroom   -0.0056   -1.3865  1.3638
SpecialKitchen     0.0056   -1.4941  1.5275
Room2              0.0068   -1.2701  1.2574
Room3              0.0054   -1.2556  1.2602
Room4              0.0082   -1.5652  1.5478
Room5              0.0318   -2.6721  2.7834
Room6              0.0407   -4.7484  4.9333
```

Listing 3: means and credible interval of β

4.3 plot for posterior means and standard deviation for vector η

Latent process have high values in north regions and Southern east regions. and it has high variances in outer areas.

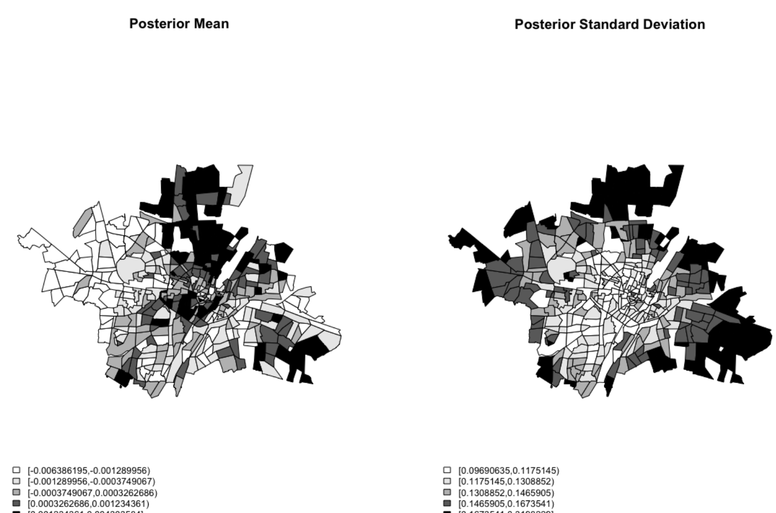

Figure 5: posterior mean and sd for η

Rcode

```
#######################################
############### HW3 ##################
#######2021311169 Hyeonki Seo#########
#######################################

load("munichrents.RData")
head(rents)
coords
##################################
########### 1.###################
##################################


linmod <- lm(RentPerM2~.-Location-Room1, rents)
summary(linmod)

##################################
########### 2.###################
##################################

districts.sp@polygons[[359]]@Polygons[[1]]@labpt <-  districts.sp@polygons[[359]]@Polygons[[2]]@labpt
districts.sp@polygons[[359]]@Polygons[[1]]@area <- districts.sp@polygons[[359]]@Polygons[[2]]@area
districts.sp@polygons[[359]]@Polygons[[1]]@hole <- districts.sp@polygons[[359]]@Polygons[[2]]@hole
districts.sp@polygons[[359]]@Polygons[[1]]@ringDir <- districts.sp@polygons[[359]]@Polygons[[2]]@ring
districts.sp@polygons[[359]]@Polygons[[1]]@coords <- districts.sp@polygons[[359]]@Polygons[[2]]@coord
library(spdep)
library(rgeos)

# neighbor lists
nb.bound <- poly2nb(districts.sp)
coord = coordinates(districts.sp)
coord
print(nb.bound) # spatial neighbors list object
nb.bound[[1]] # Neighbors; no neighbors indicated by a 0
summary(nb.bound)

par(mfrow=c(1,1),mar=c(1,0,2,0))
plot(districts.sp, border = "gray")
plot(parks.sp, col="green", border="gray", add = TRUE)
plot(nb.bound, coord, pch = 19, cex = 0.6, add = TRUE)
title(main = "Neighboring plot")
coords

##################################
########### 3.###################
##################################
rents$Location
library(fields)
library(classInt)
dim(H)
head(rents)

pal <- two.colors(n=5, start="white", end="black", middle="grey", alpha=1.0)
q <- classIntervals(colSums(H), n = 5, style = "quantile")
col <- findColours(q, pal)
```

```
plot(districts.sp, col = col)
legend("topright", fill = attr(col, "palette"),
       legend = names(attr(col, "table")),
       bty="n", cex = 0.8, y.intersp = 1.5)
title(main = "number of apartment for each district")

#################################
########### 4.###################
#################################

# Create W, Dw
help(nb2mat)
W <- nb2mat(nb.bound, style="B")
D <- diag(rowSums(W))

## Preliminary model fitting

linmod <- lm(RentPerM2~.-Location-Room1, rents)
summary(linmod)

## Prior parameters

a.s2 <- 0.001; b.s2 <- 0.001
a.t2 <- 0.001; b.t2 <- 0.001

## Setup, storage, and starting values

n <- nrow(X); m <- nrow(W)
B <- 10000

#load("HW3.RData")
beta.samps <- matrix(NA, nrow = 12, ncol = B)
beta.samps[,1] <- coef(linmod)

s2.samps <- rep(NA, B)
t2.samps <- rep(NA, B)
s2.samps[1] <- 1
t2.samps[1] <- 1

eta.samps <- matrix(NA, nrow = m, ncol = B)


library(MCMCpack)
## Gibbs sampler

for(i in 2:100){

  if(i%%100==0) print(i)

  ## eta_obs | Rest
  V <- solve(t(H) %*% H/s2.samps[i-1] + (D-W)/t2.samps[i-1])
  mu <- V %*% t(H) %*% (y - X %*% beta.samps[,i-1])/s2.samps[i-1]
  eta.samps[,i] <- rmvnorm(1, mean = mu, Sigma = V, method = "svd")
  eta.samps[,i] <- eta.samps[,i] - mean(eta.samps[,i]) # subtracting mean of eta_j

  ## beta | Rest
```

```
  V <- s2.samps[i-1]*solve(t(X) %*% X)
  mu <-  solve(t(X) %*% X) %*% t(X) %*% (y - H %*% eta.samps[,i])
  beta.samps[,i] <- rmvnorm(1, mean = mu, Sigma = V, method = "svd")

  ## s2 | Rest
  a <- a.s2 + n/2
  resid <- y - X %*% beta.samps[,i] - H %*% eta.samps[,i]
  b <- b.s2 + t(resid) %*% resid /2
  s2.samps[i] <- rinvgamma(1, a, b)

  ## t2 | Rest
  a <- a.t2 + (m-1)/2
  b <- b.t2 + t(eta.samps[,i]) %*% (D-W) %*% eta.samps[,i]/2
  t2.samps[i] <- rinvgamma(1, a, b)

}

#save(beta.samps,s2.samps,t2.samps,eta.samps,file="HW3.RData")
load('HW3.RData')



## Diagnostics

# burnin
burnin <- 100
s2.burn <- s2.samps[-(1:burnin)]
t2.burn <- t2.samps[-(1:burnin)]
beta.burn <- beta.samps[,-(1:burnin)]
eta.burn <- eta.samps[,-(1:burnin)]


## s2 trace plot and ACF plot
par(mfrow=c(1,2),mar = c(3,3,3,3) )
plot(s2.burn, type = "l",
     xlab="Iteration Index", ylab="Estimate", main=expression("Trace plot of"~sigma^2))
acf(s2.burn, main=expression("ACF plot of"~sigma^2))
## t2 trace plot and ACF plot
plot(t2.burn, type = "l",
     xlab="Iteration Index", ylab="Estimate", main=expression("Trace plot of"~tau^2))
acf(t2.burn, main=expression("ACF plot of"~tau^2))

## Find posterior means and sds

# beta
beta.mean <- apply(beta.burn,1,mean)
beta.ci <- apply(beta.burn,1,quantile,probs=c(.025,.975))
beta.result <- t(rbind(mean=beta.mean,beta.ci))
rownames(beta.result) <- names(linmod$coef)
round(beta.result,4)


# eta
eta.mean <- apply(eta.burn, 1, mean)
eta.sd <- apply(eta.burn, 1, sd)
```

```
par(mfrow=c(1,1),mar=c(1,0,2,0))
# means
par(mfrow = c(1,1), mar = c(2,2,2,2))
pal <- two.colors(n=5, start="white", end="black", middle="grey", alpha=1.0)
q <- classIntervals(eta.mean, n = 5, style = "quantile")
col <- findColours(q, pal)
plot(districts.sp, col = col)
legend("bottomleft", fill = attr(col, "palette"),
       legend = names(attr(col, "table")),
       bty="n", cex = 0.8, y.intersp = 1)
title(main=expression(bold("Posterior Mean")))

# sds
pal <- two.colors(n=5, start="white", end="black", middle="grey", alpha=1.0)
q <- classIntervals(eta.sd, n = 5, style = "quantile")
col <- findColours(q, pal)
plot(districts.sp, col = col)
legend("bottomleft", fill = attr(col, "palette"),
       legend = names(attr(col, "table")),
       bty="n", cex = 0.8, y.intersp = 1)
title(main=expression(bold("Posterior Standard Deviation")))
```