Homework 3

Stat 597a: Spatial Models

Claire Kelling

Due October 31, 2017

Problem 1:

Fit a linear model relating rent per square meter to the covariates using least squares, and extract the coefficient estimates. You can ignore the Location variable for now since we will later treat this as a random effect. Also note that the room indicator variables include one that is redundant, so treat a single room as the baseline (i.e., leave Room1 out of the model, so the intercept corresponds to a single room and coefficients for the others represent adjustments to the intercept for a different number of rooms).

Problem 2:

There are two SpatialPolygons objects associated with this dataset, districts.sp and parks.sp. The first corresponds to city districts in which apartments may be located. The second corresponds to districts with no possible apartments, such as parks or fields. Create an nb object with neighbors for the districts, defining neighbors as districts that share a common boundary. Make a plot showing the districts, then add the parks shaded a different color. Think about the way parks are treated; what else could we do?

Problem 3:

There are 380 districts in districts.sp, and the corresponding district numbers are indicated by the Location variable in rents. How many of the 380 districts appear in the rent dataset? I've included a matrix H that provides a mapping between the districts as they're ordered in districts.sp and as they appear in the rents dataframe. Use H to create a new vector containing the number of observations in each district, and make a color or grayscale plot to illustrate this. Note that inference for unobserved districts will still be possible under a hierarchical mixed effects model, since we can "borrow strength" from nearby districts that do have observations.

Problem 4

We will now create a Gibbs sampler to sample from the posterior distribution under the following Bayesian model. Let X be the matrix of covariates, including the intercept term. Let n be the number of data points in Y and m be the number of spatial locations in η. Data model:

$$Y|\beta, \eta, \sigma^2 \sim N(X\beta + H\eta, \sigma^2 I)$$

Process model:

$$p(\eta|\tau^2) \propto (\tau^2)^{-(m-1)/2} \exp\{\frac{-1}{2\tau^2}\eta^T(D_w - W)\eta\}$$

where W is the matrix of 0's and 1's indicating the neighborhood structure from problem 2, and D_w is a diagonal matrix with diagonal entries $\sum_j W_{1j}, ..., \sum_j W_{nj}$. That is, η follows an (improper) intrinsic autoregressive model. The $-(m-1)/2$ exponent on τ^2 is due to the fact that the matrix $D_w - ????W$ has rank m-1 rather than m.

Prior model: Specify independent priors for β, σ^2, and τ^2, with

$$p(\beta) \propto 1, \text{ and } \sigma^2, \tau^2 \sim InverseGamma(0:001, 0:001)$$

The full conditional distributions for β, η σ^2, and τ^2 are given at the end of this assignment. Construct a Gibbs sampler that cycles through each of the full conditionals and stores the results for B = 10,000 iterations. The full conditionals are given below.

A few notes to keep in mind when constructing the sampler:

- The matrix W can be computed from your nb object in problem 2; see help(nb2mat). I also included objects X and y with the data file.
- The function rinvgamma is in the library MCMCpack.
- IMPORTANT: The intrinsic autoregressive model is an example of a pair-wise difference prior. It defines proper distributions for the differences $n_i - n_j$, but it also implicitly contains a distribution for $\frac{1}{m}\sum_{i=1}^{m}\eta_i$ that has infinite variance. In practice, since there is also an intercept term in $X\beta$, we impose the constraint $\sum_{i=1}^{m}\eta_i = 0$ when we sample from the full conditional for η. **Do this numerically by subtracting the mean $\frac{1}{m}\sum_{i=1}^{m}\eta_i^{(j)}$ from $\eta^{(j)}$ in each iteration j.**

Full Conditionals:

$$\beta|Rest \sim N((X^TX)^{-1}X^T(Y - H_\eta), \sigma^2(X^TX)^{-1})$$

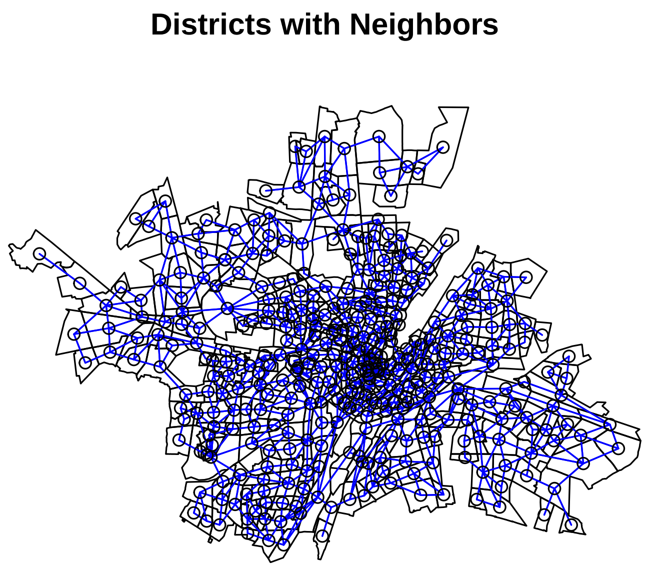
$$\eta|Rest \sim N([H^TH/\sigma^2 + (D_w - W)/\tau^2]^{-1}H^T(Y_X\beta)/\sigma^2, [H^TH/\sigma^2 + (D_w - W)/\tau^2]^{-1})$$

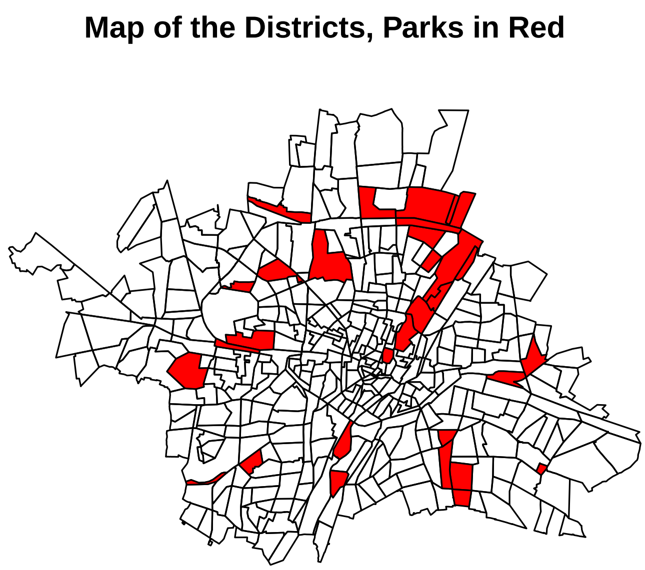
$$\sigma^2|Rest \sim InverseGamma(0.001 + n/2, 0.001 + (Y - X\beta - H\eta)^T(Y - X\beta - H\eta)/2)$$

$$\tau^2|Rest \sim InverseGamma(0.001 + (m-1)/2, 0.001 + \eta^T(D_w - W)\eta/2)$$

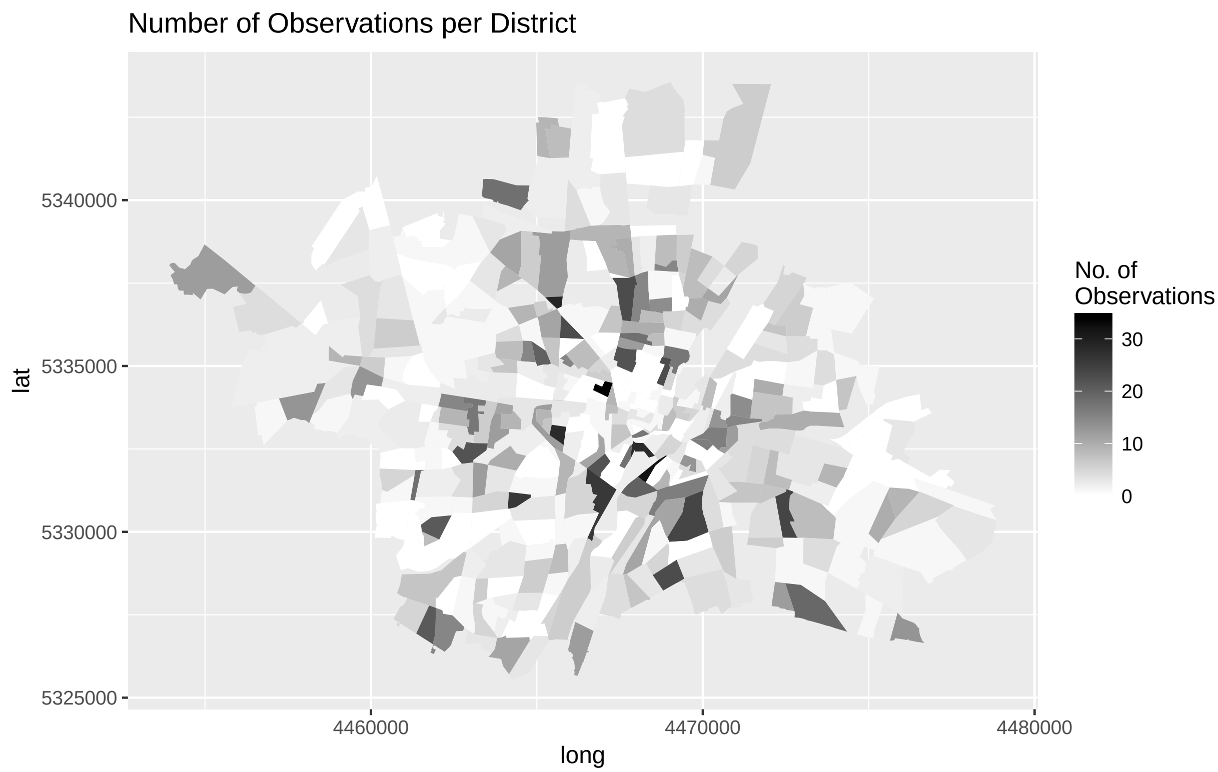
Turn in the following:

- Your map with the neighbors from problem

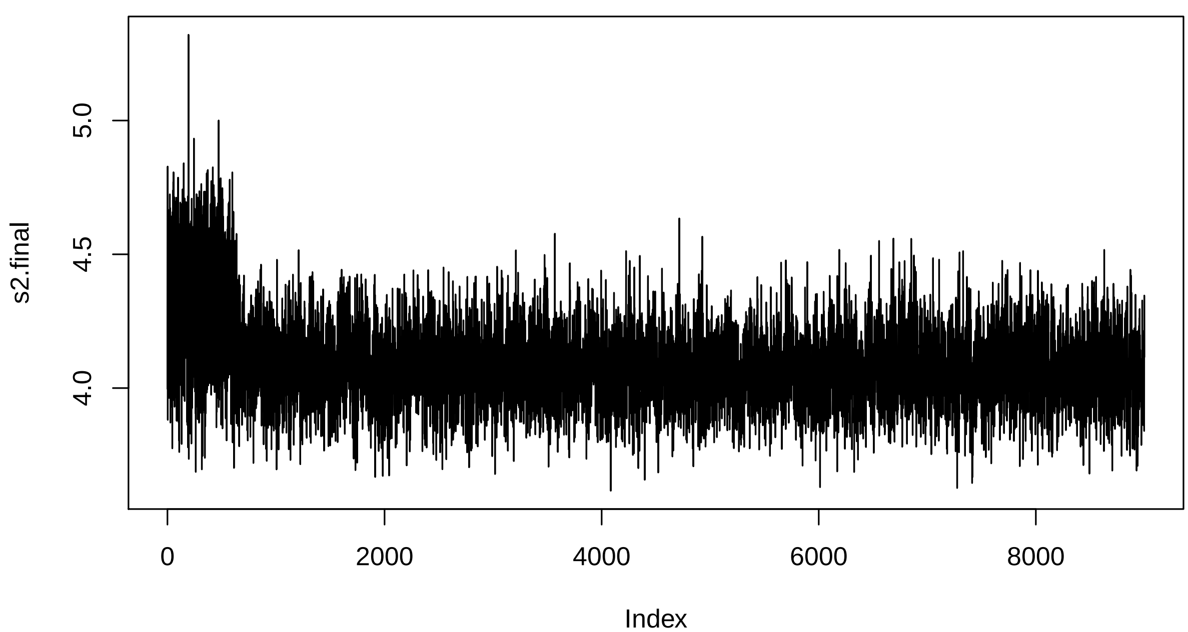


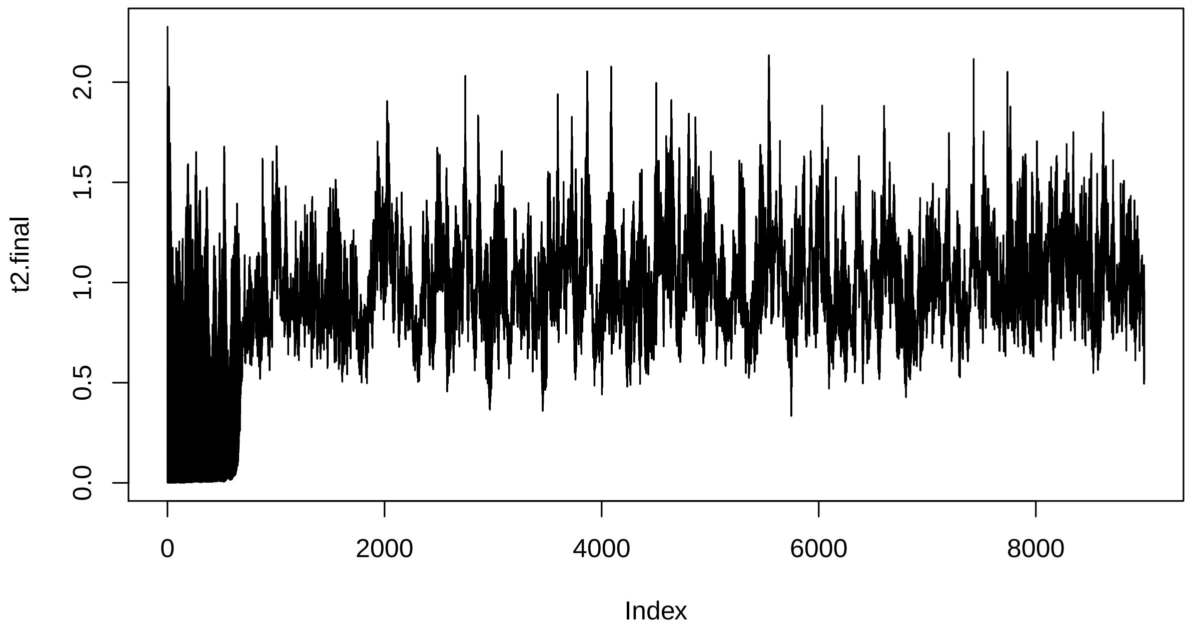


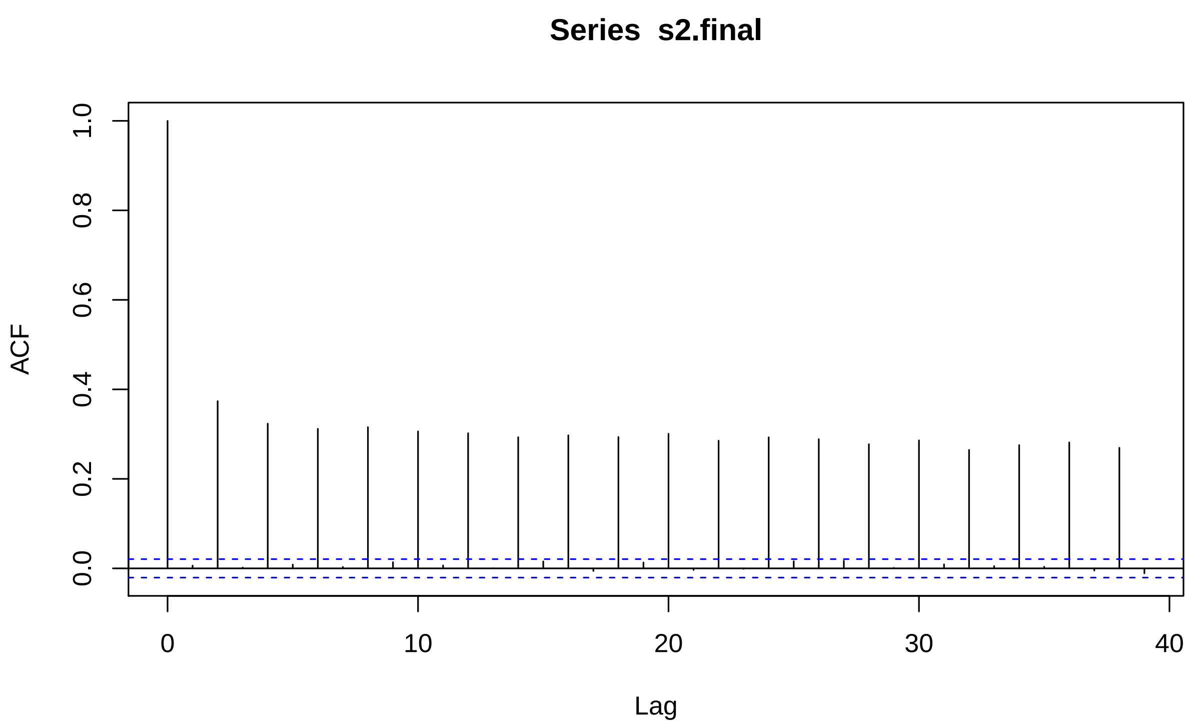
- Your map of the apartment counts for each district

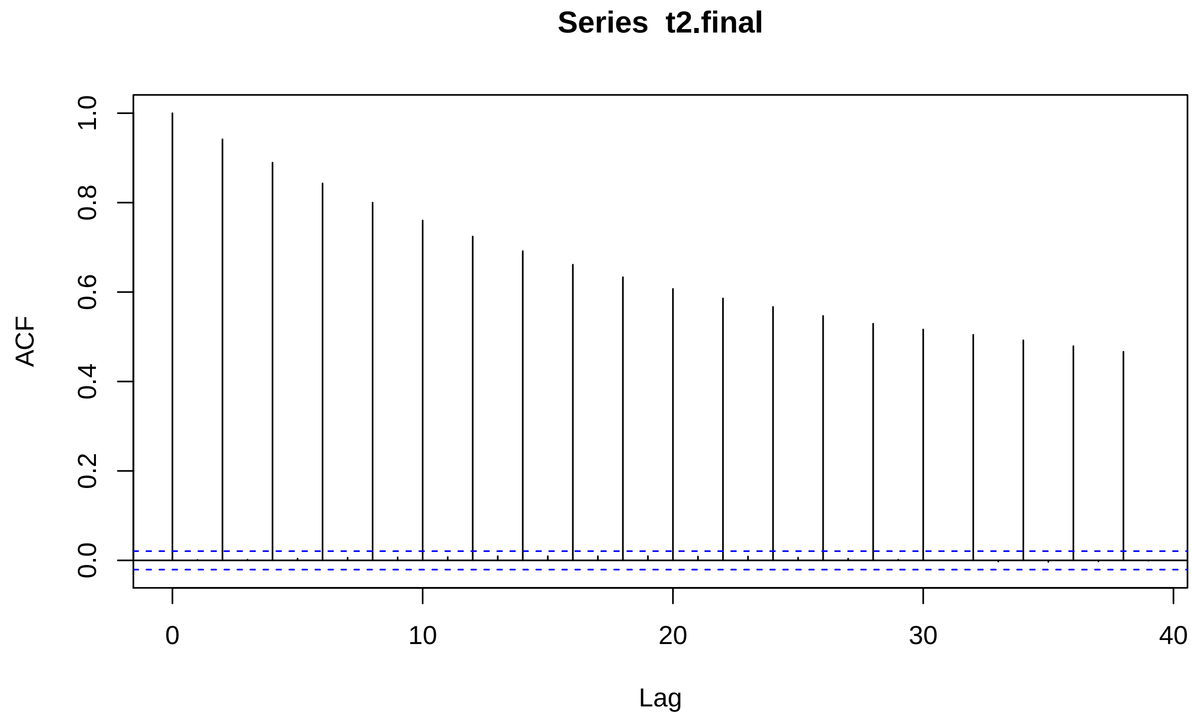


- Trace plots and ACF plots for σ^2 and τ^2




t2.final
0.0
0.5
1.0
1.5
2.0
0
2000
4000
6000
8000
Index

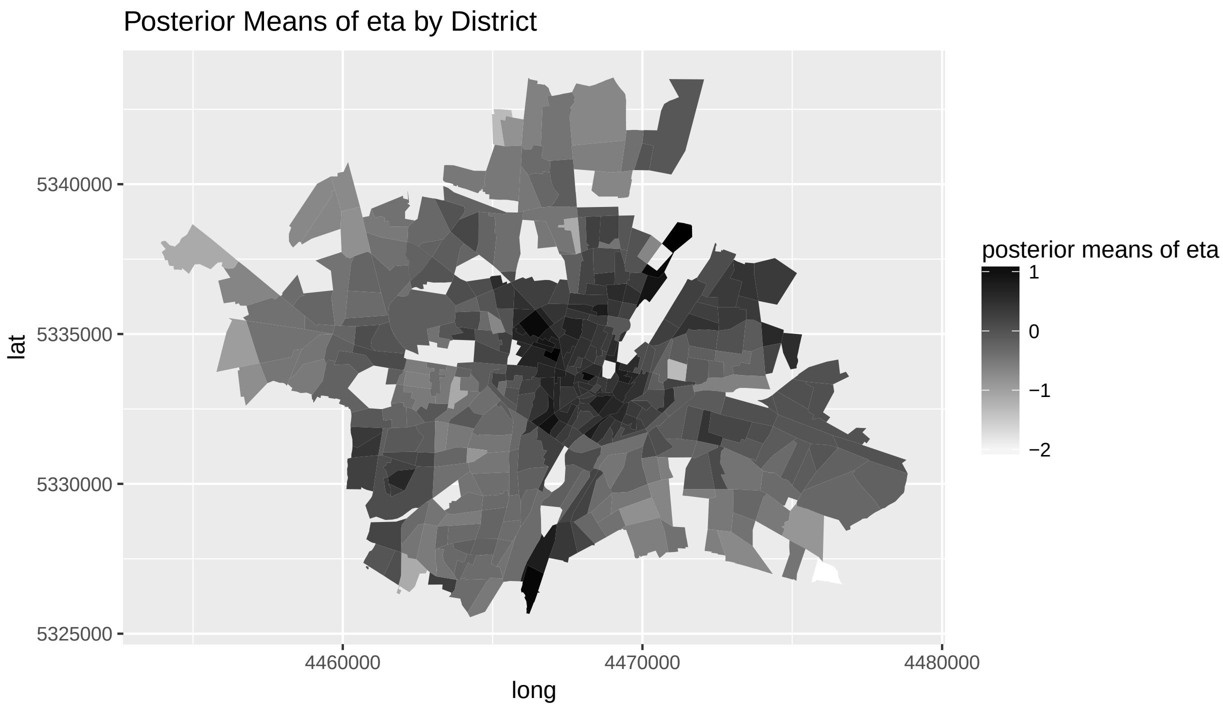

Series s2.final
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
Lag



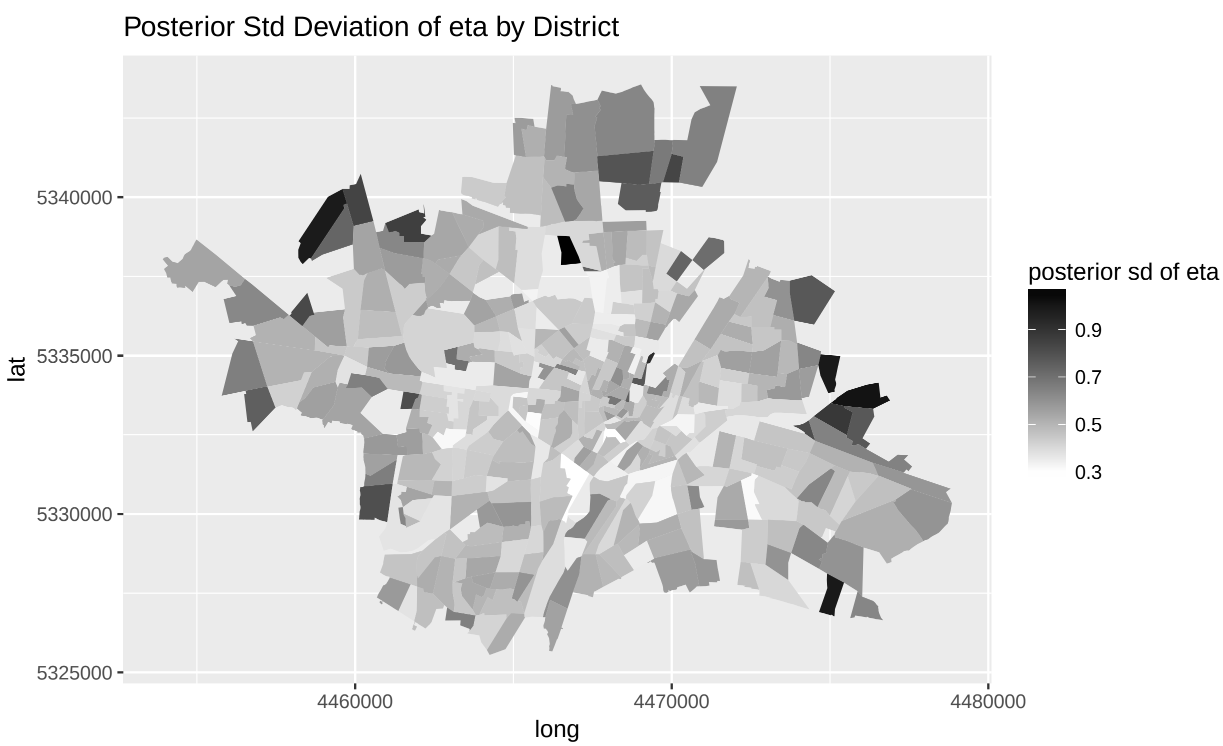
- A table with posterior means of the β's and 95

```
##     0.025 Quantile Posterior Mean 0.975 Quantile
## b1     -34.0853570   -25.26776689   -15.69927628
## b2       0.0131476     0.01800722     0.02248055
## b3      -2.4560496    -1.87554267    -1.30166235
## b4      -1.7007211    -1.31129643    -0.91316700
## b5      -0.8880378    -0.65612526    -0.41734344
## b6       0.3036925     0.62242555     0.93780758
## b7       1.0077771     1.36826462     1.72779897
## b8      -1.5454352    -1.24606456    -0.93916903
## b9      -2.0640910    -1.76308877    -1.46013900
## b10     -2.6599202    -2.28317595    -1.90936506
## b11     -2.8842007    -2.22835413    -1.55267152
## b12     -3.6434326    -2.48887803    -1.36137025
```

- A color or grayscale map of the posterior means for the vector η



- A color or grayscale map of the posterior standard deviations for the vector η




```
#load the data
load("C:/Users/ckell/OneDrive/Penn State/2017-2018/597/spatial_statistics_597/Homework 3/data/munichren
```

```r
head(rents)

#Fit a linear model relating rent per square meter to the covariates using least squares
#  This model is fit without Location (used this later as a random effect)
#  This model is also fit without Room1 because we treat a single room as the baseline
lin_mod <- lm(RentPerM2~ Year+ NoHotWater+NoCentralHeat+ NoBathTiles+
                SpecialBathroom + SpecialKitchen +Room2+Room3 +Room4 +Room5+ Room6, data=rents)
summary(lin_mod)

#extract the coefficient estimates
beta <- lin_mod$coefficients
```

```r
plot(parks.sp)
plot(districts.sp)

#Create an nb object with neighbors for the districts, defining neighbors as districts that
#share a common boundary.
#queen is the option that has neigbor being at least one point shared, including corners
#rook does not include corners, only borders
district_neighb <- poly2nb(districts.sp, queen = FALSE)

#plot the neighborhoods
coords <- coordinates(districts.sp)
plot(districts.sp, main = "Districts with Neighbors")
plot(district_neighb, coords, col="blue", add = TRUE)


#Make a plot showing the districts, then add the parks shaded a different color.
plot(districts.sp, main = "Map of the Districts, Parks in Red")
plot(parks.sp,col = "red", add=TRUE)
```

```r
#How many of the 380 districts appear in the rent dataset?
length(unique(rents$Location))
# There are 312 districts represented in the rents dataset, of the 380 total districts.

# Use H to create a new vector containing the number of observations in each district
# There are 380 districts, so I want a 1x380 or 380x1 vector
dim(H) #H is of dimension 2035X380
# So, I need (1x2035)(2035x380) = 1x380
rents$indicator <- rep(1, nrow(rents))
num_per_dist <- rents$indicator%*% H
dim(num_per_dist) # this is my 1x380 vector
sum(num_per_dist) # This is equal to 2035, the total number of observations (rows) in the rent dataset
num_per_dist <- as.numeric(num_per_dist)

#Now I will create the data structure that I need to create a plot
sp_f <- fortify(districts.sp)
head(sp_f)
districts.sp$id <- row.names(districts.sp)
head(districts.sp@data, n=2)
districts.sp@data$num <- num_per_dist
sp_f <- left_join(sp_f, districts.sp@data)
```

```
#make a color or grayscale plot to illustrate this
obs_by_dist <- ggplot(sp_f, aes(long, lat, group = group, fill = num)) + geom_polygon() + coord_equal()
  labs(fill = "No. of \nObservations") +
  ggtitle("Number of Observations per District")+ scale_fill_gradient(low = "white", high = "black")
```

```
# Construct a Gibbs sampler that cycles through each of the full conditionals and stores
# the results for B = 10,000 iterations.

# I will start by creating the input parameters/vectors/matrices
n <-  nrow(rents) # number of data points in Y
m <- nrow(districts.sp) # number of spatial locations in eta
W <- nb2mat(neighbours = district_neighb, style = "B")
#W is matrix of 0's and 1's indicating neighb structure from 2, need style = B to be binary (W is row s
#X and y are given in the data file
D <- diag(rowSums(W)) # diagonal matrix with row sums of W

#is there a typo in this question? row sum up to m, not n?

#how many iterations
B <- 10000

#Prior parameters
b <- 1
a.s2 <- 0.001; b.s2 <- 0.001
a.t2 <- 0.001; b.t2 <- 0.001
```

```
#setup and storage and starting values
beta.samps <- matrix(NA, nrow = 12, ncol = B)
beta.samps[,1] <- beta
#coefficients from problem 1

s2.samps <- t2.samps <- rep(NA, B)
#eta.obs.samps <- matrix(NA, nrow = n, ncol = B)
s2.samps[1] <- t2.samps[1] <- 1

eta.obs.samps <- matrix(NA, nrow = m, ncol = B)
eta.obs.samps[,1] <- rep(1, m)
y <- as.matrix(y)

## MCMC sampler
for(i in 2:B){
  print(i)

  ## beta_obs | Rest
  mu_b <- solve(t(X)%*%X)%*%t(X)%*%(y-H%*%as.matrix(eta.obs.samps[,i-1]))
  Sig_b <- s2.samps[i-1]*solve(t(X)%*%X)
  beta.samps[,i] <- rmvnorm(1, mean= mu_b, sigma = Sig_b, method = "svd")

  ## eta | Rest
  mu_e <- solve((t(H)%*%H)/s2.samps[i-1] + (D-W)/t2.samps[i-1])%*%t(H)%*%(y-X%*%beta.samps[,i-1])/s2.sa
  Sig_e <- solve(t(H)%*%H/s2.samps[i-1] + (D-W)/t2.samps[i-1])
  eta.pre <- rmvnorm(1, mean = mu_e, sigma = Sig_e, method = "svd")
  #need to adjust for the pairwise difference prior
  eta.obs.samps[,i] <- eta.pre - mean(eta.pre)
```

```
  ## s2 | Rest
  a <- 0.001 + n/2
  resid <- y-X%*%beta.samps[,i-1]
  second_part <- H%*%as.matrix(eta.obs.samps[,i-1])
  b <- 0.001 + t(resid-second_part)%*%(resid-second_part)/2
  s2.samps[i] <- rinvgamma(1, a, b)

  ## t2 | Rest
  a <- 0.001 + (m-1)/2
  b <- 0.001 + t(as.matrix(eta.obs.samps[,i-1]))%*%(D-W)%*%as.matrix(eta.obs.samps[,i-1])/2
  t2.samps[i] <- rinvgamma(1, a, b)

}

save(t2.samps, s2.samps, beta.samps, eta.obs.samps, file =  "C:/Users/ckell/OneDrive/Penn State/2017-20
```

```
load("C:/Users/ckell/OneDrive/Penn State/2017-2018/597/spatial_statistics_597/Homework 3/data/mcmc_data
## Diagnostics

plot(beta.samps[1,], type = "l")
plot(s2.samps, type = "l")
plot(t2.samps, type = "l")
plot(eta.obs.samps[1,], type = "l")


burnin <- 1000
s2.final <- s2.samps[-(1:burnin)]
t2.final <- t2.samps[-(1:burnin)]
beta.final <- beta.samps[,-(1:burnin)]
eta.obs.final <- eta.obs.samps[,-(1:burnin)]

acf(s2.final)
acf(t2.final)
acf(beta.final[1,])
acf(eta.obs.final[1,])

effectiveSize(s2.final)
effectiveSize(t2.final)
effectiveSize(beta.final[1,])
effectiveSize(eta.obs.final[1,])

#table with posterior means and credible intervals
b_post_mean <- apply(beta.final, 1, mean)
b_CI <- apply(beta.final, 1, function(x) quantile(x, probs = c(0.025, 0.975)))
est_and_ci <- cbind(b_CI[1,], b_post_mean, b_CI[2,])

colnames(est_and_ci) <- c("0.025 Quantile", "Posterior Mean", "0.975 Quantile")
rownames(est_and_ci) <- c("b1","b2","b3","b4","b5","b6","b7","b8","b9","b10","b11","b12")

## Find pointwise posterior means and sds
eta.m <- apply(eta.obs.final, 1, mean)
eta.sd <- apply(eta.obs.final, 1, sd)
```

```
districts.sp@data$eta.m <- eta.m
districts.sp@data$eta.sd <- eta.sd
#districts.sp@data <- districts.sp@data[,-c(2)]
sp_f <- left_join(sp_f, districts.sp@data)


mean_map <- ggplot(sp_f, aes(long, lat, group = group, fill = eta.m)) + geom_polygon() + coord_equal() +
  labs(fill = "posterior means of eta") +
  ggtitle("Posterior Means of eta by District")+ scale_fill_gradient(low = "white", high = "black")

sd_map <- ggplot(sp_f, aes(long, lat, group = group, fill = eta.sd)) + geom_polygon() + coord_equal() +
  labs(fill = "posterior sd of eta") +
  ggtitle("Posterior Std Deviation of eta by District")+ scale_fill_gradient(low = "white", high = "bla
```