

Lecture 18

Fitting CAR and SAR Models

Colin Rundel

11/07/2018

Fitting areal models

- Formula Model

$$y(s_i) = X_{i\cdot}\beta + \phi \sum_{j=1}^n D_{jj}^{-1} A_{ij} (y(s_j) - X_{j\cdot}\beta) + \epsilon_i$$
$$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{D}^{-1})$$

- Joint Model

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{X}\beta, (\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^{-1} \sigma^2 \mathbf{D}^{-1} ((\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^{-1})^t\right)$$

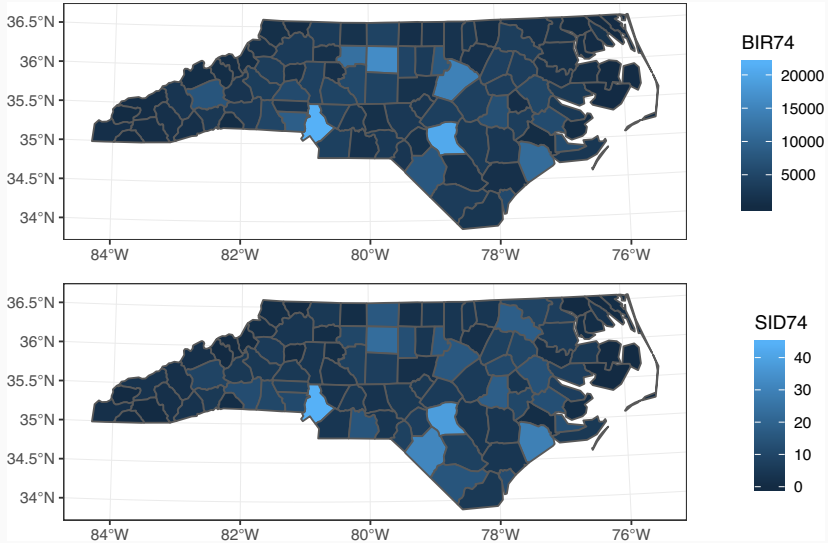
- Conditional Model

$$y(s_i) | \mathbf{y}_{-s_i} \sim \mathcal{N} \left(X_{i \cdot} \beta + \phi \sum_{j=1}^n \frac{A_{ij}}{D_{ii}} (y(s_j) - X_{j \cdot} \beta), \sigma^2 D_{ii}^{-1} \right)$$

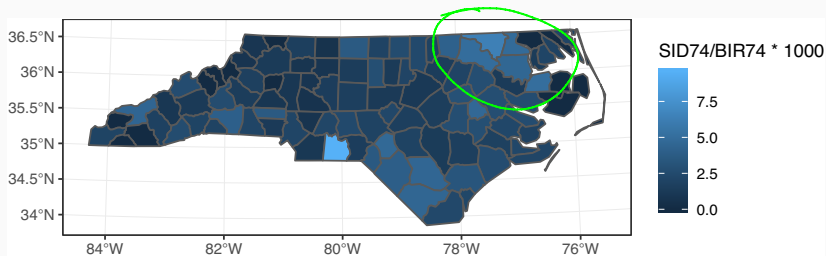
- Joint Model

$$\mathbf{y} \sim \mathcal{N}(\mathbf{X}\beta, \sigma^2(\mathbf{D} - \phi\mathbf{A})^{-1})$$

Example - NC SIDS



```
ggplot() + geom_sf(data=nc, aes(fill=SID74/BIR74*1000))
```



Using `spautolm` from `spdep`

```
library(spdep)

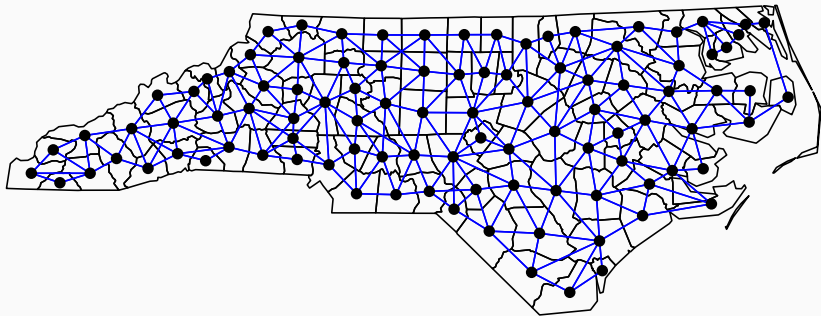
A = st_touches(nc, sparse=FALSE)
listW = mat2listw(A)

listW
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 490
## Percentage nonzero weights: 4.9
## Average number of links: 4.9
##
## Weights style: M
## Weights constants summary:
##      n      nn  S0  S1   S2
## M 100 10000 490 980 10696
```

```
nc_coords = nc %>% st_centroid() %>% st_coordinates()
```

```
plot(st_geometry(nc))
```

```
plot(listW, nc_coords, add=TRUE, col="blue", pch=16)
```



Moran's I

```
spdep::moran.test(nc$SID74, listW)
##
## Moran I test under randomisation
##
## data: nc$SID74
## weights: listW
##
## Moran I statistic standard deviate = 2.1707, p-value = 0.01498
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
## 0.119089049          -0.010101010      0.003542176
spdep::moran.test(1000*nc$SID74/nc$BIR74, listW)
##
## Moran I test under randomisation
##
## data: 1000 * nc$SID74/nc$BIR74
## weights: listW
##
## Moran I statistic standard deviate = 3.6355, p-value = 0.0001387
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
## 0.210046454          -0.010101010      0.003666802
```

```
spdep::geary.test(nc$SID74, listW)
##
## Geary C test under randomisation
##
## data: nc$SID74
## weights: listW
##
## Geary C statistic standard deviate = 0.91949, p-value = 0.1789
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic      Expectation      Variance
##      0.88988684      1.00000000      0.01434105
spdep::geary.test(nc$SID74/nc$BIR74, listW)
##
## Geary C test under randomisation
##
## data: nc$SID74/nc$BIR74
## weights: listW
##
## Geary C statistic standard deviate = 3.0989, p-value = 0.0009711
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic      Expectation      Variance
##      0.67796679      1.00000000      0.01079878
```

CAR Model

```
nc_car = spautolm(formula = SID74/BIR74 ~ 1, data = nc,
                  listw = listW, family = "CAR")

summary(nc_car)
##
## Call: spautolm(formula = SID74/BIR74 ~ 1, data = nc, listw = listW,
##               family = "CAR")
##
## Residuals:
##           Min           1Q       Median           3Q          Max
## -0.00213872 -0.00083535 -0.00022355  0.00055014  0.00768640
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.00200203 0.00024272  8.2484 2.22e-16
##
## Lambda: 0.13062 LR test value: 8.6251 p-value: 0.0033157
## Numerical Hessian standard error of lambda: 0.030472
##
## Log likelihood: 508.3767
## ML residual variance (sigma squared): 2.1205e-06, (sigma: 0.0014562)
## Number of observations: 100
## Number of parameters estimated: 3
## AIC: -1010.8
```

0



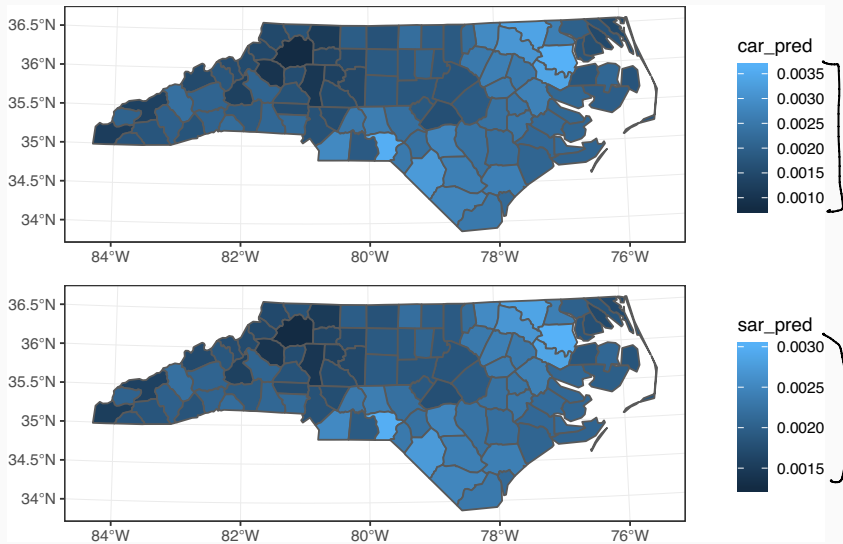
SAR Model

```
nc_sar = spautolm(formula = SID74/BIR74 ~ 1, data = nc,
                  listw = listW, family = "SAR")

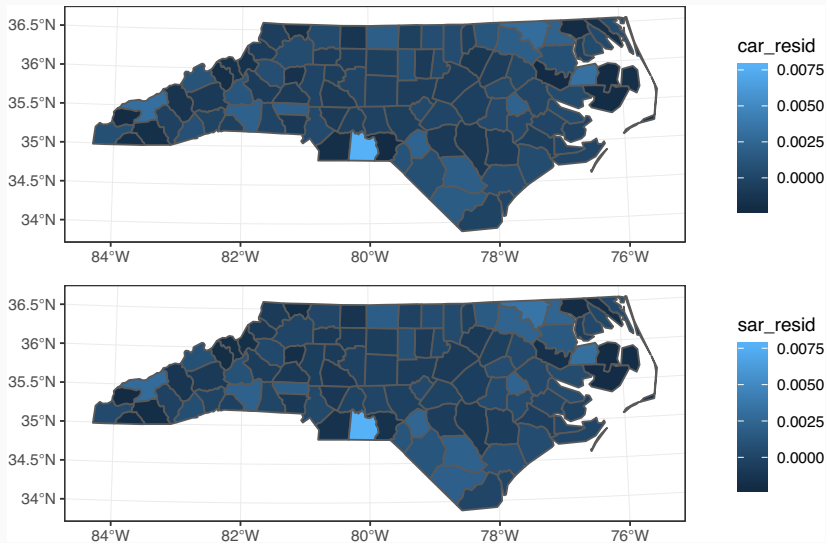
summary(nc_sar)
##
## Call: spautolm(formula = SID74/BIR74 ~ 1, data = nc, listw = listW,
##               family = "SAR")
##
## Residuals:
##           Min             1Q           Median             3Q            Max
## -0.00209307 -0.00087039 -0.00020274  0.00051156  0.00762830
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.00201084 0.00023622  8.5127 < 2.2e-16
##
## Lambda: 0.079934 LR test value: 8.8911 p-value: 0.0028657
## Numerical Hessian standard error of lambda: 0.0246
##
## Log likelihood: 508.5097
## ML residual variance (sigma squared): 2.1622e-06, (sigma: 0.0014704)
## Number of observations: 100
## Number of parameters estimated: 3
## AIC: -1011
```

$\lambda(\phi)$
↓

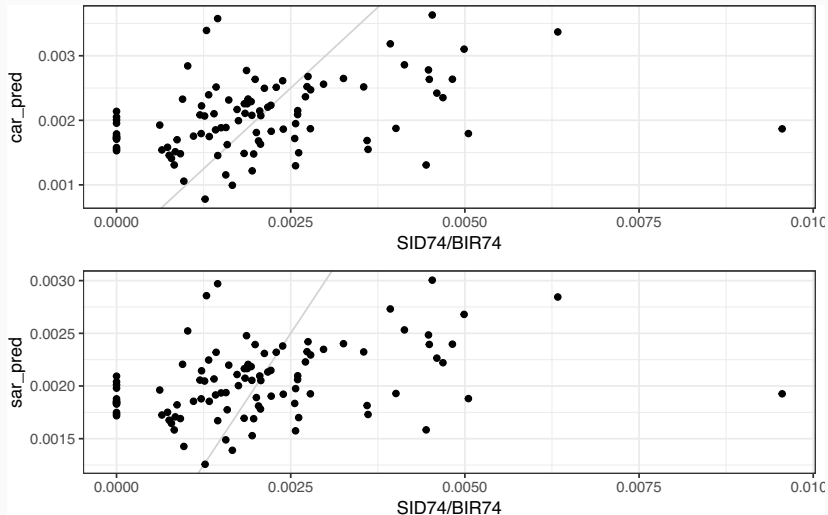
Predictions



Residuals

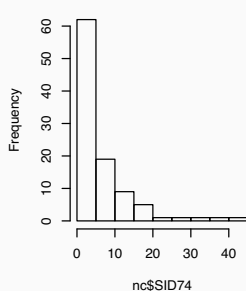


Predicted vs Observed

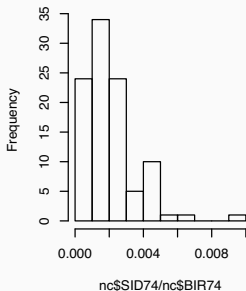


What's wrong?

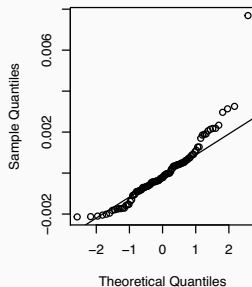
Histogram of nc\$SID74



Histogram of nc\$SID74/nc\$BIR74

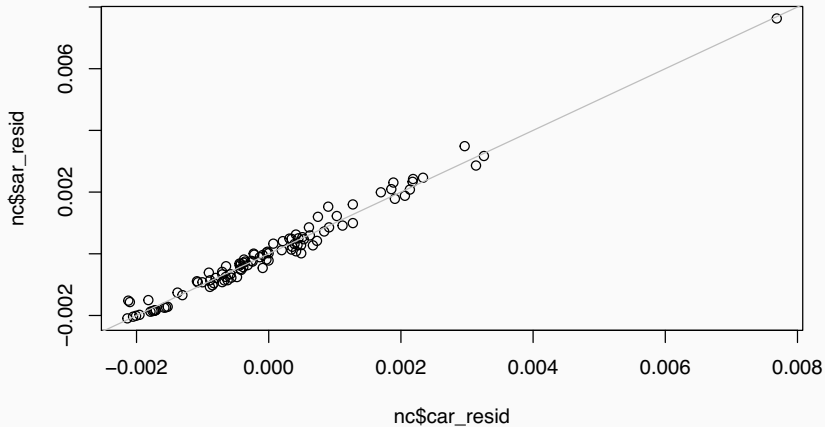


CAR Residuals



Comparing CAR vs SAR.

CAR vs SAR Residuals



Stan CAR Model

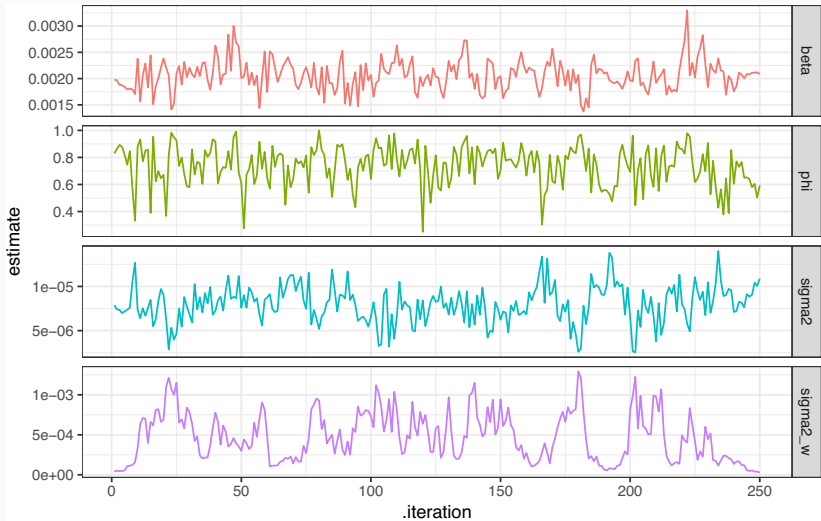
```
car_model = "  
data {  
  int<lower=0> N;  
  vector[N] y;  
  matrix[N,N] A;  
  matrix[N,N] D;  
}  
parameters {  
  vector[N] w_s;  
  real beta;  
  real<lower=0> sigma2;  
  real<lower=0> sigma2_w;  
  real<lower=0,upper=1> phi;  
}  
transformed parameters {  
  vector[N] y_pred = beta + w_s;  
}  
model {  
  ↓ Precision  
  matrix[N,N] Sigma_inv = (D - phi * A) / sigma2;  
  w_s ~ multi_normal_prec(rep_vector(0,N), Sigma_inv);  
  
  beta ~ normal(0,10);  
  sigma2 ~ cauchy(0,5);  
  sigma2_w ~ cauchy(0,5);  
  
  y ~ normal(beta+w_s, sigma2_w);  
}  
"
```

```
data = list(  
  N = nrow(nc),  
  y = nc$SID74 / nc$BIR74,  
  A = A * 1,  
  D = diag(rowSums(A))  
)  
  
car_fit = rstan::stan(  
  model_code = car_model, data = data,  
  iter = 10000, chains = 1, thin=20  
)
```

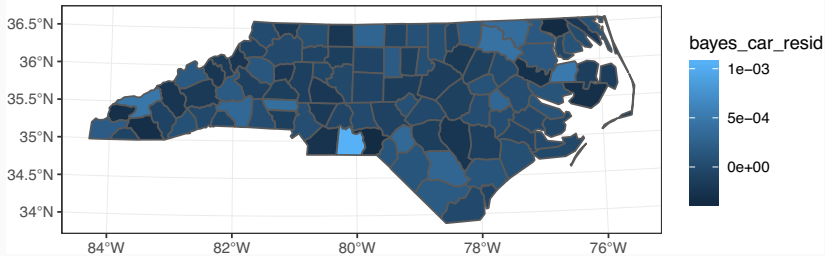
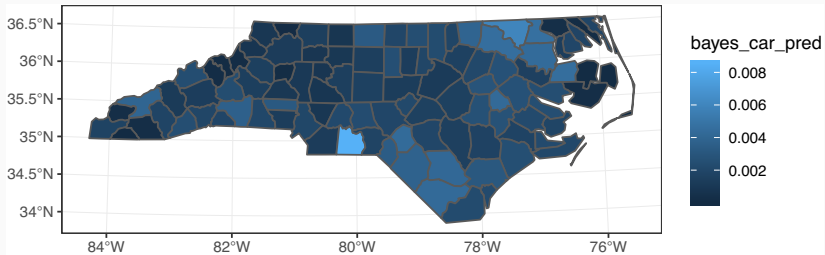
```
data = list(  
  N = nrow(nc),  
  y = nc$SID74 / nc$BIR74,  
  A = A * 1,  
  D = diag(rowSums(A))  
)  
  
car_fit = rstan::stan(  
  model_code = car_model, data = data,  
  iter = 10000, chains = 1, thin=20  
)
```

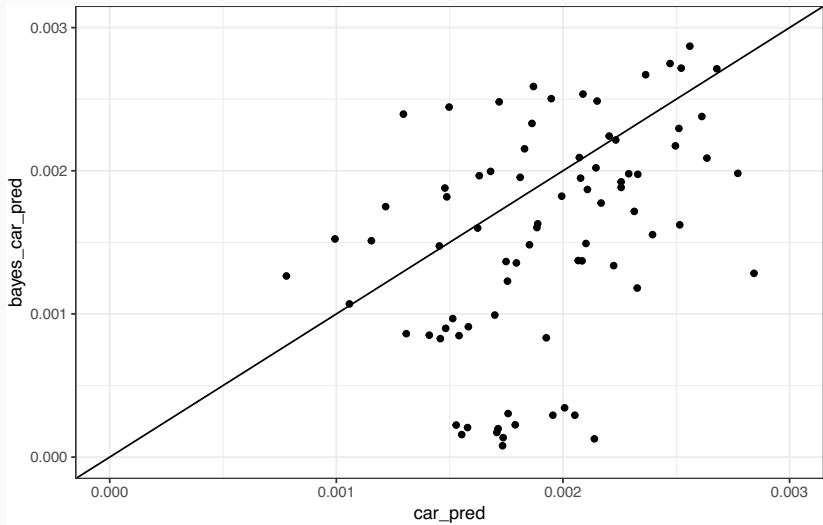
Why don't we use the conditional definition for the y 's?

Model Results



Predictions

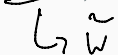




↑
spdep::spautolm

$$\Sigma_{SAR} = (\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^{-1} \sigma^2 \mathbf{D}^{-1} ((\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^{-1})^t$$

$$\begin{aligned} \Sigma_{SAR}^{-1} &= \left((\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^{-1} \sigma^2 \mathbf{D}^{-1} ((\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^{-1})^t \right)^{-1} \\ &= \left(((\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^{-1})^t \right)^{-1} \frac{1}{\sigma^2} \mathbf{D} (\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A}) \\ &= \frac{1}{\sigma^2} (\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A})^t \mathbf{D} (\mathbf{I} - \phi \mathbf{D}^{-1} \mathbf{A}) \end{aligned}$$

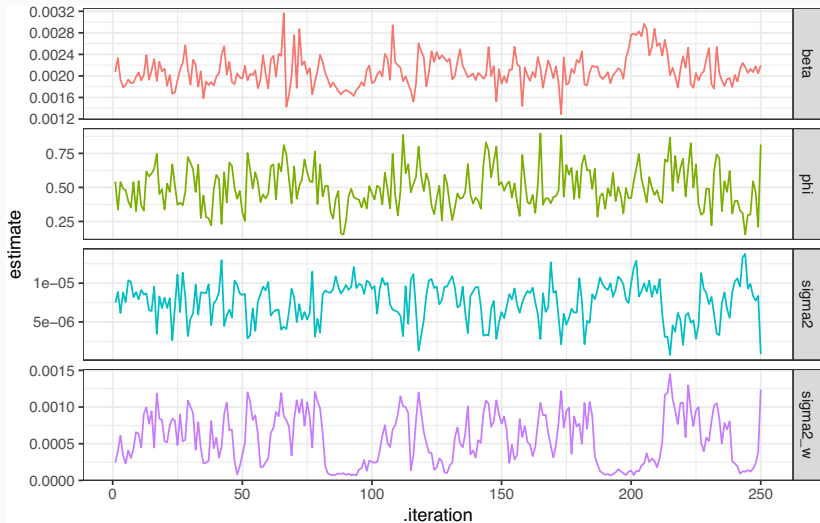


Jags SAR Model

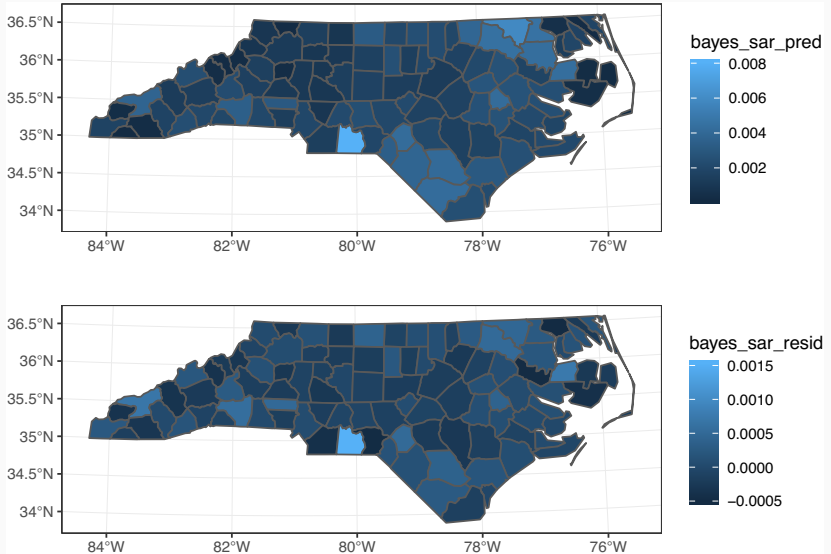
```
sar_model = "  
data {  
  int<lower=0> N;  
  vector[N] y;  
  matrix[N,N] W_tilde;  
  matrix[N,N] D;  
}  
transformed data {  
  matrix[N,N] I = diag_matrix(rep_vector(1, N));  
}  
parameters {  
  vector[N] w_s;  
  real beta;  
  real<lower=0> sigma2;  
  real<lower=0> sigma2_w;  
  real<lower=0,upper=1> phi;  
}  
transformed parameters {  
  vector[N] y_pred = beta + w_s;  
}  
model {  
  matrix[N,N] C = I - phi * W_tilde;  
  matrix[N,N] Sigma_inv = C' * D * C / sigma2;  
  
  w_s ~ multi_normal_prec(rep_vector(0,N), Sigma_inv);  
  
  beta ~ normal(0,10);  
  sigma2 ~ cauchy(0,5);  
  sigma2_w ~ cauchy(0,5);  
  
  y ~ normal(beta + w_s, sigma2_w);  
}  
"
```

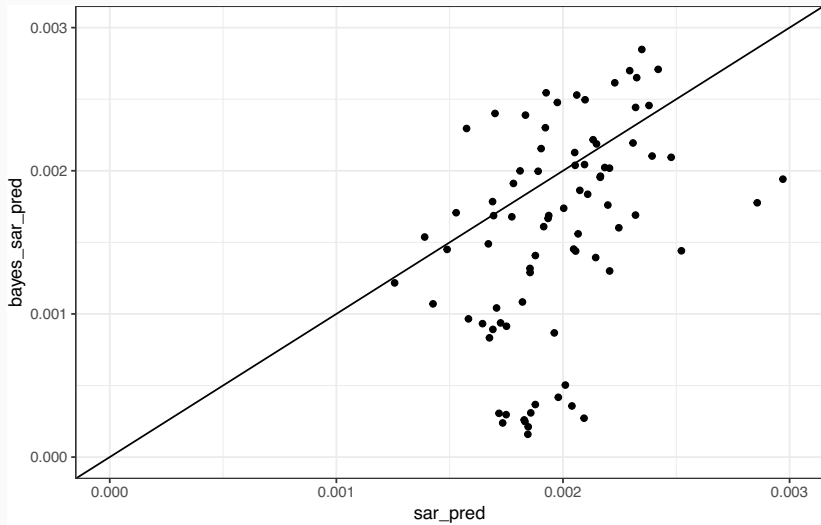
```
D = diag(rowSums(A))  
D_inv = diag(1/diag(D))  
data = list(  
  N = nrow(nc),  
  y = nc$SID74 / nc$BIR74,  
  x = rep(1, nrow(nc)),  
  D_inv = D_inv,  
  W_tilde = D_inv %*% A  
)  
  
sar_fit = rstan::stan(  
  model_code = sar_model, data = data,  
  iter = 10000, chains = 1, thin=20  
)
```

Model Results



Predictions





Comparing Predictions

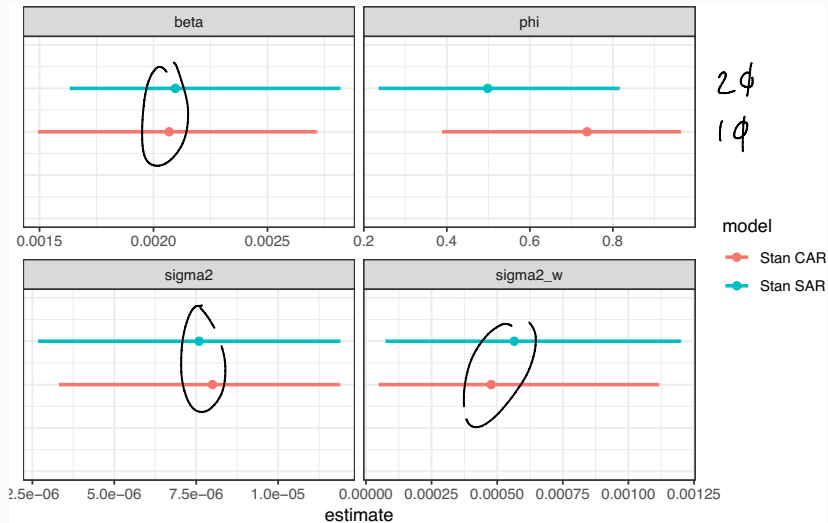
```
# RMSE
sqrt(mean(nc$bayes_car_resid^2))
## [1] 0.0002092447

sqrt(mean(nc$bayes_sar_resid^2))
## [1] 0.0002983034

sqrt(mean(nc$car_resid^2))
## [1] 0.001448564

sqrt(mean(nc$sar_resid^2))
## [1] 0.001470432
```

Comparing Parameters



CAR vs SAR?

- SAR is a bit smoother
- Diffe. interpretations of ϕ
- For diff W or A, D

$$SAR \leftrightarrow \underbrace{CAR}_{\Sigma} \leftrightarrow EP$$

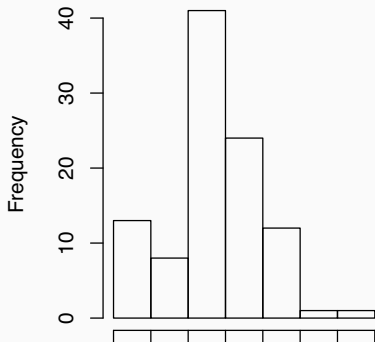
Transforming the data

Freeman-Tukey's transformation

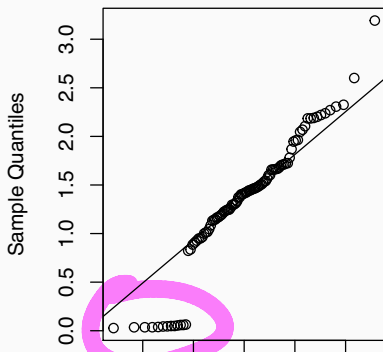
This is the transformation used by Cressie and Road in Spatial Data Analysis of Regional Counts (1989).

$$FT = \sqrt{1000} \left(\sqrt{\frac{SID74}{BIR74}} + \sqrt{\frac{SID74 + 1}{BIR74}} \right)$$

Histogram of nc\$FT



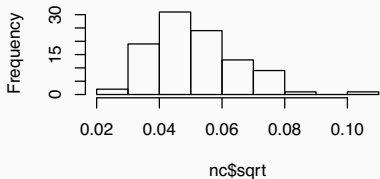
Normal Q-Q Plot



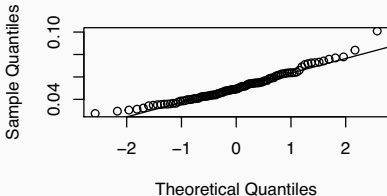
Other options

```
nc = mutate(nc,  
  sqrt = sqrt((SID74+1)/BIR74),  
  log = log((SID74+1)/BIR74),  
)
```

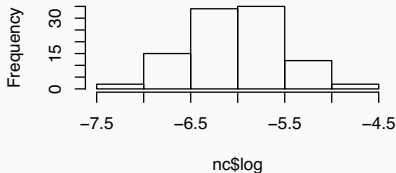
Histogram of nc\$sqrt



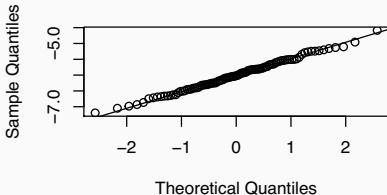
Normal Q-Q Plot



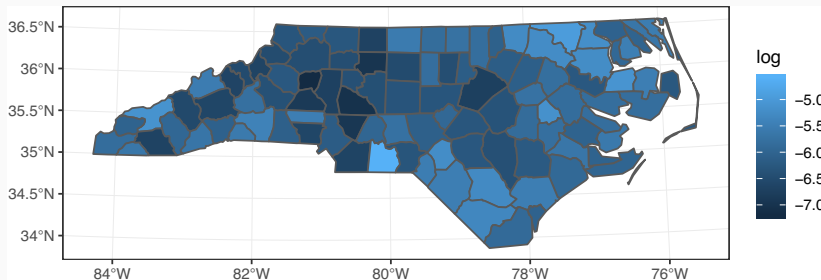
Histogram of nc\$log



Normal Q-Q Plot



log transformation



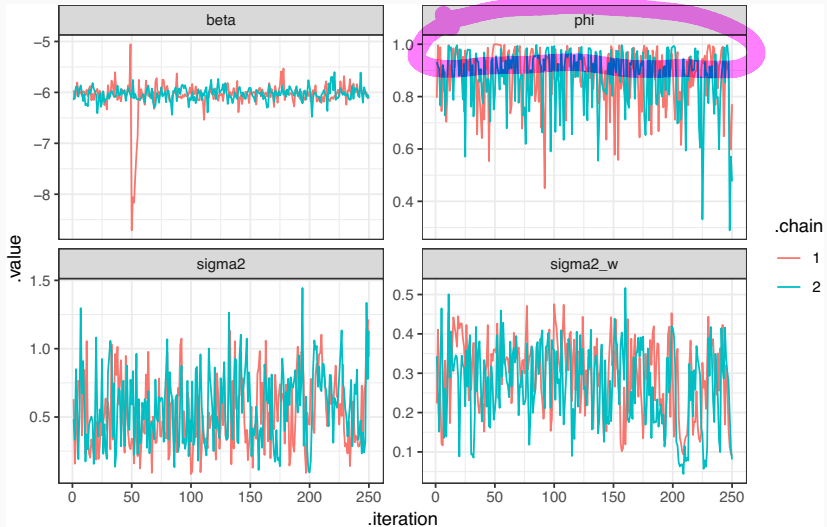
```
##  
## Moran I test under randomisation  
##  
## data: nc$log  
## weights: listW  
##  
## Moran I statistic standard deviate = 4.9895, p-value = 3.027e-07  
## alternative hypothesis: greater  
## sample estimates:  
## Moran I statistic      Expectation      Variance  
## 0.299245438          -0.010101010          0.003843927
```

log CAR Model

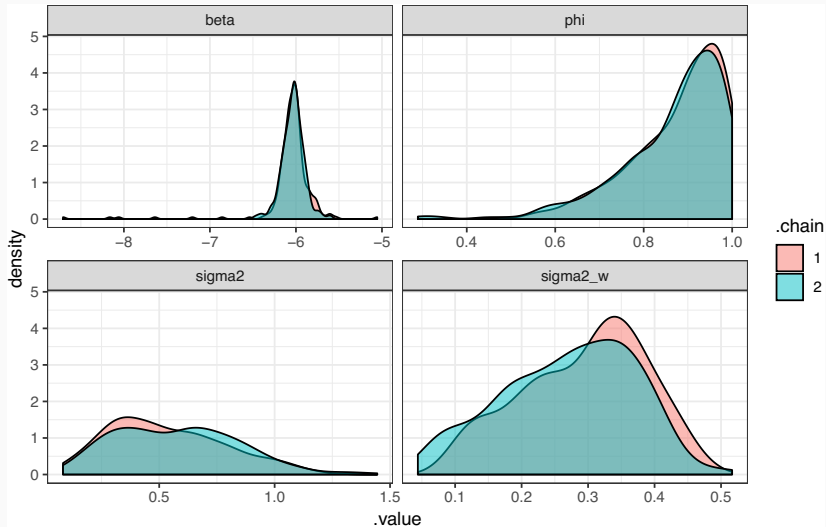
```
car_model = "  
data {  
  int<lower=0> N;  
  vector[N] y;  
  matrix[N,N] A;  
  matrix[N,N] D;  
}  
parameters {  
  vector[N] w_s;  
  real beta;  
  real<lower=0> sigma2;  
  real<lower=0> sigma2_w;  
  real<lower=0,upper=1> phi;  
}  
transformed parameters {  
  vector[N] y_pred = beta + w_s;  
}  
model {  
  matrix[N,N] Sigma_inv = (D - phi * A) / sigma2;  
  w_s ~ multi_normal_prec(rep_vector(0,N), Sigma_inv);  
  
  beta ~ normal(0,10);  
  sigma2 ~ cauchy(0,5);  
  sigma2_w ~ cauchy(0,5);  
  
  y ~ normal(beta+w_s, sigma2_w);  
}  
"
```

```
data = list(  
  N = nrow(nc),  
  y = nc$log,  
  x = rep(1, nrow(nc)),  
  A = A * 1,  
  D = diag(rowSums(A))  
)  
  
car_log_fit = rstan::stan(  
  model_code = car_model, data = data,  
  iter = 10000, thin=20, chains = 2, cores = 2  
)
```

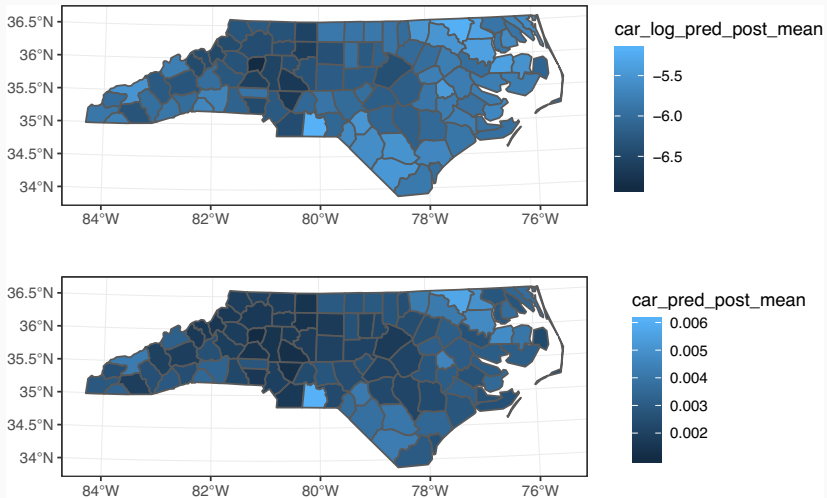
Chains



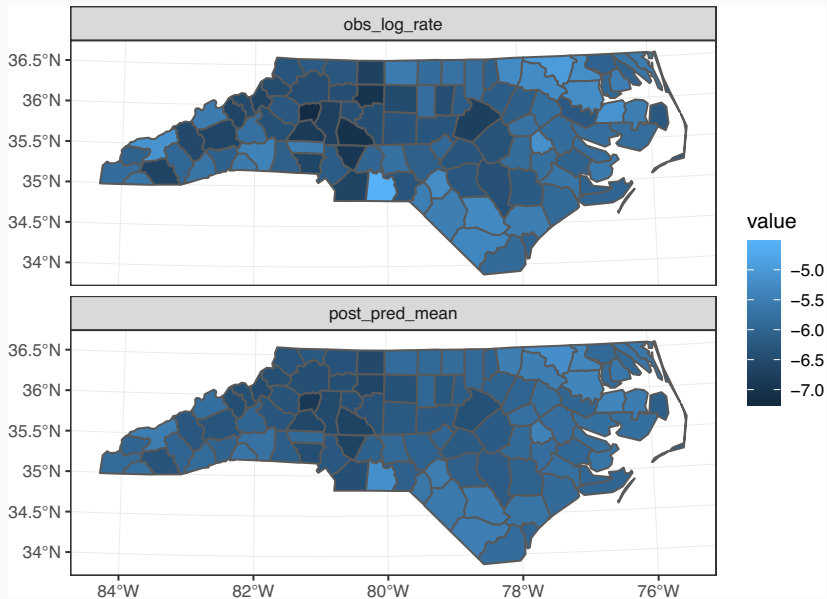
Posteriors



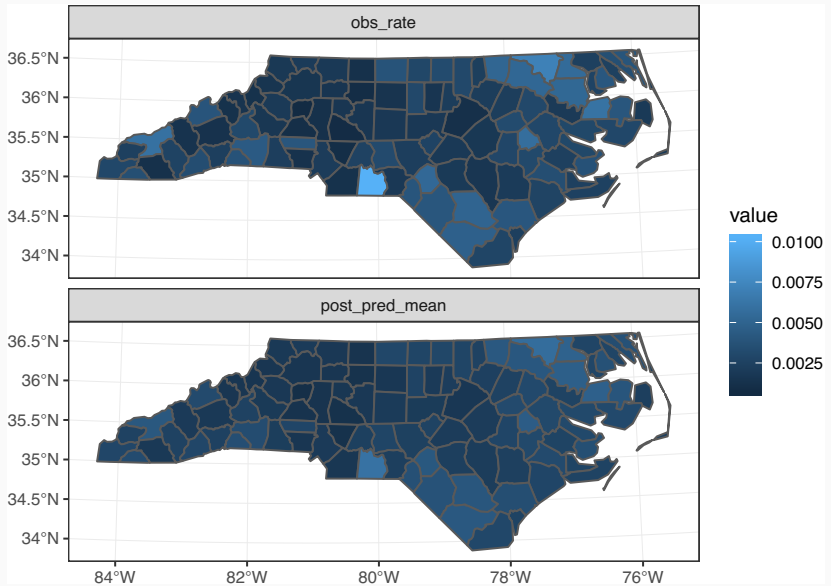
Predictions



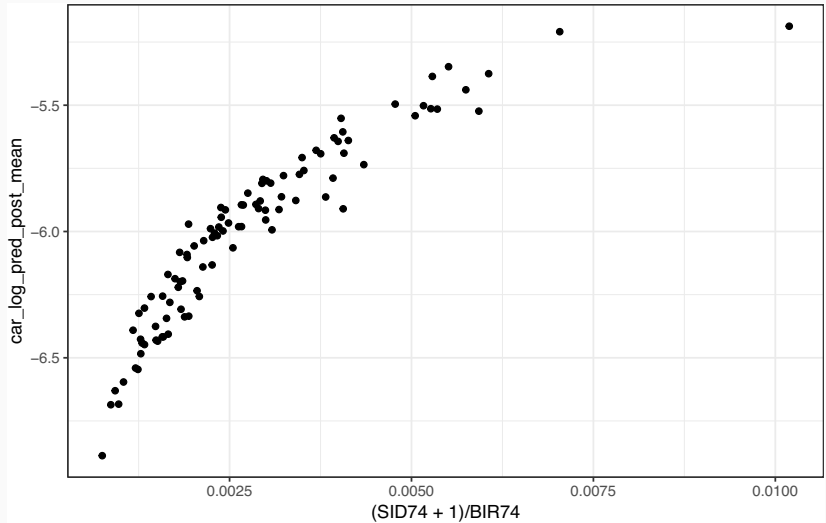
Predicted vs Observed



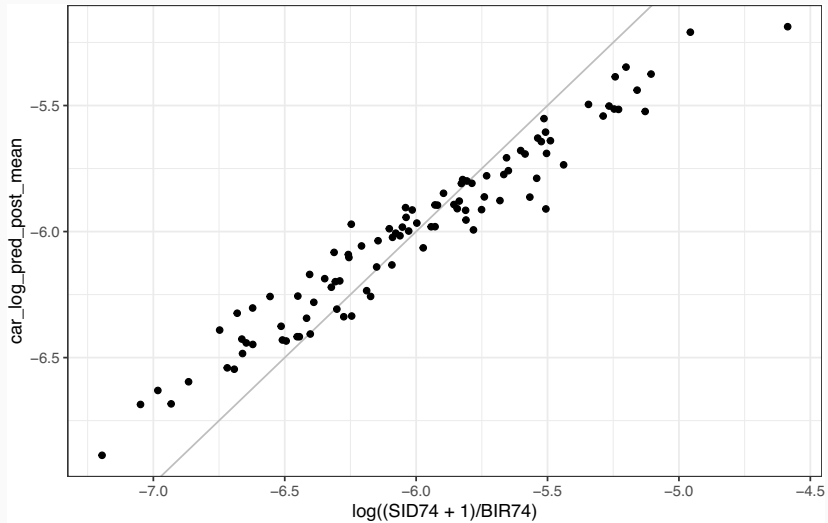
Predicted vs Observed (cont)



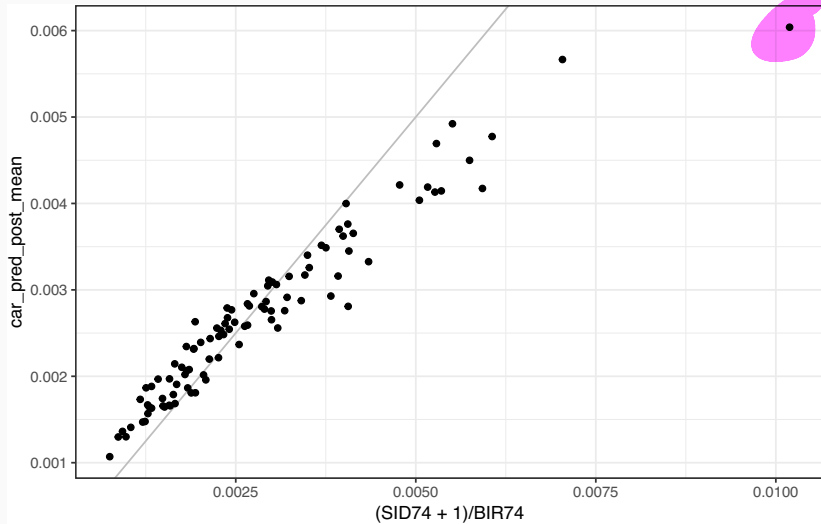
Model Fit



Model Fit (cont.)

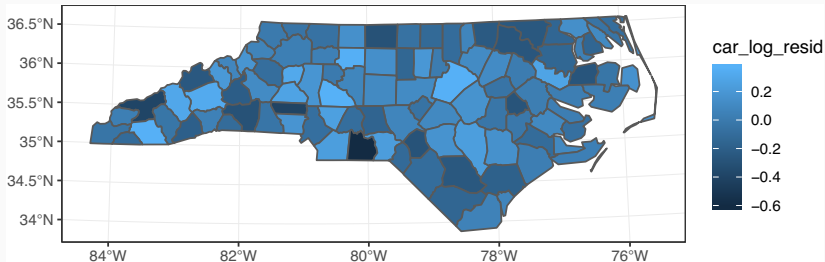


Model Fit (cont.)



Residuals

```
nc = mutate(nc, car_log_resid = car_log_pred_post_mean - log((SID74+1)/BIR74))  
ggplot(nc, aes(fill=car_log_resid)) + geom_sf()
```



```
spdep::moran.test(nc$car_log_resid, listW)  
##  
## Moran I test under randomisation  
##  
## data: nc$car_log_resid  
## weights: listW  
##  
## Moran I statistic standard deviate = -1.1361, p-value = 0.872  
## alternative hypothesis: greater  
## sample estimates:  
## Moran I statistic            Expectation            Variance
```