Lecture 18

Fitting CAR and SAR Models

Colin Rundel

3/29/2018

Fitting areal models

CAR vs SAR

- Simultaneous Autogressve (SAR)

$$y(s_i) = \phi \sum_{j=1}^{n} W_{ij}\, y(s_j) + \epsilon$$

$$\mathbf{y} \sim \mathcal{N}(0,\ \sigma^2\, ((\mathbf{I} - \phi \mathbf{W})^{-1})((\mathbf{I} - \phi \mathbf{W})^{-1})^t)$$

- Conditional Autoregressive (CAR)

$$y(s_i)|\mathbf{y}_{-s_i} \sim \mathcal{N}\left(\phi \sum_{j=1}^{n} W_{ij}\, y(s_j),\ \sigma^2\right)$$

$$\mathbf{y} \sim \mathcal{N}(0,\ \sigma^2\, (\mathbf{I} - \phi \mathbf{W})^{-1})$$

Generally speaking we will want to work with a scaled / normalized version of the weight matrix,

$$\frac{W_{ij}}{W_{i\cdot}}$$

When W is an adjacency matrix we can express this as

$$\mathbf{D}^{-1}\mathbf{W}$$

where $\mathbf{D}^{-1} = \text{diag}(1/|N(s_i)|)$.

We can also allow σ^2 to vary between locations, we can define this as $\mathbf{D}_\tau = \text{diag}(1/\sigma_i^2)$ and most often we use

$$\mathbf{D}_\sigma^{-1} = \text{diag}\left(\frac{\sigma^2}{|N(s_i)|}\right) = \sigma^2\mathbf{D}^{-1}.$$

Revised SAR Model

- Formula Model

$$y(s_i) = X_{i\cdot}\beta + \phi \sum_{j=1}^{n} D_{jj}^{-1} W_{ij} \left(y(s_j) - X_{j\cdot}\beta\right) + \epsilon_i$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{D}^{-1})$$

- Joint Model

$$\underline{Y} = \underline{X}\underline{\beta} + \phi\, \underline{D}^{-1} \underline{W} (\underline{Y} - \underline{X}\underline{\beta}) + \underline{\epsilon}$$

$$(\underline{Y} - \underline{X}\underline{\beta}) = \phi\, \underline{D}^{-1} \underline{W} (\underline{Y} - \underline{X}\underline{\beta}) + \underline{\epsilon}$$

$$(\underline{Y} - \underline{X}\underline{\beta}) = (\underline{I} - \phi\, \underline{D}^{-1} \underline{W})^{-1} \underline{\epsilon}$$

$$\underline{Y} = \underline{X}\underline{\beta} + (\underline{I} - \phi\, \underline{D}^{-1} \underline{W})^{-1} \underline{\epsilon}$$

$$E(\underline{Y}) = \underline{X}\,\underline{\beta}$$

$$\mathrm{Var}(\underline{Y}) = (I - \phi D^{-1} W)^{-1} \sigma^2 D^{-1} (I - \phi D^{-1} W)^{-1^t}$$

Revised SAR Model

- Formula Model

$$y(s_i) = X_{i\cdot}\beta + \phi \sum_{j=1}^{n} D_{jj}^{-1}\, W_{ij}\, \big(y(s_j) - X_{j\cdot}\beta\big) + \epsilon_i$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\, \sigma^2 \mathbf{D}^{-1})$$

- Joint Model

$$\begin{aligned}
\mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \phi \mathbf{D}^{-1}\mathbf{W}\, (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\epsilon} \\
(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) &= \phi \mathbf{D}^{-1}\mathbf{W}\, (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\epsilon} \\
(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})(\mathbf{I} - \phi \mathbf{D}^{-1}\mathbf{W})^{-1} &= \boldsymbol{\epsilon} \\
\mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + (\mathbf{I} - \phi \mathbf{D}^{-1}\mathbf{W})^{-1}\boldsymbol{\epsilon}
\end{aligned}$$

Revised SAR Model

- Formula Model

$$y(s_i) = X_{i\cdot}\beta + \phi \sum_{j=1}^{n} D_{jj}^{-1}\, W_{ij}\, \big(y(s_j) - X_{j\cdot}\beta\big) + \epsilon_i$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\, \sigma^2 \mathbf{D}^{-1})$$

- Joint Model

$$\begin{aligned}
\mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \phi \mathbf{D}^{-1}\mathbf{W}\,\big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big) + \boldsymbol{\epsilon} \\
\big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big) &= \phi \mathbf{D}^{-1}\mathbf{W}\,\big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big) + \boldsymbol{\epsilon} \\
\big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big)(\mathbf{I} - \phi \mathbf{D}^{-1}\mathbf{W})^{-1} &= \boldsymbol{\epsilon} \\
\mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + (\mathbf{I} - \phi \mathbf{D}^{-1}\mathbf{W})^{-1}\boldsymbol{\epsilon}
\end{aligned}$$

$$\mathbf{y} \sim \mathcal{N}\Big(\mathbf{X}\boldsymbol{\beta}, (\mathbf{I} - \phi \mathbf{D}^{-1}\mathbf{W})^{-1}\sigma^2 \mathbf{D}^{-1}\big((\mathbf{I} - \phi \mathbf{D}^{-1}\mathbf{W})^{-1}\big)^{t}\Big)$$

Revised CAR Model

- Conditional Model

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

- Joint Model

Revised CAR Model

- Conditional Model

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

- Joint Model

$$\mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta},\, \Sigma_{CAR})$$

Revised CAR Model

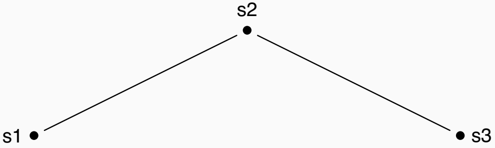
- Conditional Model

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

- Joint Model

$$\mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta},\, \Sigma_{CAR})$$

$$\begin{aligned}
\Sigma_{CAR} &= (\mathbf{D}_{\sigma}\,(\mathbf{I} - \phi\mathbf{D}^{-1}\mathbf{W}))^{-1} \\
&= (1/\sigma^2\mathbf{D}\,(\mathbf{I} - \phi\mathbf{D}^{-1}\mathbf{W}))^{-1} \\
&= (1/\sigma^2(\mathbf{D} - \phi\mathbf{W}))^{-1} \\
&= \sigma^2(\mathbf{D} - \phi\mathbf{W})^{-1}
\end{aligned}$$

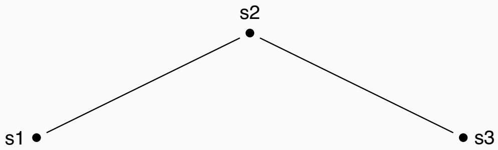


$$W = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad D_{ii} = N(s_i)$$

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad D^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma^{-1} = 1/\sigma^2 \left(D - \phi W \right) = \frac{1}{\sigma^2}\left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & \phi & 0 \\ \phi & 0 & \phi \\ 0 & \phi & 0 \end{pmatrix} \right)$$

Toy CAR Example



$$\mathbf{W} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \mathbf{D} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

. . .

$$\boldsymbol{\Sigma} = \sigma^2 \left(\mathbf{D} - \phi\, \mathbf{W}\right)^{-1} = \sigma^2 \begin{pmatrix} 1 & -\phi & 0 \\ -\phi & 2 & -\phi \\ 0 & -\phi & 1 \end{pmatrix}^{-1}$$

When does Σ exist?

```r
check_sigma = function(phi) {
  Sigma_inv = matrix(c(1,-phi,0,-phi,2,-phi,0,-phi,1), ncol=3, byrow=TRUE)
  solve(Sigma_inv)
}

check_sigma(phi=0)
##      [,1] [,2] [,3]
## [1,]    1  0.0    0
## [2,]    0  0.5    0
## [3,]    0  0.0    1

check_sigma(phi=0.5)
##           [,1]      [,2]      [,3]
## [1,] 1.1666667 0.3333333 0.1666667
## [2,] 0.3333333 0.6666667 0.3333333
## [3,] 0.1666667 0.3333333 1.1666667

check_sigma(phi=-0.6)
##          [,1]     [,2]     [,3]
## [1,]  1.28125 -0.46875  0.28125
## [2,] -0.46875  0.78125 -0.46875
## [3,]  0.28125 -0.46875  1.28125
```

```r
check_sigma(phi=1)
## Error in solve.default(Sigma_inv): Lapack routine dgesv: system is exactl

check_sigma(phi=-1)
## Error in solve.default(Sigma_inv): Lapack routine dgesv: system is exactl

check_sigma(phi=1.2)
##            [,1]      [,2]       [,3]
## [1,] -0.6363636 -1.363636 -1.6363636
## [2,] -1.3636364 -1.136364 -1.3636364
## [3,] -1.6363636 -1.363636 -0.6363636

check_sigma(phi=-1.2)
##            [,1]      [,2]       [,3]
## [1,] -0.6363636  1.363636 -1.6363636
## [2,]  1.3636364 -1.136364  1.3636364
## [3,] -1.6363636  1.363636 -0.6363636
```

When is Σ positive semidefinite?

```
check_sigma_pd = function(phi) {
  Sigma_inv = matrix(c(1,-phi,0,-phi,2,-phi,0,-phi,1), ncol=3, byrow=TRUE)
  chol(solve(Sigma_inv))
}

check_sigma_pd(phi=0)
##      [,1]      [,2] [,3]
## [1,]    1 0.0000000    0
## [2,]    0 0.7071068    0
## [3,]    0 0.0000000    1

check_sigma_pd(phi=0.5)
##          [,1]      [,2]      [,3]
## [1,] 1.080123 0.3086067 0.1543033
## [2,] 0.000000 0.7559289 0.3779645
## [3,] 0.000000 0.0000000 1.0000000

check_sigma_pd(phi=-0.6)
##          [,1]       [,2]       [,3]
## [1,] 1.131923 -0.4141182  0.2484709
## [2,] 0.000000  0.7808688 -0.4685213
## [3,] 0.000000  0.0000000  1.0000000
```

```r
check_sigma_pd(phi=1)
## Error in solve.default(Sigma_inv): Lapack routine dgesv: system is exactl

check_sigma_pd(phi=-1)
## Error in solve.default(Sigma_inv): Lapack routine dgesv: system is exactl

check_sigma_pd(phi=1.2)
## Error in chol.default(solve(Sigma_inv)): the leading minor of order 1 is

check_sigma_pd(phi=-1.2)
## Error in chol.default(solve(Sigma_inv)): the leading minor of order 1 is
```

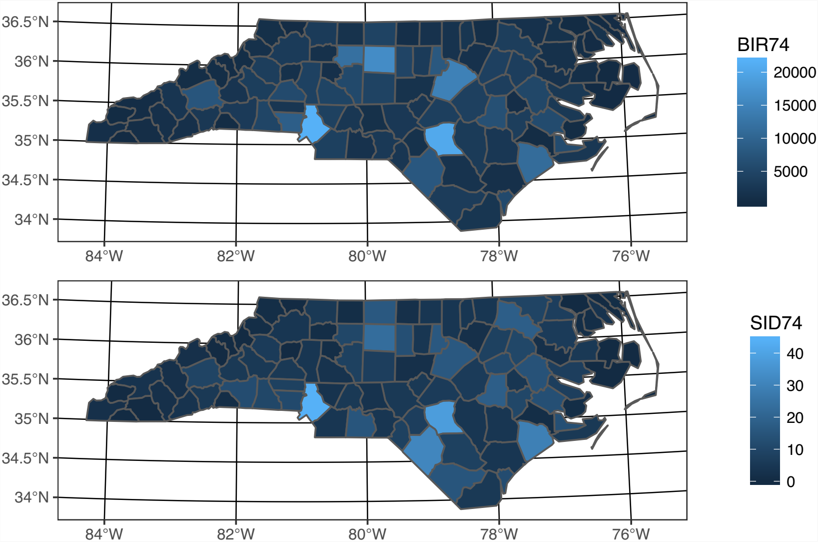
Conclusions

Generally speaking just like the AR(1) model for time series we require that $|\phi| < 1$ for the CAR model to be proper.

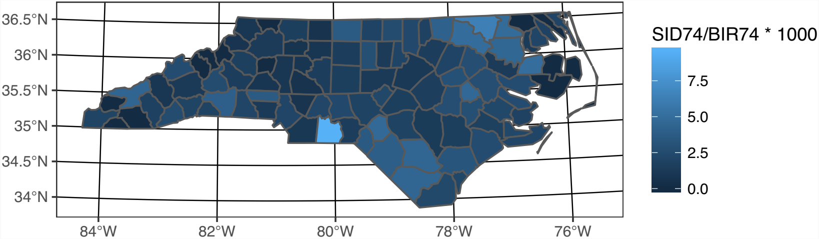
These results for ϕ also apply in the context where σ_i^2 is constant across locations (i.e. $\boldsymbol{\Sigma} = \left(\sigma^2 \left(\mathbf{I} - \phi \mathbf{D}^{-1}\mathbf{W}\right)\right)^{-1}$)

As a side note, the special case where $\phi = 1$ is known as an intrinsic autoregressive (IAR) model and they are popular as an *improper* prior for spatial random effects. An additional sum constraint is necessary for identifiability ($\sum_{i=1}^{n} y(s_i) = 0$).

Example - NC SIDS



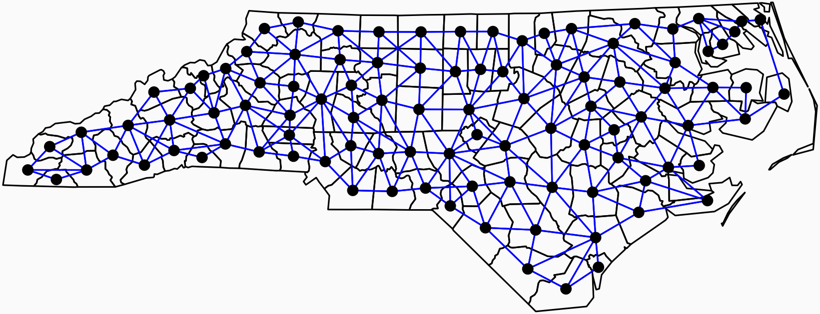
```r
ggplot() + geom_sf(data=nc, aes(fill=SID74/BIR74*1000))
```



Using spautolm from spdep

```r
library(spdep)

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

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
```

Geary's C

```
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")
```

spdep

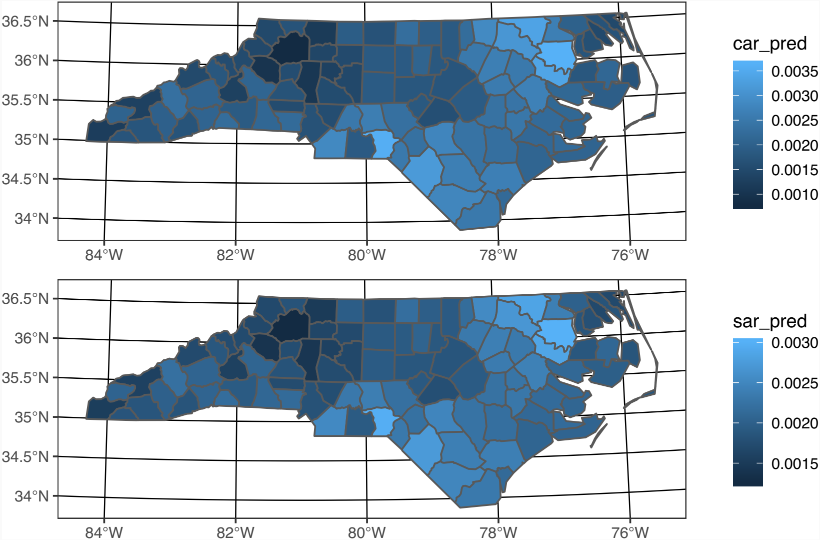
```
summary(nc_car)
##
## Call: spautolm(formula = SID74/BIR74 ~ 1, data = nc, listw = listW,
##     family = "CAR")
##
## Residuals:
##         Min          1Q      Median          3Q         Max
## -0.00213872 -0.00083535 -0.00022355  0.00055014  0.00768640
##
## Coefficients:
##                Estimate 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: 0.030469
##
## 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
```

ϕ

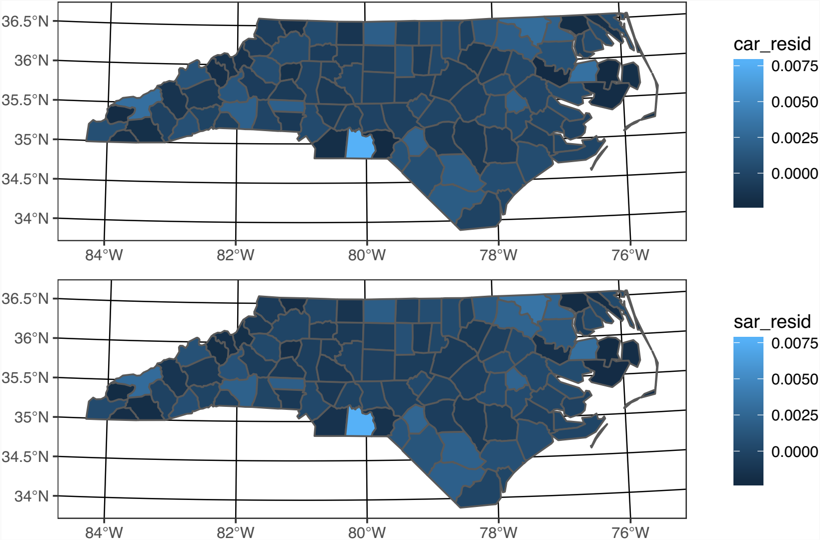
SAR Model

```
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
```

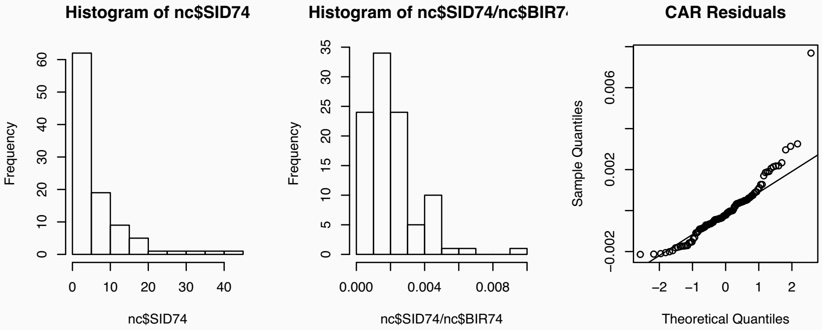
Predictions



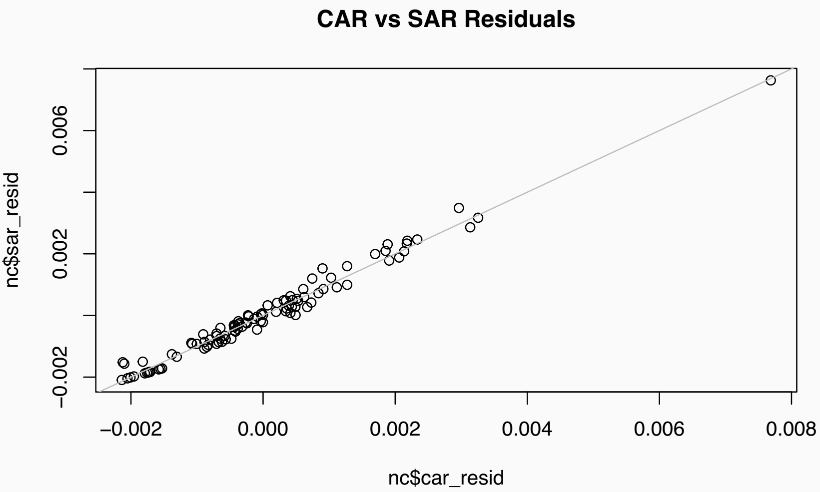
Residuals


36.5°N
36°N
35.5°N
35°N
34.5°N
34°N
84°W
82°W
80°W
78°W
76°W
car_resid
0.0075
0.0050
0.0025
0.0000
sar_resid
0.0075
0.0050
0.0025
0.0000

What's wrong?



Histogram of nc$SID74
Frequency
0
10
20
30
40
50
60
0
10
20
30
40
nc$SID74
Histogram of nc$SID74/nc$BIR7
Frequency
0
5
10
15
20
25
30
35
0.000
0.004
0.008
nc$SID74/nc$BIR74
CAR Residuals
Sample Quantiles
−0.002
0.002
0.006
−2
−1
0
1
2
Theoretical Quantiles


CAR vs SAR Residuals
nc$sar_resid
−0.002
0.002
0.006
nc$car_resid
−0.002
0.000
0.002
0.004
0.006
0.008

Stan CAR Model

```
car_model = "
data {
  int<lower=0> N;
  vector[N] y;
  matrix[N,N] W;
  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 * W) / 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);
}
```

$$y = XB + W(s) + \varepsilon$$

$$\varepsilon_i \sim N(0, \sigma^2_{\varepsilon})$$

$$w(s) \sim N\left(0, (D - \phi W)^{-1} \sigma^2\right)$$

```r
data = list(
  N = nrow(nc),
  y = nc$SID74 / nc$BIR74,
  W = W * 1,
  D = diag(rowSums(W))
)
```

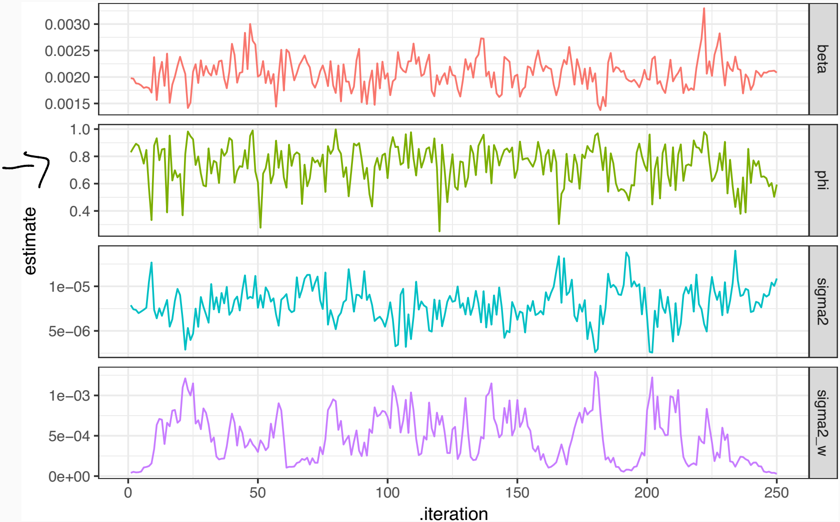
```r
car_fit = rstan::stan(
  model_code = car_model, data = data,
  iter = 10000, chains = 1, thin=20
)
```

```r
data = list(
  N = nrow(nc),
  y = nc$SID74 / nc$BIR74,
  W = W * 1,
  D = diag(rowSums(W))
)
```

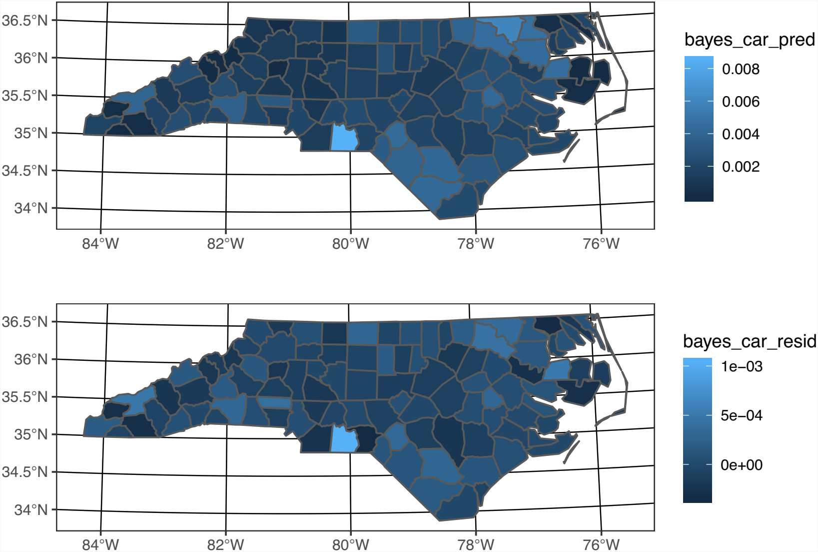
```r
car_fit = rstan::stan(
  model_code = car_model, data = data,
  iter = 10000, chains = 1, thin=20
)
```

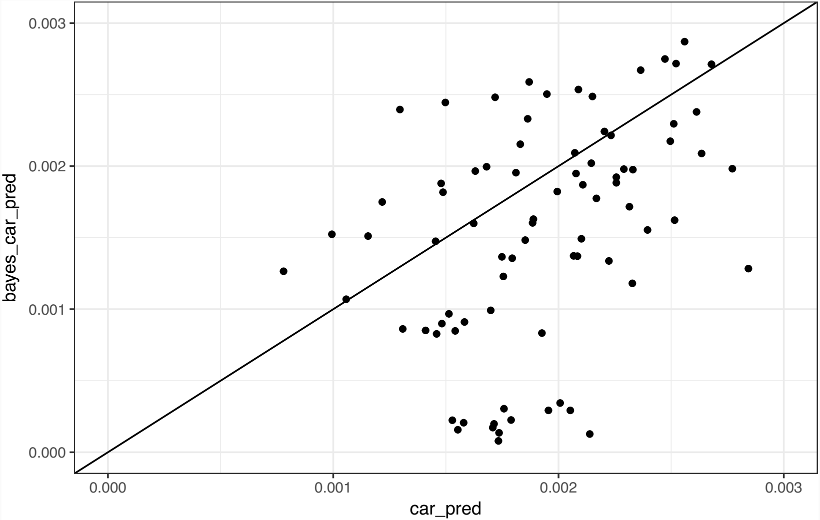
Why don't we use the conditional definition for the y's?

Model Results



Predictions


36.5°N
36°N
35.5°N
35°N
34.5°N
34°N
84°W
82°W
80°W
78°W
76°W
bayes_car_pred
0.008
0.006
0.004
0.002
bayes_car_resid
1e−03
5e−04
0e+00


0.003
0.002
0.001
0.000
bayes_car_pred
0.000
0.001
0.002
0.003
car_pred

Brief Aside - SAR Precision Matrix

$$\Sigma_{SAR} = (\mathbf{I} - \phi \mathbf{D}^{-1}\,\mathbf{W})^{-1} \sigma^2\, \mathbf{D}^{-1} \left((\mathbf{I} - \phi \mathbf{D}^{-1}\,\mathbf{W})^{-1} \right)^t$$

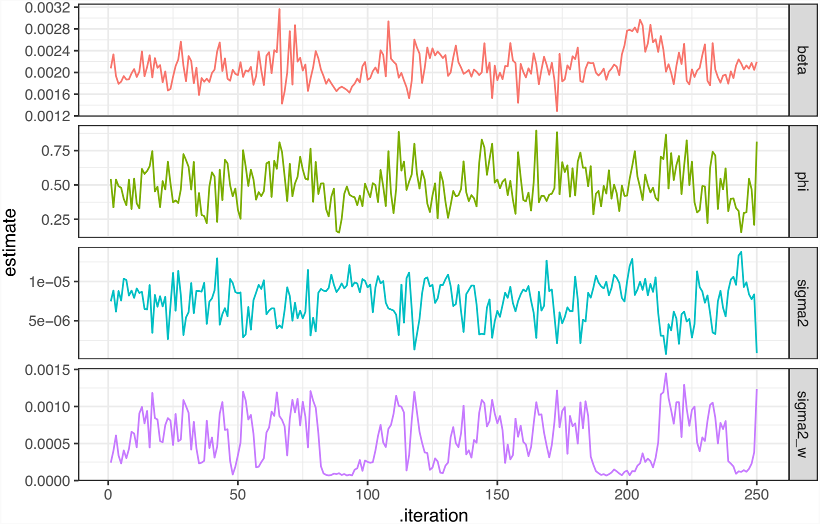
$$\begin{aligned}
\Sigma_{SAR}^{-1} &= \left((\mathbf{I} - \phi \mathbf{D}^{-1}\,\mathbf{W})^{-1} \sigma^2\, \mathbf{D}^{-1} \left((\mathbf{I} - \phi \mathbf{D}^{-1}\,\mathbf{W})^{-1} \right)^t \right)^{-1} \\
&= \left(\left((\mathbf{I} - \phi \mathbf{D}^{-1}\,\mathbf{W})^{-1} \right)^t \right)^{-1}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

Jags SAR Model

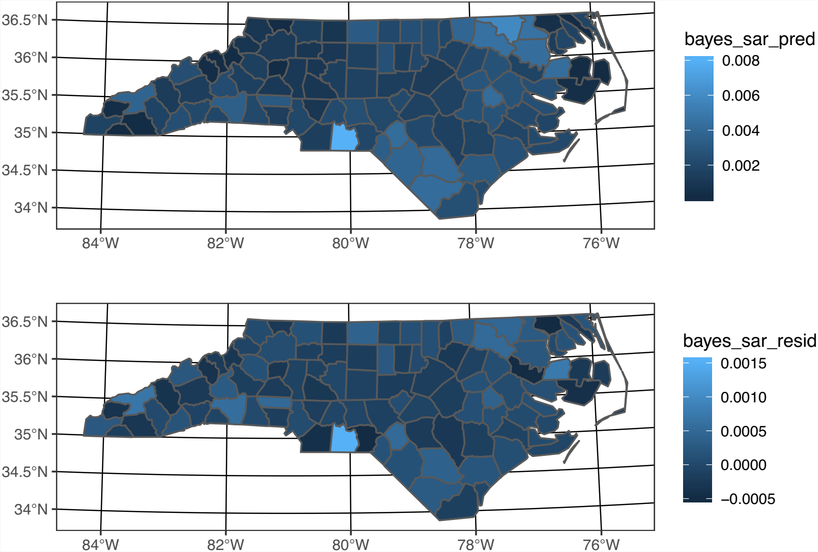
```
sar_model = "
data {
  int<lower=0> N;
  vector[N] y;
  matrix[N,N] W_tilde;
  matrix[N,N] D;
}
transformed data {
  matrix[N,N] I = diag_matrix(rep_vector(1, N));
}
parameters {
  vector[N] w_s;
  real beta;
  real<lower=0> sigma2;
  real<lower=0> sigma2_w;
  real<lower=0,upper=1> phi;
}
transformed 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);
}
"
```

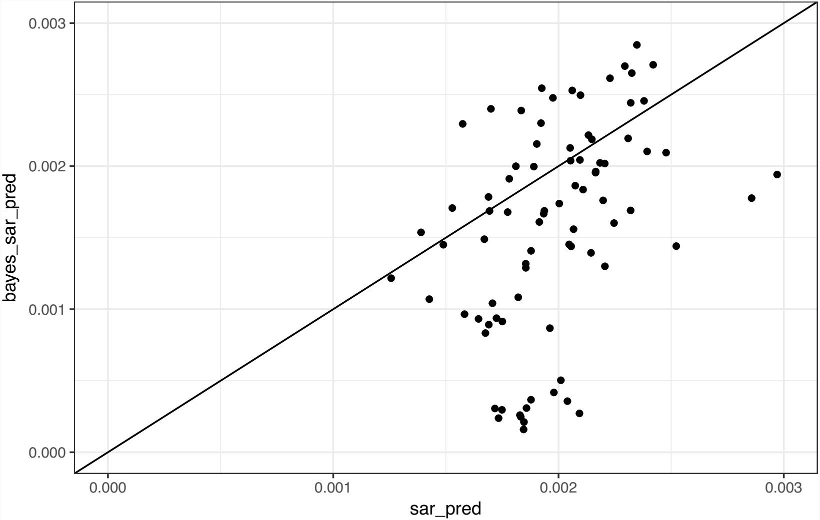
```
D = diag(rowSums(W))
D_inv = diag(1/diag(D))
data = list(
  N = nrow(nc),
  y = nc$SID74 / nc$BIR74,
  x = rep(1, nrow(nc)),
  D_inv = D_inv,
  W_tilde = D_inv %*% W
)

sar_fit = rstan::stan(
  model_code = sar_model, data = data,
  iter = 10000, chains = 1, thin=20
)
```

Model Results



Predictions




0.003
0.002
0.001
0.000
bayes_sar_pred
0.000
0.001
0.002
0.003
sar_pred

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