Fitting CAR and SAR Models

Lecture 19

Dr. Colin Rundel

Fitting areal models

Revised CAR Model

- Conditional Model

$$y(\boldsymbol{s}_i)|\boldsymbol{y}_{-s_i} \sim N\left(X_{i\cdot}\beta + \phi \sum_{j=1}^{n} \frac{A_{ij}}{D_{ii}} \, \big(y(\boldsymbol{s}_j) - X_{j\cdot}\beta\big), \; \sigma^2 D_{ii}^{-1}\right)$$

- Joint Model

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

SAR odel - mean structure

Let us consider what happens to our derivation of the SAR model when we include a $\boldsymbol{X\beta}$ in our formula model.

$$\boldsymbol{y} = \boldsymbol{X\beta} + \phi \boldsymbol{D}^{-1} \boldsymbol{A y} + \boldsymbol{\epsilon}$$
$$\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{D}^{-1})$$

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

$$\boldsymbol{y} = (I - \phi\, \boldsymbol{D}^{-1}\, \boldsymbol{A})^{-1} \boldsymbol{X\beta} + (I - \phi\, \boldsymbol{D}^{-1}\, \boldsymbol{A})^{-1} \boldsymbol{\epsilon}$$

Properties

$$E(\boldsymbol{y}) = (I - \phi\,\boldsymbol{D}^{-1}\,\boldsymbol{A})^{-1}\boldsymbol{X}\boldsymbol{\beta}$$

$$\begin{aligned} Var(\boldsymbol{y}) &= \left((I - \phi\,\boldsymbol{D}^{-1}\,\boldsymbol{A})^{-1}\right)\sigma^2\boldsymbol{D}^{-1}\left((I - \phi\,\boldsymbol{D}^{-1}\,\boldsymbol{A})^{-1}\right)^t \\ &= \sigma^2\left((I - \phi\,\boldsymbol{D}^{-1}\,\boldsymbol{A})^{-1}\right)\boldsymbol{D}^{-1}\left((I - \phi\,\boldsymbol{D}^{-1}\,\boldsymbol{A})^{-1}\right)^t \end{aligned}$$

This is the same behavior we saw with the AR(1) model, where the mean of the process is $\delta/(1-\phi)$ and **not** δ.

This is a relatively minor issue but it does have practical implications when it comes to interpretation of the model parameters (particularly the slope coefficients $\boldsymbol{\beta}$).

SAR error model

The previous model is sometimes refered to as the **SAR lag model**, a variation of this model, common in spatial econometrics, is the **SAR error model** which is defiend as follows,

$$\begin{aligned}
\boldsymbol{y} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{u} \\
\boldsymbol{u} &= \phi \boldsymbol{D}^{-1} \boldsymbol{A}\boldsymbol{u} + \boldsymbol{\epsilon} \\
\boldsymbol{\epsilon} &\sim N(\boldsymbol{0},\, \sigma^2 \boldsymbol{D}^{-1})
\end{aligned}$$

As with the SAR lag model we can solve for $\boldsymbol{u}$,

$$\begin{aligned}
\boldsymbol{u} &= \phi \boldsymbol{D}^{-1} \boldsymbol{A}\boldsymbol{u} + \boldsymbol{\epsilon} \\
\boldsymbol{u} - \phi \boldsymbol{D}^{-1} \boldsymbol{A}\boldsymbol{u} &= \boldsymbol{\epsilon} \\
\boldsymbol{u} &= (I - \phi \boldsymbol{D}^{-1} \boldsymbol{A})^{-1} \boldsymbol{\epsilon}
\end{aligned}$$

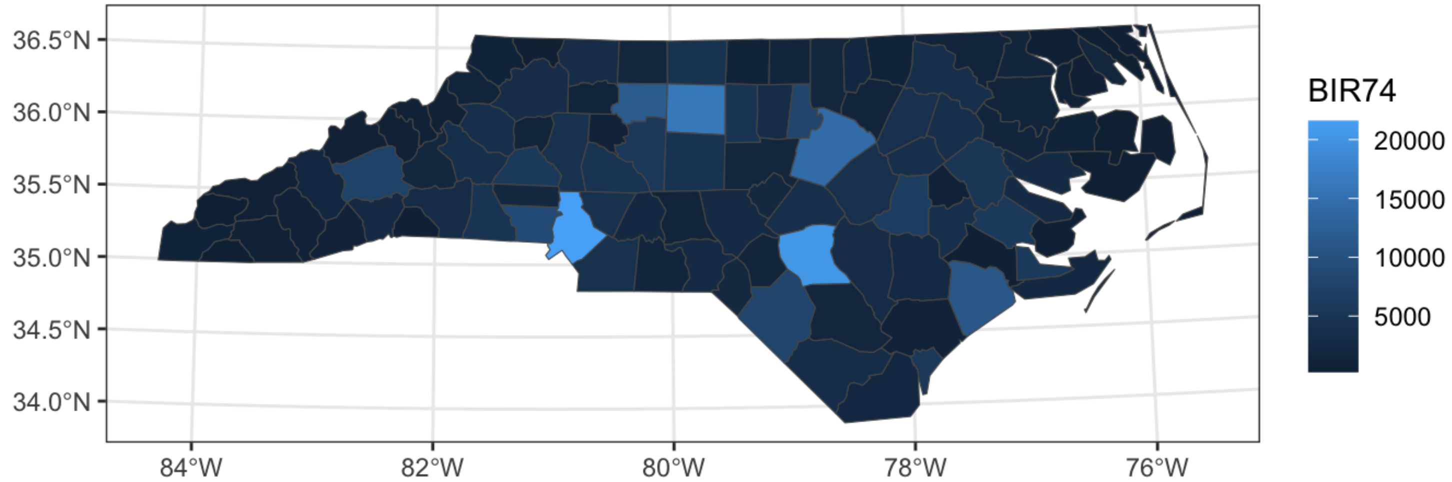
Properties

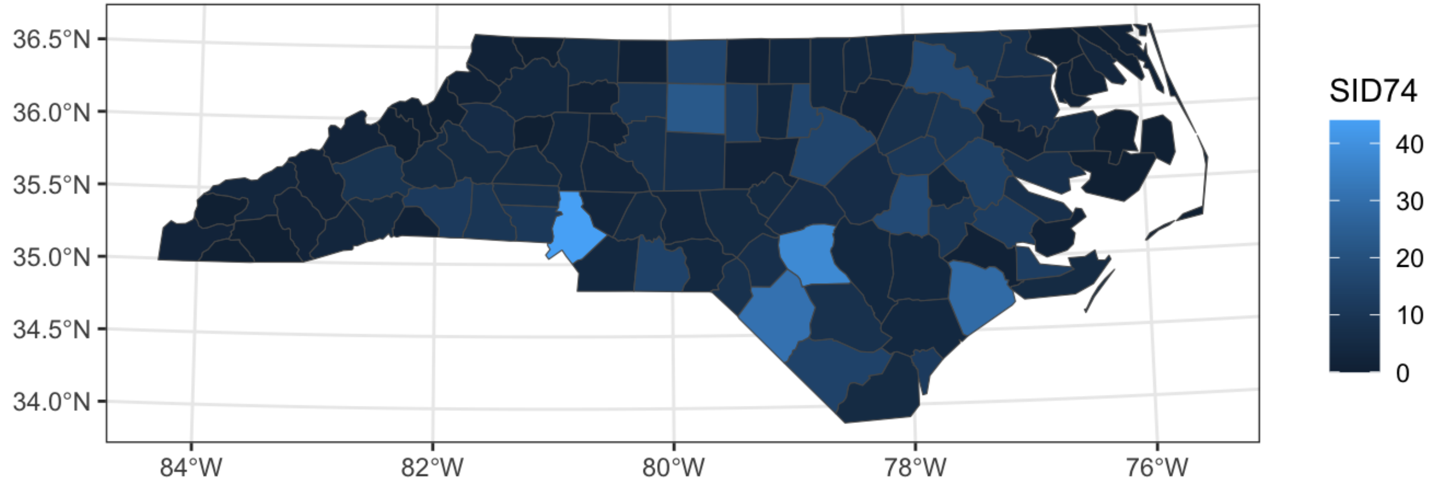
$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + (I - \phi \boldsymbol{D}^{-1}\, \boldsymbol{A})^{-1} \boldsymbol{\epsilon}$$

$$E(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\beta}$$

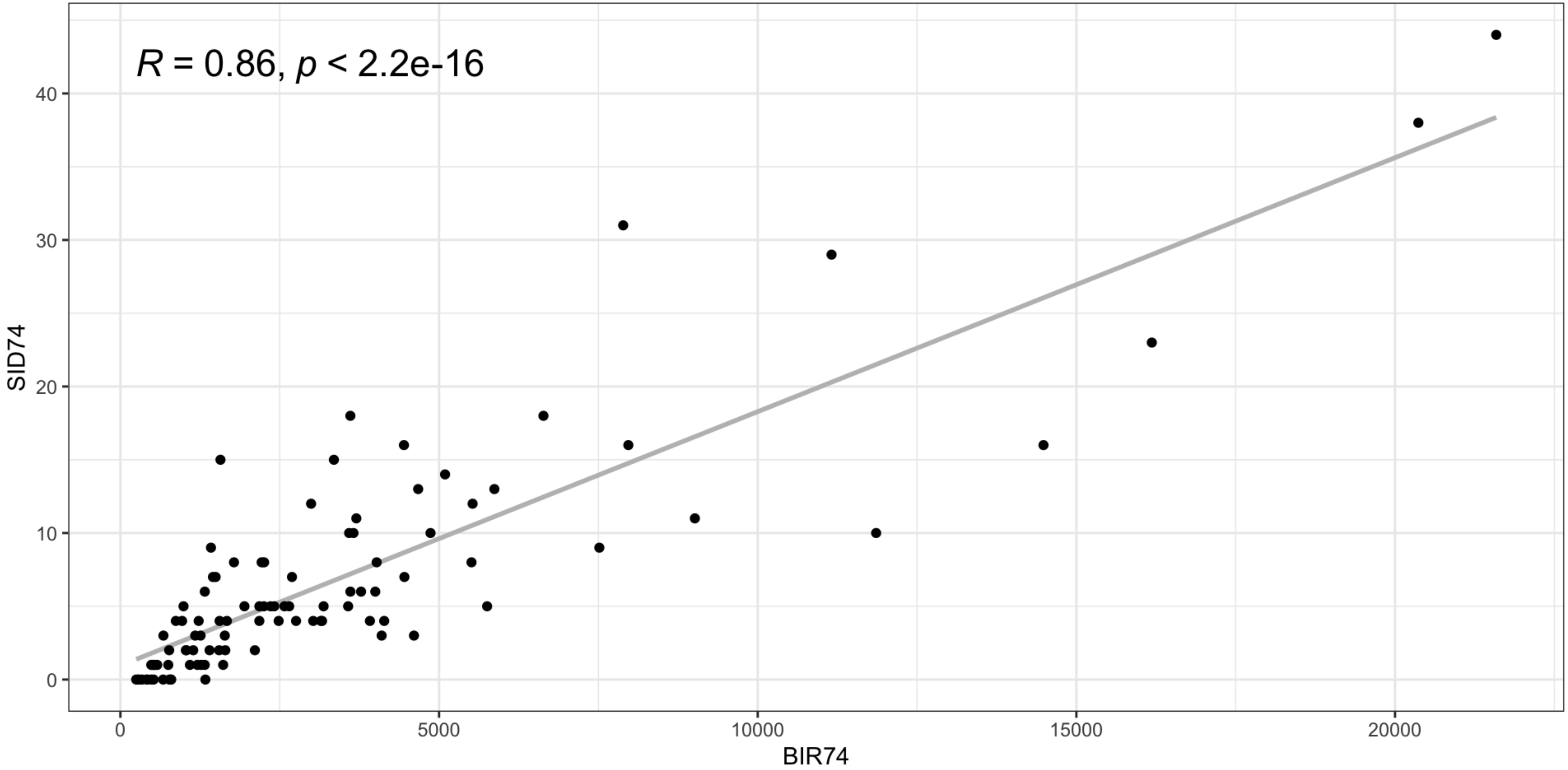
$$Var(\boldsymbol{y}) = \sigma^2 \left((I - \phi\, \boldsymbol{D}^{-1}\, \boldsymbol{A})^{-1} \right) \boldsymbol{D}^{-1} \left((I - \phi\, \boldsymbol{D}^{-1}\, \boldsymbol{A})^{-1} \right)^t$$

Example - NC SIDS

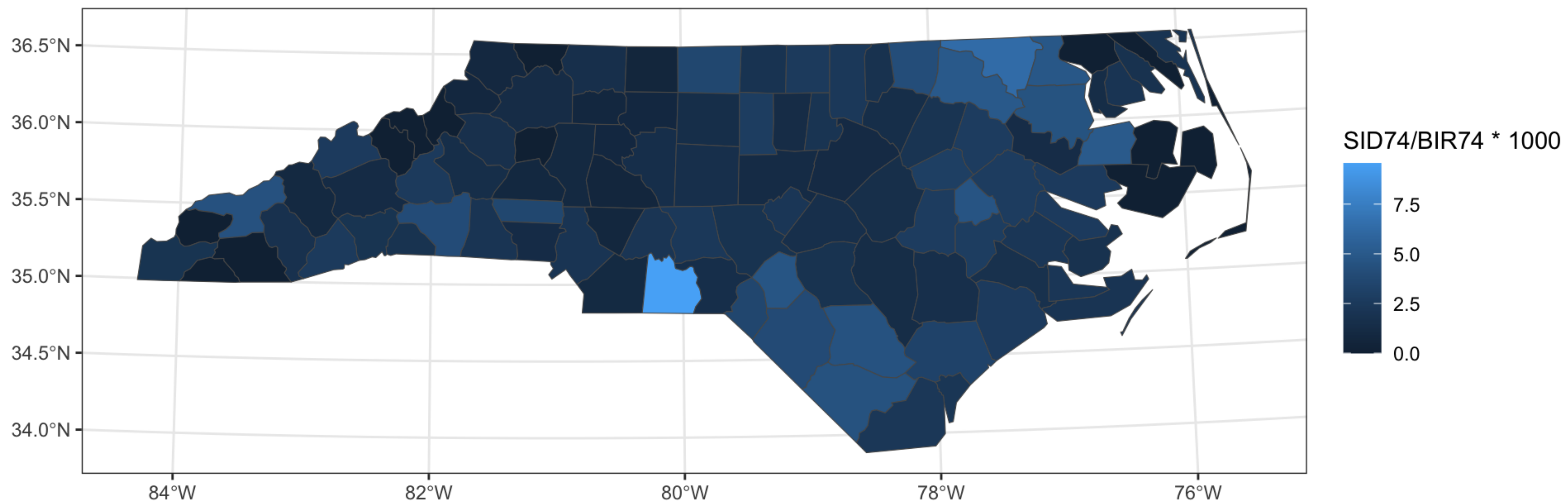




BIR74 vs SID74



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



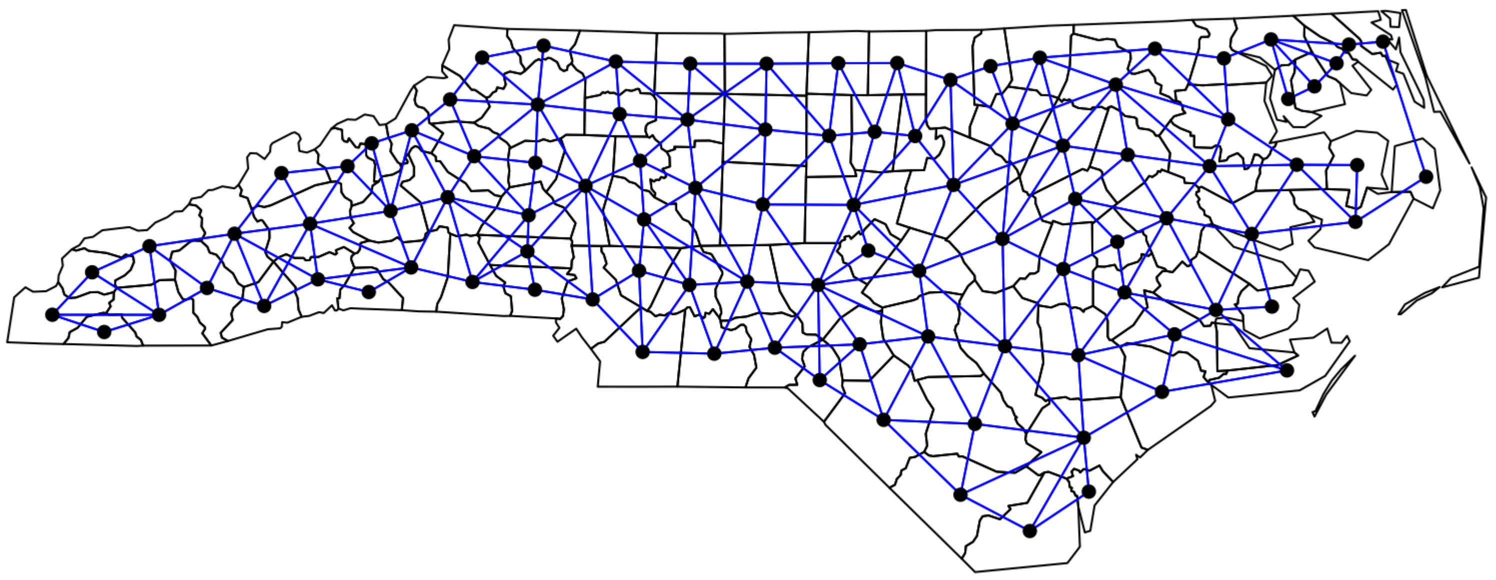
Using spdep + spatialreg

```r
A = st_touches(nc, sparse=FALSE)
(listW = spdep::mat2listw(A))
```

```
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
```

Plotting listw

```r
nc_coords = nc |> st_centroid() |> st_coordinates()
plot(st_geometry(nc))
plot(listW, nc_coords, add=TRUE, col="blue", pch=16)
```



Moran's I

```r
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 
      0.119089049      -0.010101010 
         Variance 
      0.003542176 
```

```r
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 
      0.210046454      -0.010101010 
         Variance 
      0.003666802 
```

Geary's C

```r
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
       0.88988684        1.00000000
         Variance
       0.01434105
```

```r
spdep::geary.test(
  1000*nc$SID74/nc$BIR74, listW
)
```

```
	Geary C test under randomisation

data:  1000 * 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
       0.67796679        1.00000000
         Variance
       0.01079878
```

CAR Model

```r
nc_car = spatialreg::spautolm(
  formula = 1000*SID74/BIR74 ~ 1, data = nc,
  listw = listW, family = "CAR"
)
summary(nc_car)
```

```

Call: spatialreg::spautolm(formula = 1000 * SID74/BIR74 ~ 1, data = nc,
    listw = listW, family = "CAR")

Residuals:
     Min       1Q   Median       3Q      Max 
-2.13872 -0.83535 -0.22355  0.55014  7.68640 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.00203    0.24272  8.2484 2.22e-16

Lambda: 0.13062 LR test value: 8.6251 p-value: 0.0033157 
Numerical Hessian standard error of lambda: 0.030475 

Log likelihood: -182.3989 
ML residual variance (sigma squared): 2.1205, (sigma: 1.4562)
Number of observations: 100 
```

SAR Model (error)

```r
nc_sar_err = spatialreg::spautolm(
  formula = 1000*SID74/BIR74 ~ 1, data = nc,
  listw = listW, family = "SAR"
)
summary(nc_sar_err)
```

```
Call: spatialreg::spautolm(formula = 1000 * SID74/BIR74 ~ 1, data = nc,
    listw = listW, family = "SAR")

Residuals:
     Min       1Q   Median       3Q      Max
-2.09307 -0.87039 -0.20274  0.51156  7.62830

Coefficients:
            Estimate Std. Error z value  Pr(>|z|)
(Intercept)  2.01084    0.23622  8.5127 < 2.2e-16

Lambda: 0.079934 LR test value: 8.8911 p-value: 0.0028657
Numerical Hessian standard error of lambda: 0.024597

Log likelihood: -182.2659
ML residual variance (sigma squared): 2.1622, (sigma: 1.4704)
Number of observations: 100
```

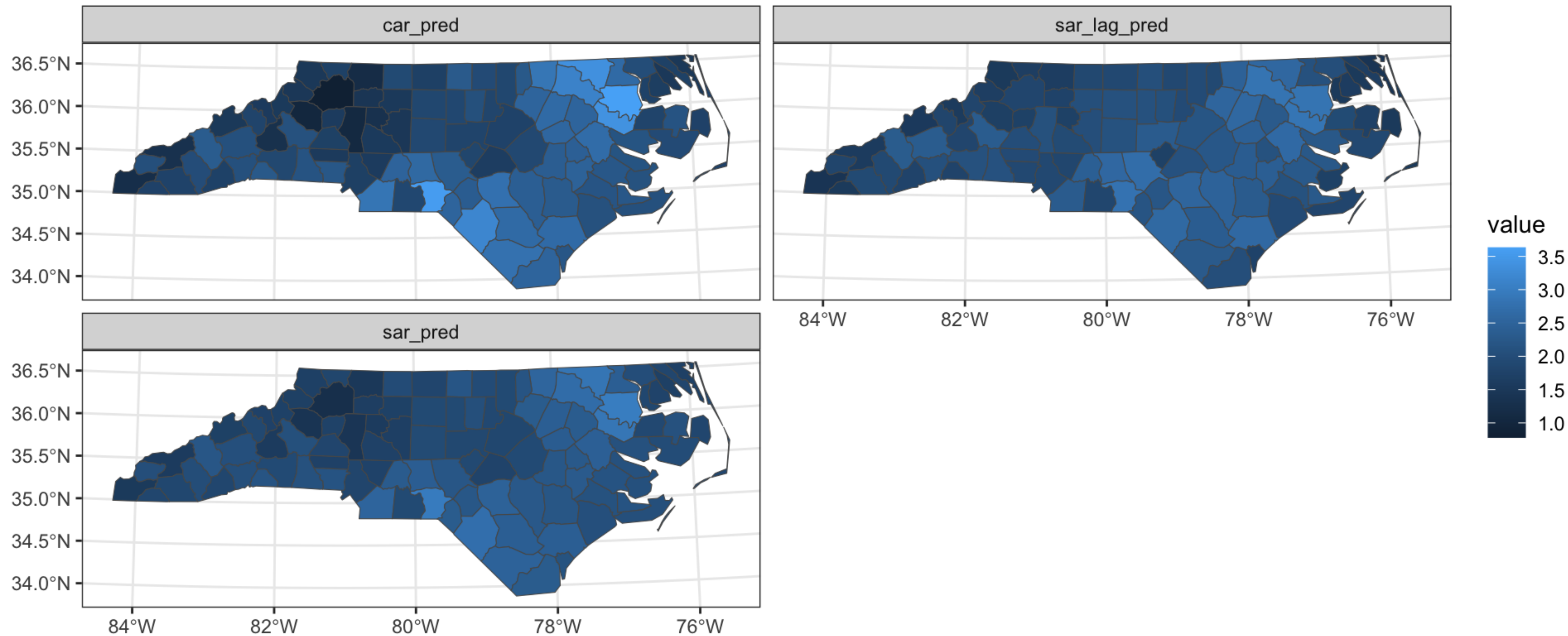
SAR Model (lag)

```r
nc_sar_lag = spatialreg::lagsarlm(
  formula = 1000*SID74/BIR74 ~ 1, data = nc,
  listw = listW
)
summary(nc_sar_lag)
```

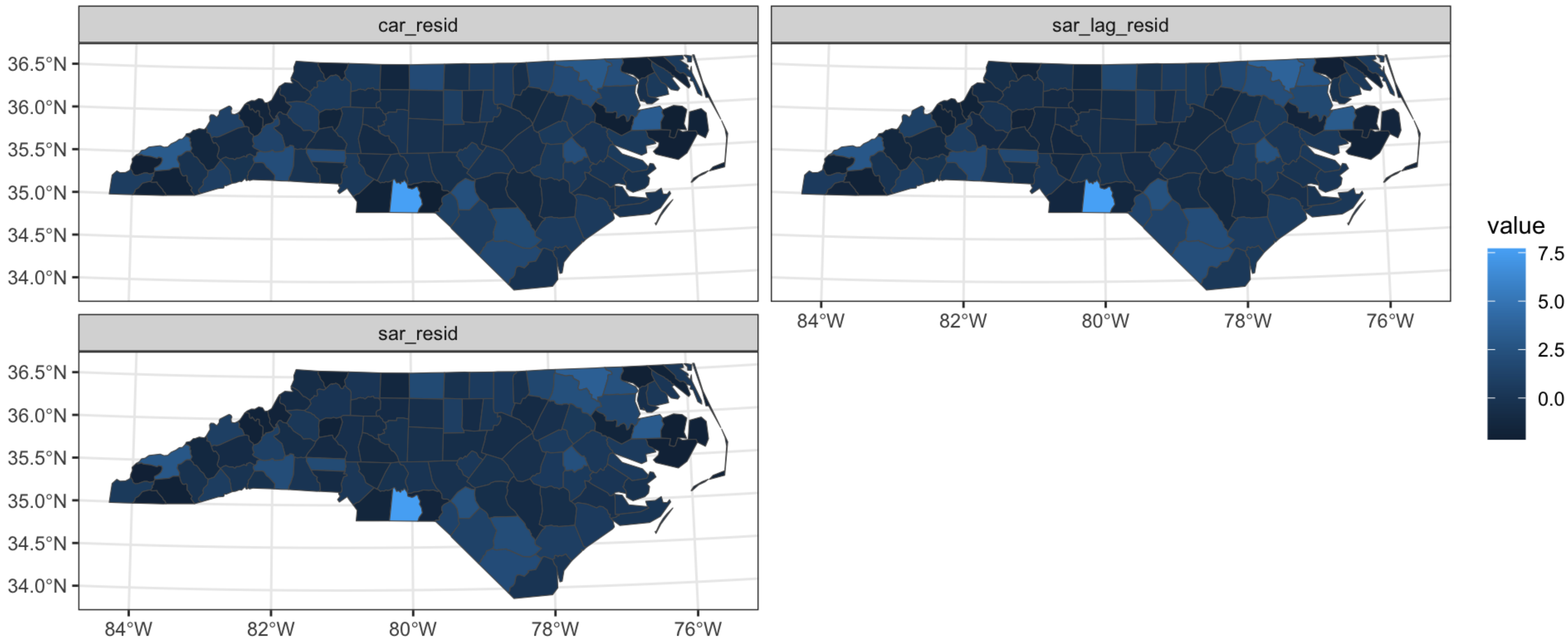
```

Call:spatialreg::lagsarlm(formula = 1000 * SID74/BIR74 ~ 1, data = nc,
    listw = listW)

Residuals:
     Min       1Q   Median       3Q      Max
-2.03968 -1.10398 -0.14413  0.54949  7.70889

Type: lag
Coefficients: (asymptotic standard errors)
            Estimate Std. Error z value  Pr(>|z|)
(Intercept)  1.40320    0.27296  5.1406 2.738e-07

Rho: 0.063331, LR test value: 7.5512, p-value: 0.0059971
Asymptotic standard error: 0.021655
    z-value: 2.9246, p-value: 0.0034493
Wald statistic: 8.5531, p-value: 0.0034493

Log likelihood: -182.9358 for lag model
```

```
ML residual variance (sigma squared): 2.2232, (sigma: 1.491)
Number of observations: 100
```

Predictions

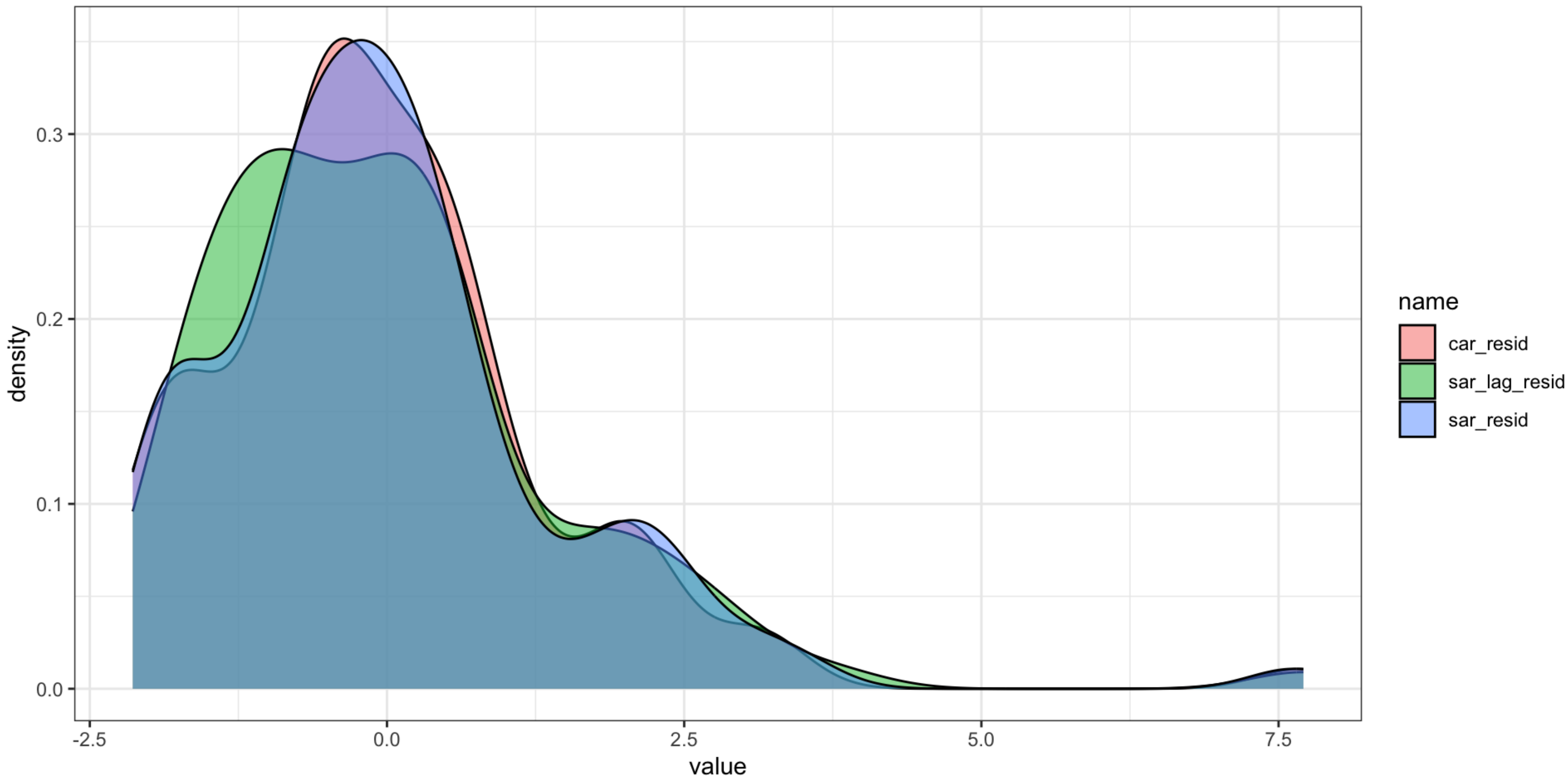


Residuals



car_resid
sar_lag_resid
sar_resid
36.5°N
36.0°N
35.5°N
35.0°N
34.5°N
34.0°N
84°W
82°W
80°W
78°W
76°W
value
7.5
5.0
2.5
0.0

Residual distributions



Residual autocorrelation

```
spdep::moran.test(
  residuals(nc_car), listW,
  alternative = "two.sided"
)
```

```
	Moran I test under randomisation

data:  residuals(nc_car)
weights: listW

Moran I statistic standard deviate = -1.7952, p-
value = 0.07261
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation
Variance
     -0.117449312      -0.010101010
0.003575538
```

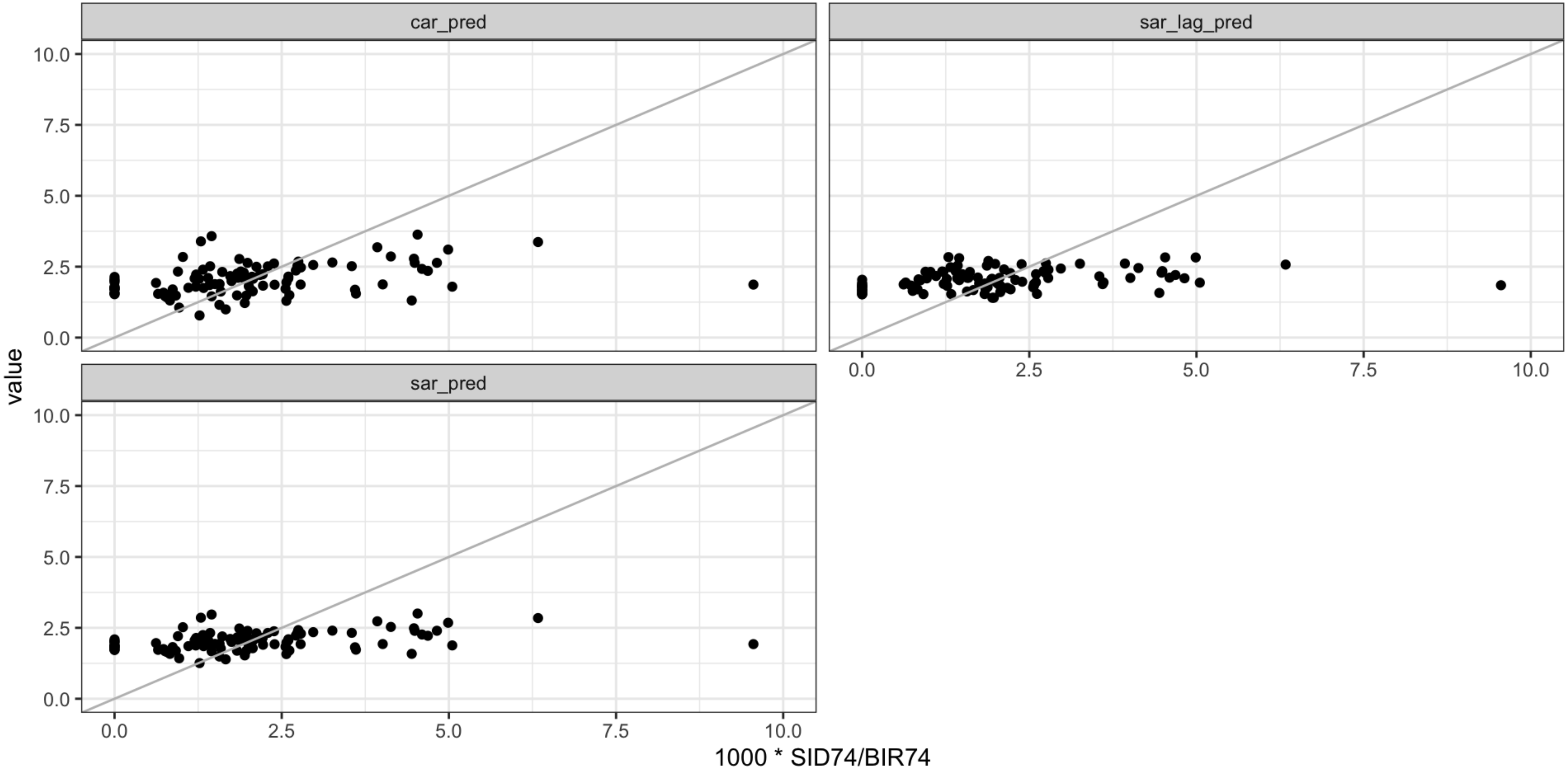
```
spdep::moran.test(
  residuals(nc_sar_err), listW,
  alternative = "two.sided"
)
```

```
	Moran I test under randomisation

data:  residuals(nc_sar_err)
weights: listW

Moran I statistic standard deviate = 0.17958, p-
value = 0.8575
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation
Variance
     0.0006769267     -0.0101010101
0.0036020941
```

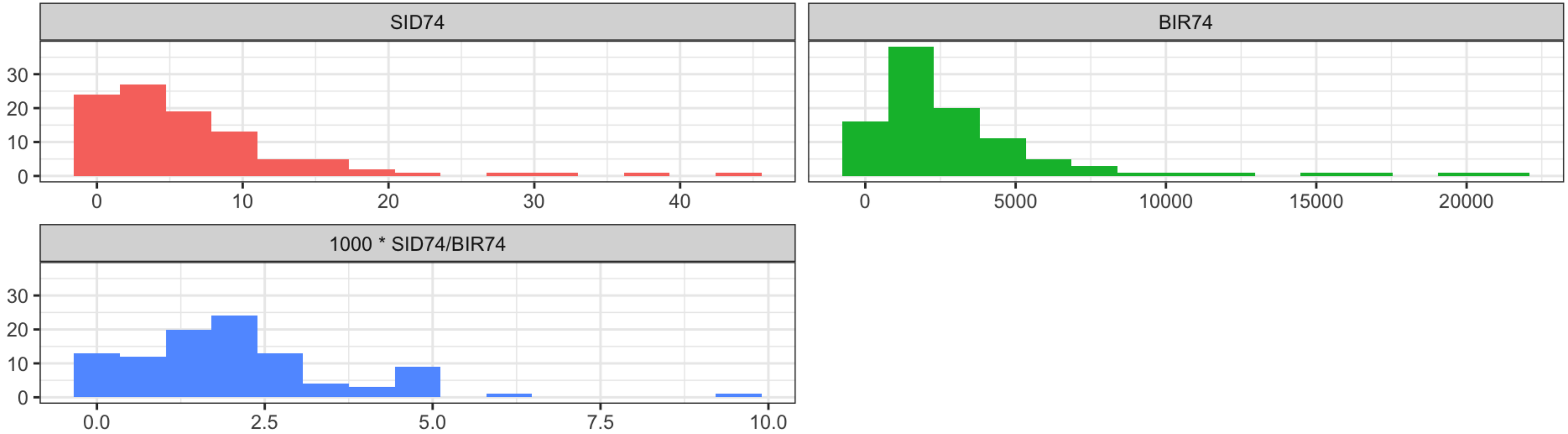
```r
spdep::moran.test(
  residuals(nc_sar_lag), listW,
  alternative = "two.sided"
)
```

```
	Moran I test under randomisation

data:  residuals(nc_sar_lag)
weights: listW

Moran I statistic standard deviate = 0.88877, p-
value = 0.3741
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation
Variance
      0.043255233      -0.010101010
0.003604055
```

Predicted vs Observed



What’s wrong?



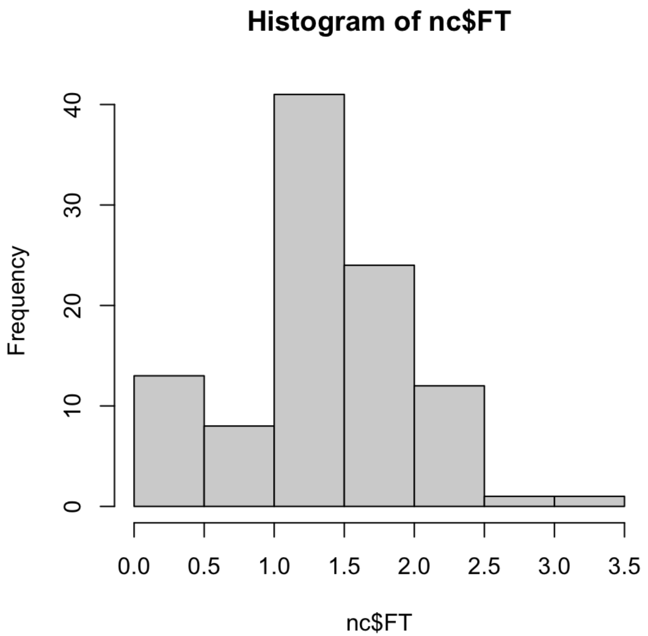
SID74
BIR74
1000 * SID74/BIR74
30
20
10
0
0
10
20
30
40
0
5000
10000
15000
20000
0.0
2.5
5.0
7.5
10.0

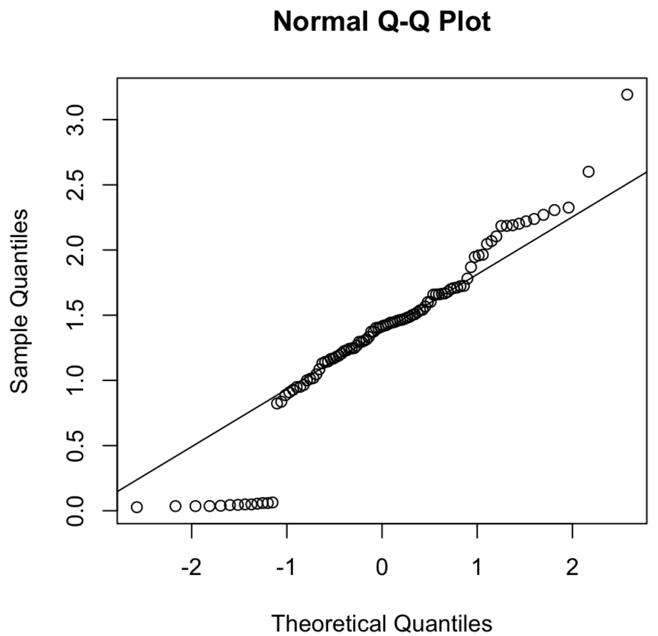
Transforming the data

Freeman-Tukey's transformation

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

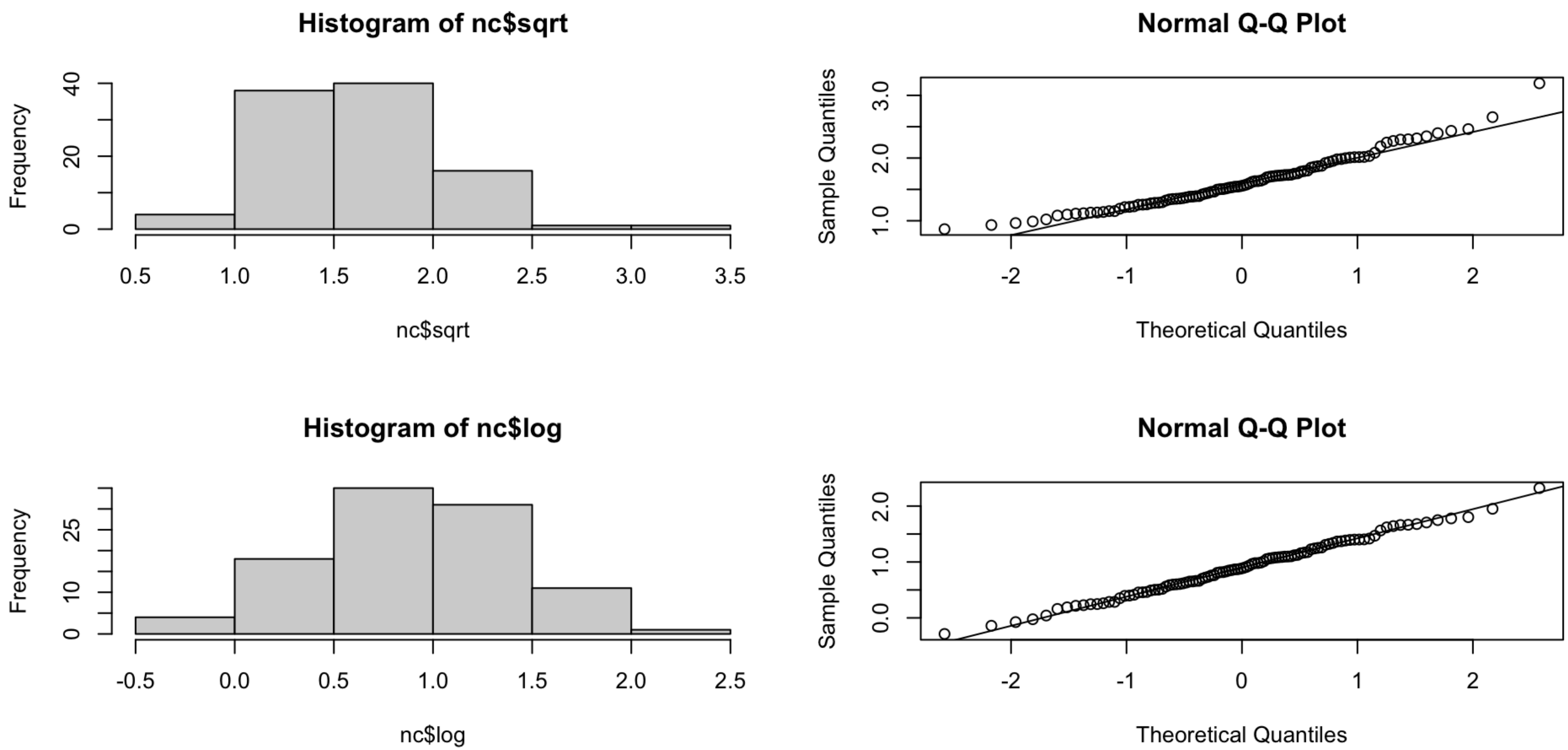
$$FT = \sqrt{1000}\left(\sqrt{\frac{SID74}{BIR74}} + \sqrt{\frac{SID74+1}{BIR74}}\right)$$





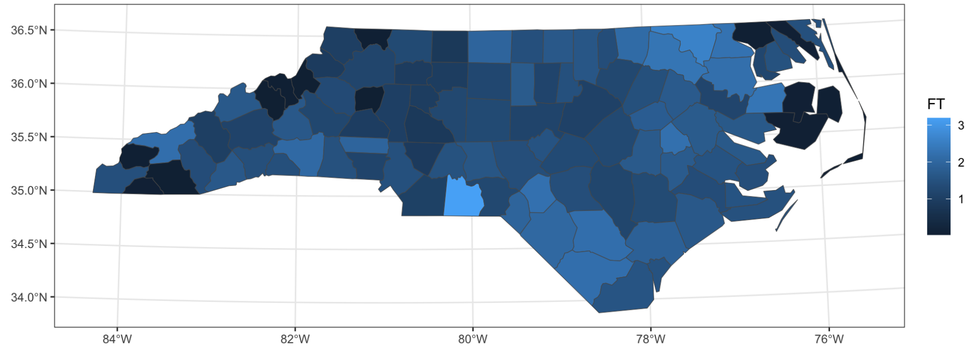
Other possibilities

```
nc = mutate(nc,
  sqrt = sqrt(1000*(SID74+1)/BIR74),
  log  = log(1000*(SID74+1)/BIR74),
)
```



FT transformation

```r
ggplot(nc) + geom_sf(aes(fill=FT))
```

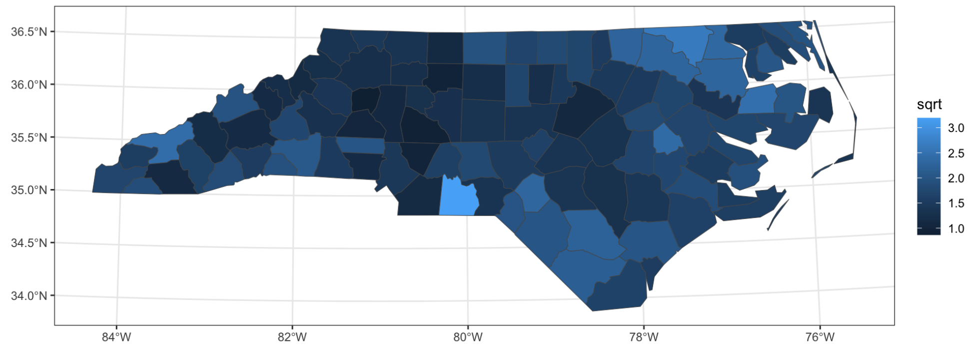



```r
spdep::moran.test(nc$FT, listW)
```

```
	Moran I test under randomisation

data:  nc$FT  
weights: listW    

Moran I statistic standard deviate = 3.664, p-
value = 0.0001242
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation
Variance 
      0.216246481      -0.010101010
0.003816298 
```

sqrt transformation

```r
ggplot(nc) + geom_sf(aes(fill=sqrt))
```



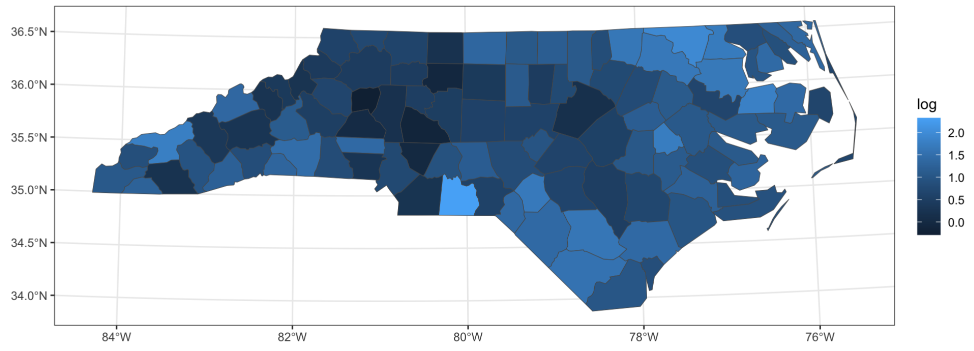

```r
spdep::moran.test(nc$sqrt, listW)
```

```
	Moran I test under randomisation

data:  nc$sqrt
weights: listW

Moran I statistic standard deviate = 4.5217, p-
value = 3.067e-06
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation
Variance
      0.268600322      -0.010101010
0.003798988
```

log transformation

```r
ggplot(nc) + geom_sf(aes(fill=log))
```

```r
spdep::moran.test(nc$log, listW)
```




```
	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 
```

Models

CAR

```r
nc_car_ft   = spatialreg::spautolm(formula = FT ~ 1,   data = nc, listw = listW, family = "CAR")
nc_car_sqrt = spatialreg::spautolm(formula = sqrt ~ 1, data = nc, listw = listW, family = "CAR")
nc_car_log  = spatialreg::spautolm(formula = log ~ 1,  data = nc, listw = listW, family = "CAR")
```

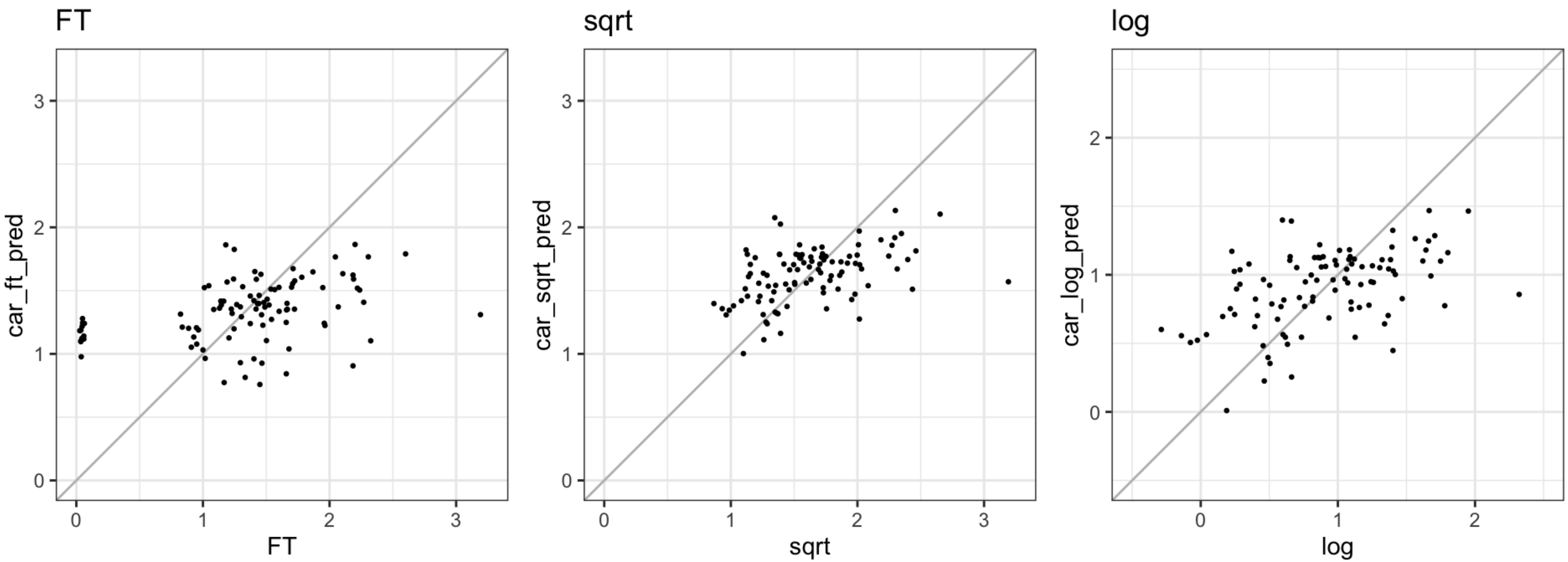
SAR (error)

```r
nc_sar_err_ft   = spatialreg::spautolm(formula = FT ~ 1,   data = nc, listw = listW, family = "S
nc_sar_err_sqrt = spatialreg::spautolm(formula = sqrt ~ 1, data = nc, listw = listW, family = "S
nc_sar_err_log  = spatialreg::spautolm(formula = log ~ 1,  data = nc, listw = listW, family = "S
```

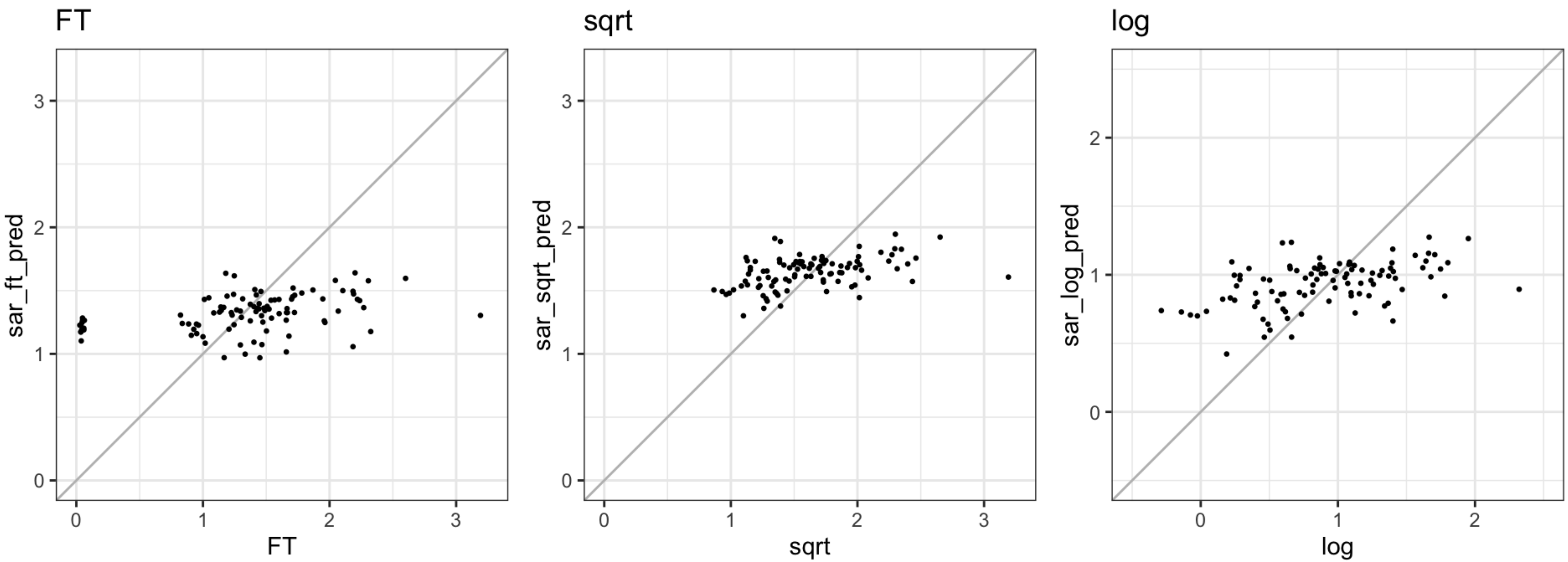
SAR (lag)

```r
nc_sar_lag_ft   = spatialreg::lagsarlm(formula = FT ~ 1,   data = nc, listw = listW)
nc_sar_lag_sqrt = spatialreg::lagsarlm(formula = sqrt ~ 1, data = nc, listw = listW)
nc_sar_lag_log  = spatialreg::lagsarlm(formula = log ~ 1,  data = nc, listw = listW)
```

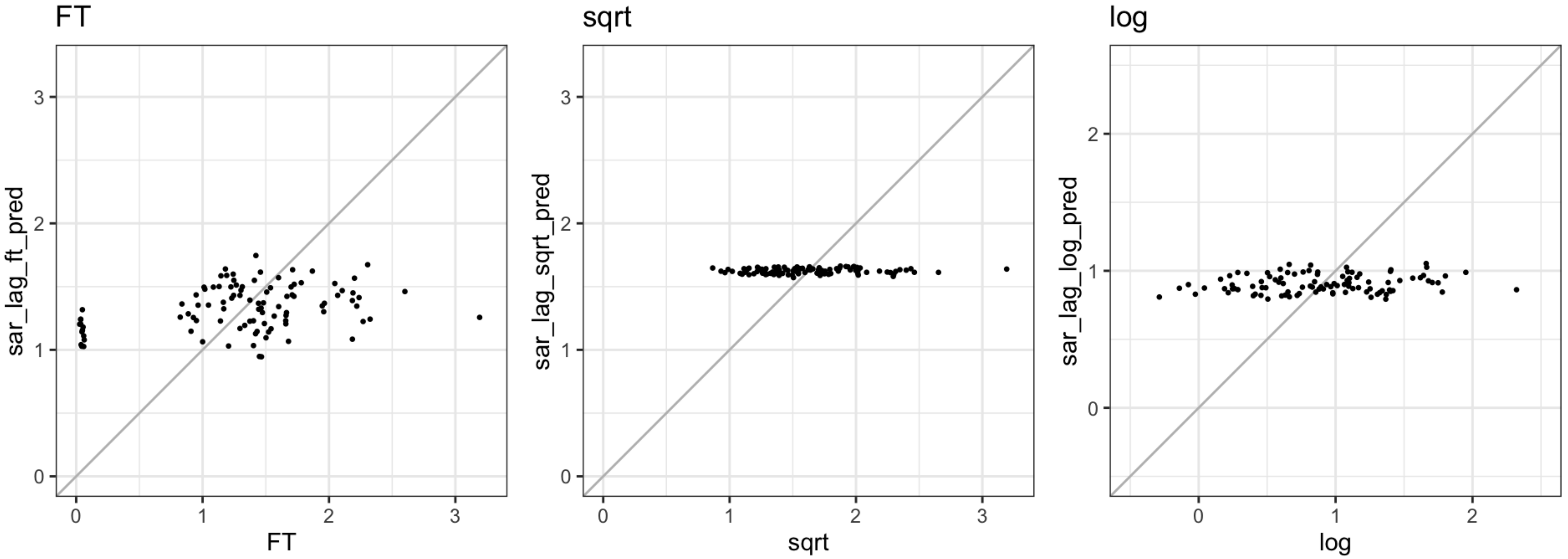
CAR predictions



SAR (error) predictions



SAR (lag) predictions



Residual spatial autocorrelation

```r
spdep::moran.test(
  residuals(nc_car_sqrt), listW
)
```

```
	Moran I test under randomisation

data:  residuals(nc_car_sqrt)
weights: listW

Moran I statistic standard deviate = -3.1196, p-
value = 0.9991
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation
Variance
     -0.200890550      -0.010101010
0.003740354
```

```r
spdep::moran.test(
  residuals(nc_sar_err_sqrt), listW
)
```

```
	Moran I test under randomisation

data:  residuals(nc_sar_err_sqrt)
weights: listW

Moran I statistic standard deviate = -0.422, p-
value = 0.6635
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation
Variance
     -0.035976084      -0.010101010
0.003759585
```

```r
spdep::moran.test(
  residuals(nc_sar_lag_sqrt), listW
)
```

```
	Moran I test under randomisation

data:  residuals(nc_sar_lag_sqrt)  
weights: listW    

Moran I statistic standard deviate = 4.7893, p-
value = 8.368e-07
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation           
Variance 
      0.285100348      -0.010101010       
0.003799187 
```

CAR with brms (and Stan)

brms CAR

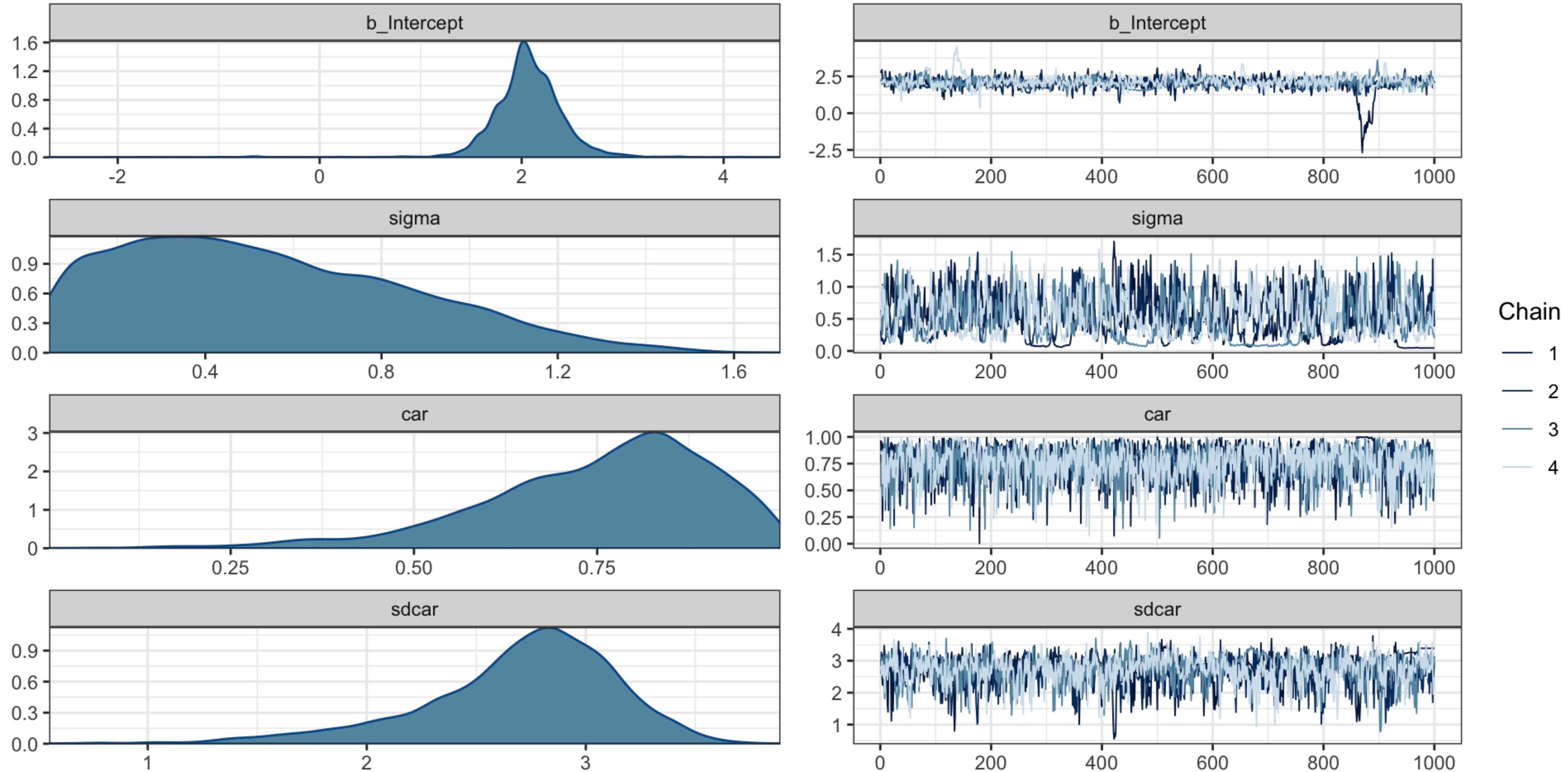
```r
rownames(A) = nc$NAME
colnames(A) = nc$NAME

b_car = brms::brm(
    1000*SID74/BIR74 ~ 1 + car(A, gr=NAME),
    data=nc, data2=list(A=A),
    control = list(adapt_delta = 0.95),
    iter=20000,
    cores = 4,
    thin=10
    #prior = c(brms::prior(normal(0, 0.01), class = sigma))
)
```

```r
b_car
```

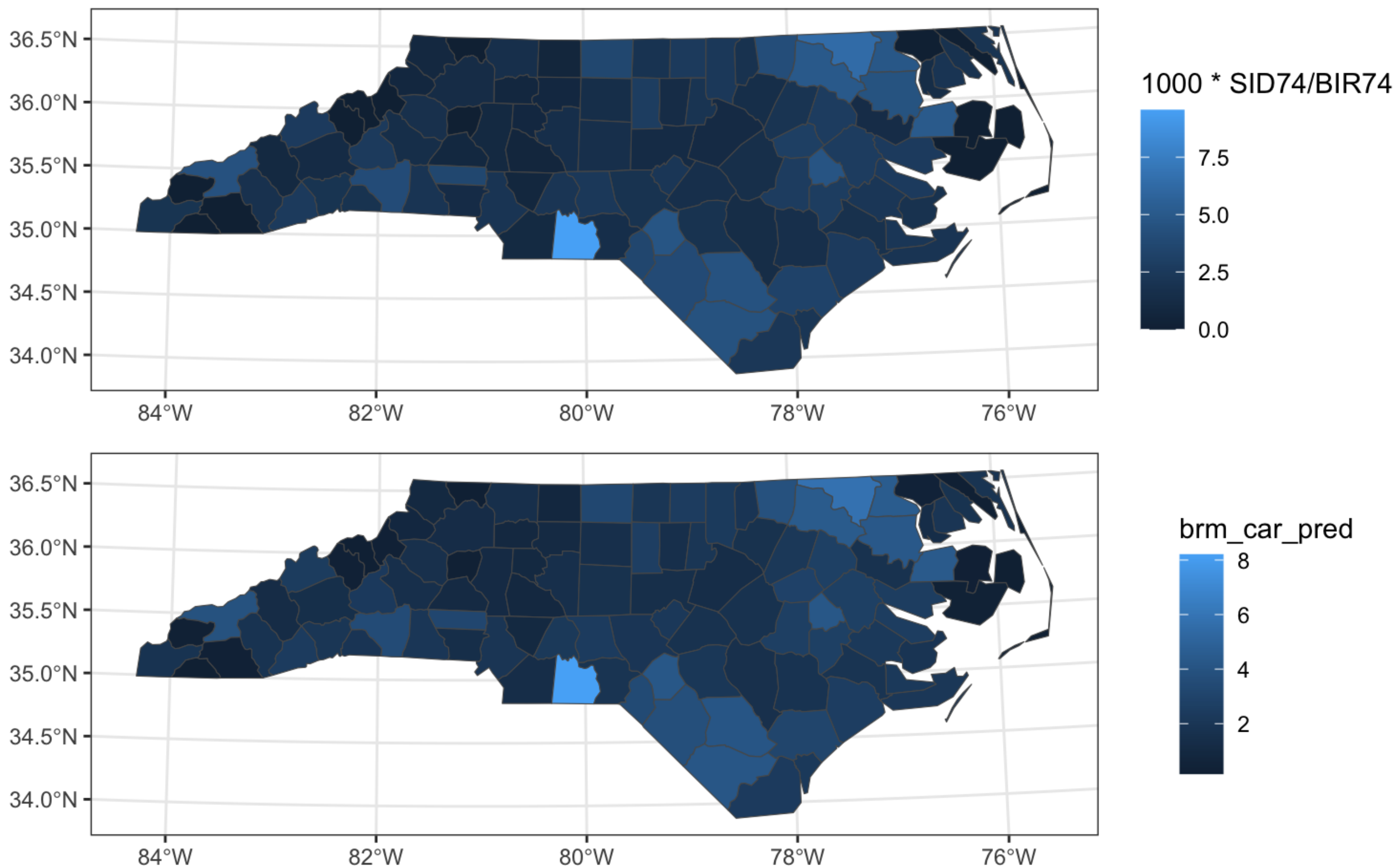
```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: 1000 * SID74/BIR74 ~ 1 + car(A, gr = NAME)
   Data: nc (Number of observations: 100)
  Draws: 4 chains, each with iter = 20000; warmup = 10000; thin = 10;
         total post-warmup draws = 4000

Correlation Structures:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
car       0.75      0.16     0.36     0.98 1.00      912      593
sdcar     2.70      0.45     1.60     3.39 1.01      352      579

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     2.06      0.41     1.43     2.76 1.01      423      467

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.54      0.33     0.07     1.24 1.03      133       84

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
```

Diagnostics

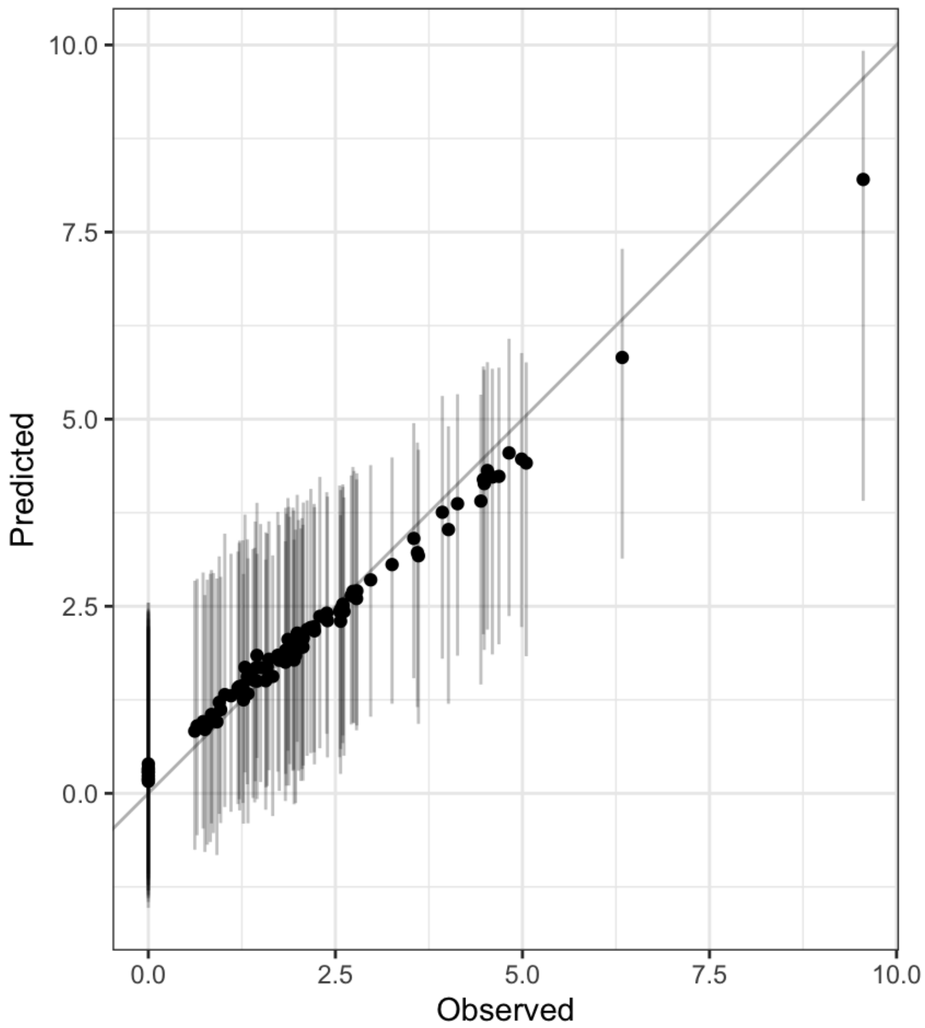
```
plot(b_car)
```



Predictions



Observed vs predicted



Stan (correct model)

```
stan_car = rstan::stan_model(
  model_code = "
    data {
      int<lower=1> n;
      int<lower=1> p;
      matrix[n, p] X;
      vector[n] y;
      matrix<lower=0, upper=1>[n, n] W;
      matrix[n, n] D;
      matrix[n, n] D_inv;
    }
    parameters {
      vector[p] beta;
      real<lower=0> tau;
      real<lower=0, upper=1> alpha;
    }
    transformed parameters {
      vector[n] y_cond = X * beta + alpha * I
      real<lower=0> sigma2 = 1/tau;
    }
    model {
```

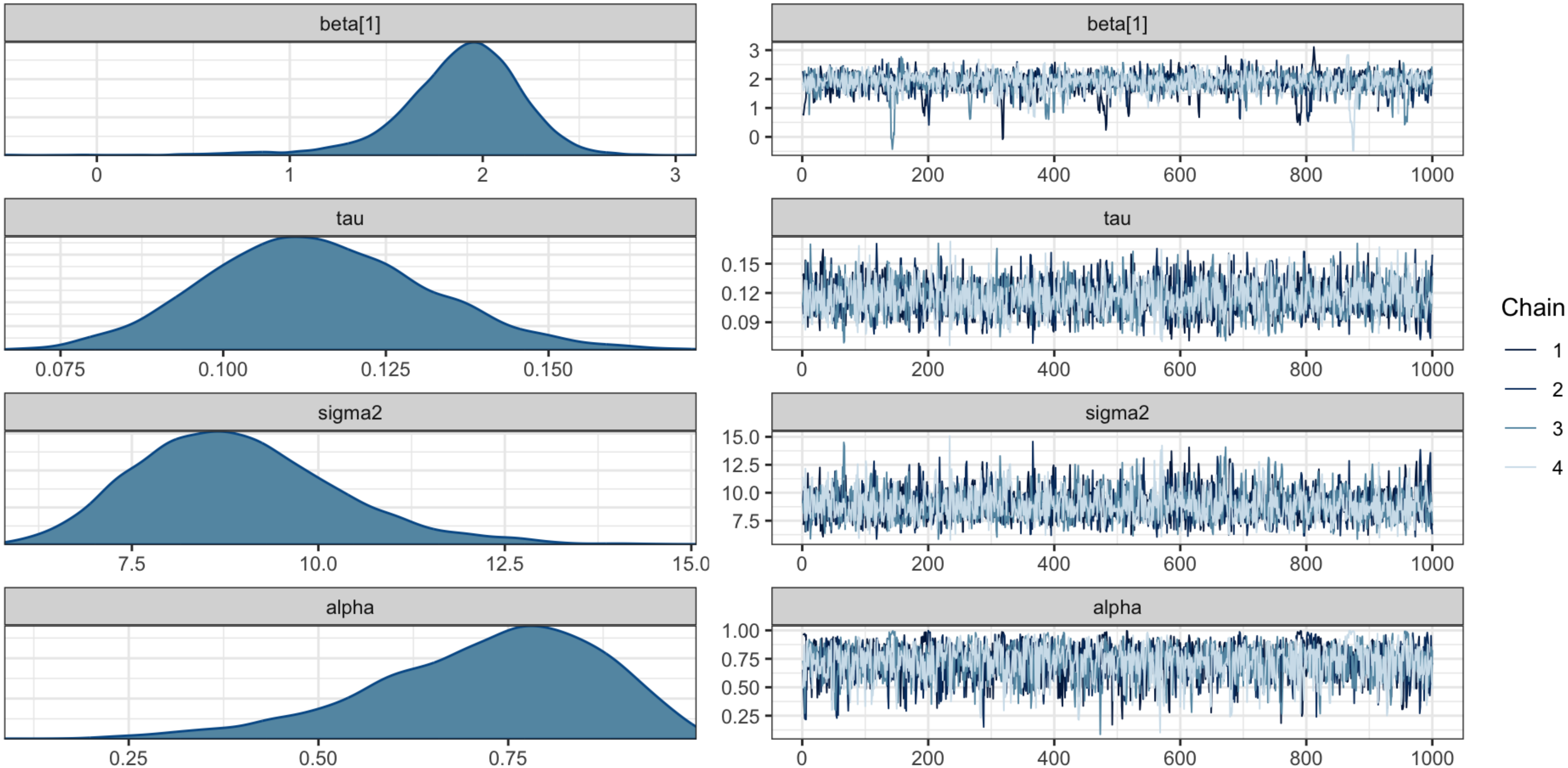
```
X = model.matrix(~1, data = nc)
d = list(
  n = nrow(X),                  # number of obs
  p = ncol(X),                  # number of coe
  X = X,                        # design matrix
  y = 1000*nc$SID74/nc$BIR74,
  W = A*1,
  D = diag(rowSums(A)),
  D_inv = diag(1/rowSums(A))
)

s_car = rstan::sampling(stan_car, data = d, c
```

Results

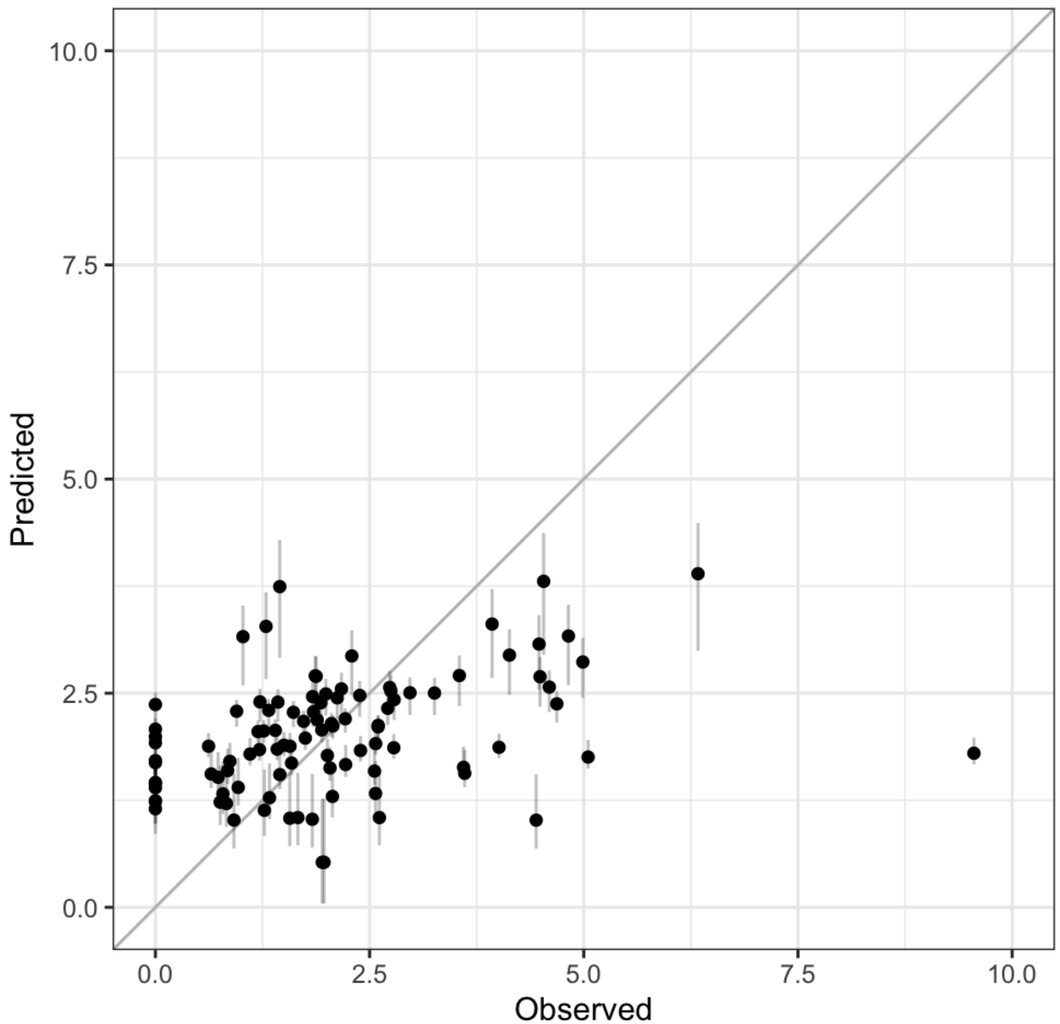
```r
rstan::extract(s_car, pars=c("beta[1]","tau", "sigma2", "alpha")) |>
  posterior::summarise_draws()
```

```
# A tibble: 4 × 10
  variable  mean median     sd    mad     q5    q95  rhat ess_bulk ess_tail
  <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl>
1 beta[1]  1.88   1.92  0.338  0.273  1.32    2.34   1.00    3872.    3961.
2 tau      0.115  0.114 0.0168 0.0168 0.0884  0.143  1.00    4032.    3692.
3 sigma2   8.92   8.80  1.34   1.29   6.98   11.3    1.00    4032.    3692.
4 alpha    0.723  0.744 0.150  0.150  0.439   0.932  1.00    3922.    3960.
```

Diagnostics



Observed vs predicted



What's different?

brms model

$$y \sim N(X\beta + w, \sigma^2) w \sim N(0, \sigma_{CAR}^2 (D - \alpha W)^{-1})$$

Stan model

$$y \sim N(X\beta + w, \sigma^2 (D - \alpha W)^{-1})$$

SAR with brms

brms SAR (error)

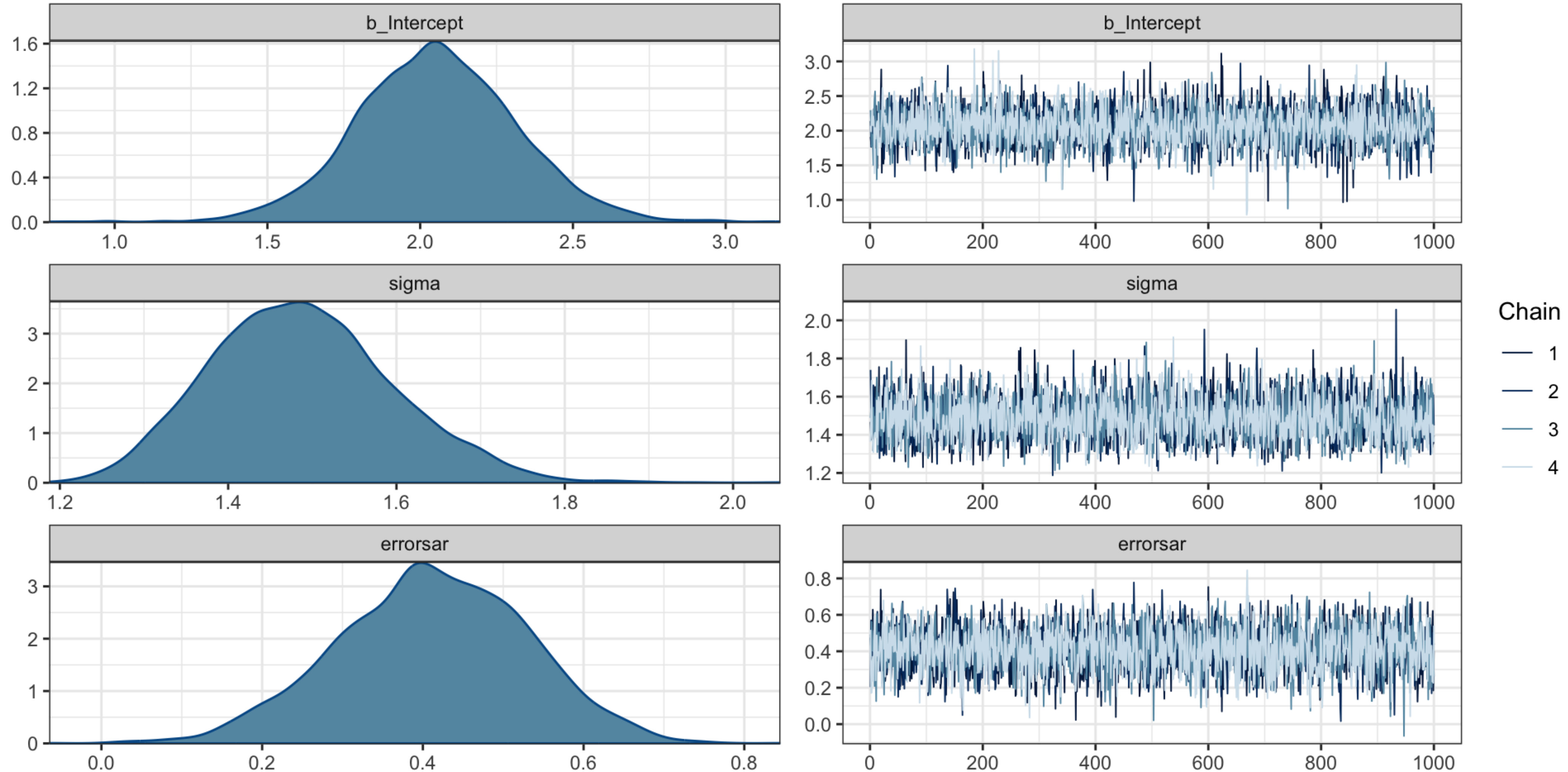
```r
W = diag(1/rowSums(A)) %*% A

b_sar_err = brms::brm(
    1000*SID74/BIR74 ~ 1 + sar(W, type="error"),
    data=nc, data2=list(W=W),
    #silent=2, refresh=0,
    iter=4000,
    cores = 4,
    thin = 2
)
```

```r
b_sar_err
```

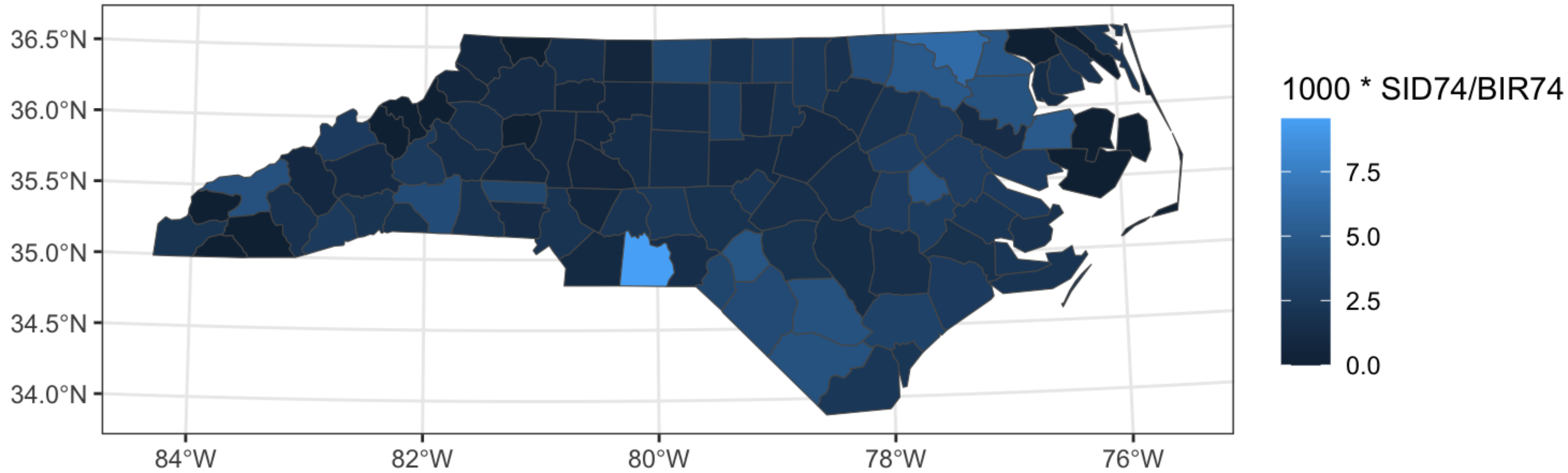
```
 Family: gaussian 
  Links: mu = identity; sigma = identity 
Formula: 1000 * SID74/BIR74 ~ 1 + sar(W, type = "error") 
   Data: nc (Number of observations: 100) 
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 2;
         total post-warmup draws = 4000

Correlation Structures:
         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
errorsar     0.41      0.12     0.18     0.63 1.00     3381     3378

Population-Level Effects: 
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     2.05      0.26     1.54     2.58 1.00     3659     3041

Family Specific Parameters: 
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.49      0.11     1.30     1.71 1.00     3814     3384

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

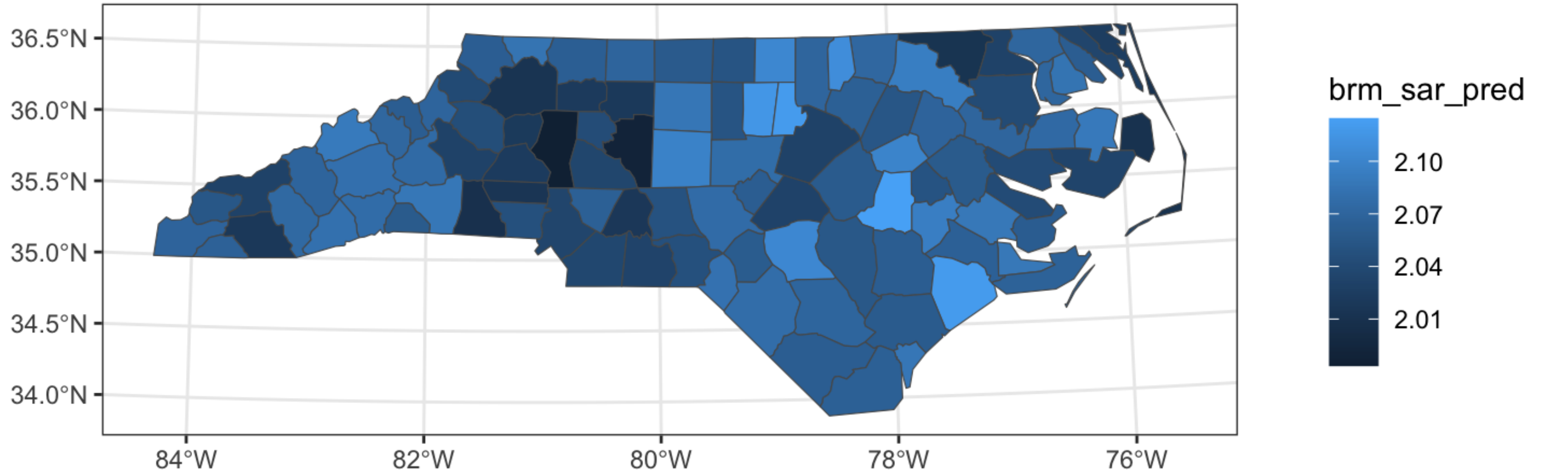
Diagnostics

```r
plot(b_sar_err)
```

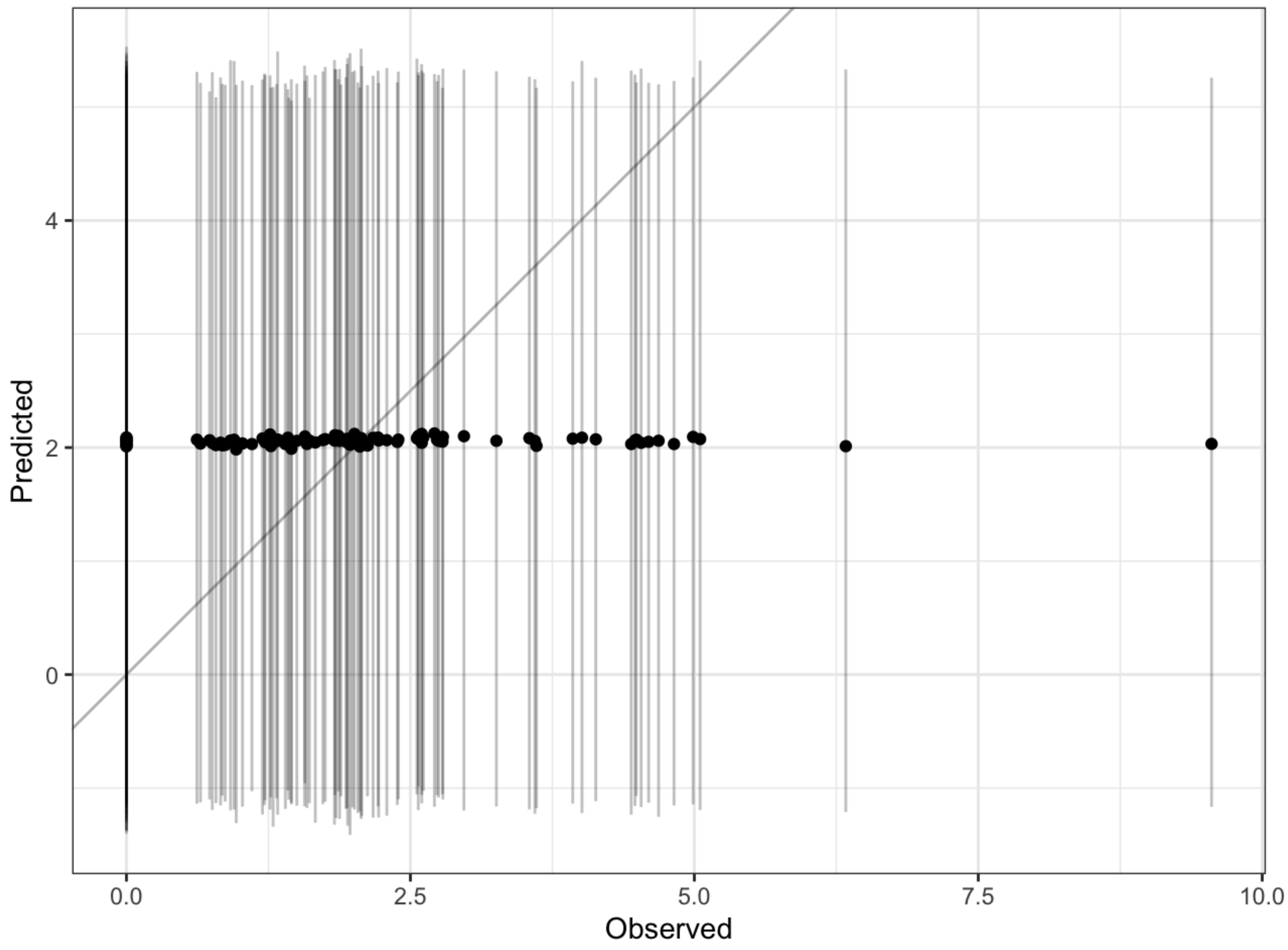


Predictions





Observed vs predicted



Correcting `predict()`

If instead we use \boldsybol$X\boldsymbol{\beta} + \phi \boldsymbol{W}\boldsymbol{y}$, we get the following:

```
p = b_sar_err |>
  tidybayes::spread_draws(b_Intercept, errorsar) |>
  filter(.chain == 1) |>
  mutate(
    y_cond = map2(
      b_Intercept, errorsar,
      ~ .x + .y * W %*% (1000 * (nc$SID74 / nc$BIR74))
    )
  ) |>
  pull(y_cond) |>
  do.call(cbind, args = _)

tibble(
    y_cond_mean = apply(p, 1, mean),
    y_cond_q025 = apply(p, 1, quantile, 0.025),
    y_cond_q975 = apply(p, 1, quantile, 0.975)
```

brms SAR (lag)

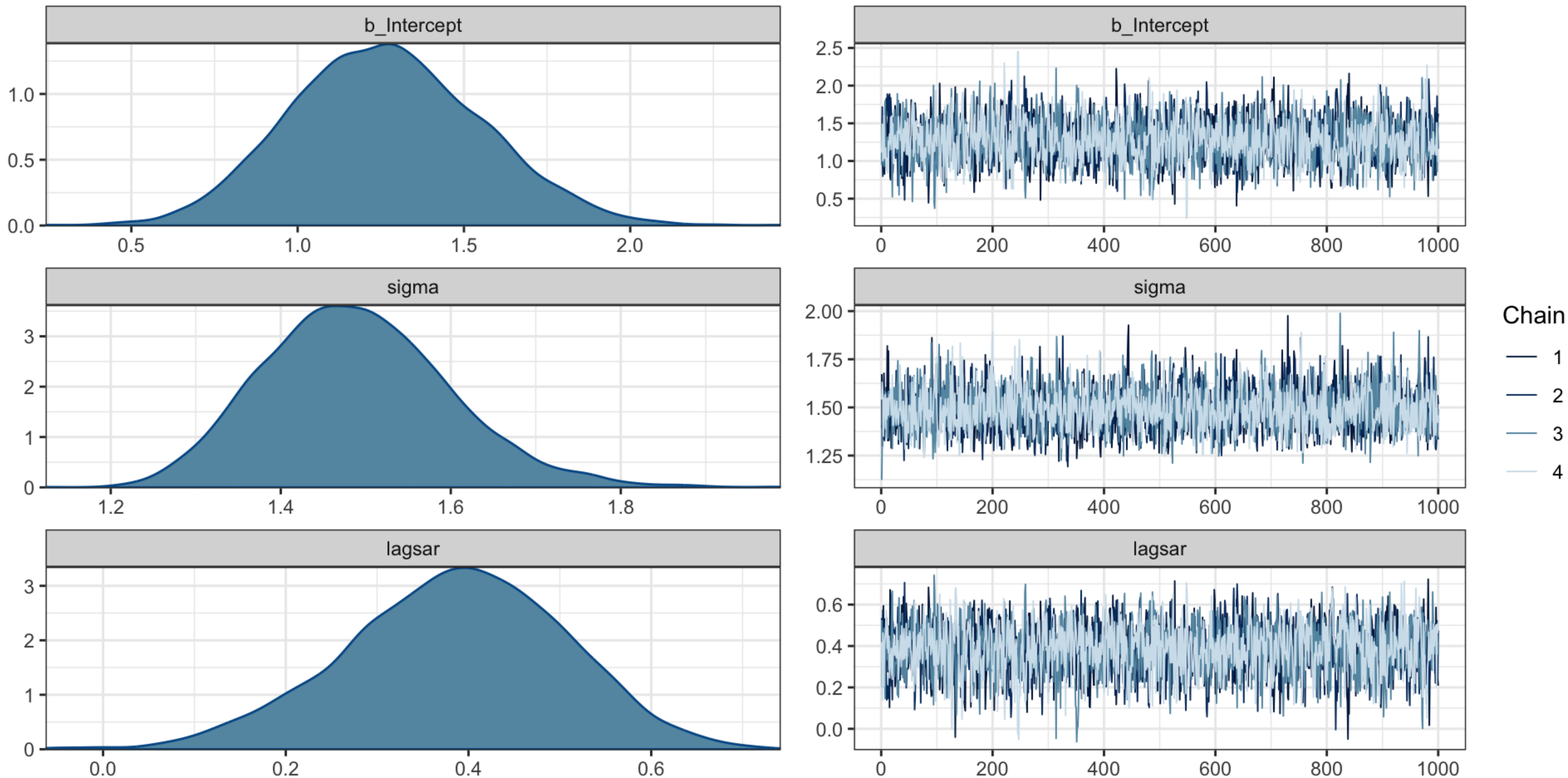
```
W = diag(1/rowSums(A)) %*% A

b_sar_lag = brms::brm(
    1000*SID74/BIR74 ~ 1 + sar(W, type="lag"),
    data=nc, data2=list(W=W),
    iter=4000,
    cores = 4,
    thin = 2
)
```

```r
b_sar_lag
```

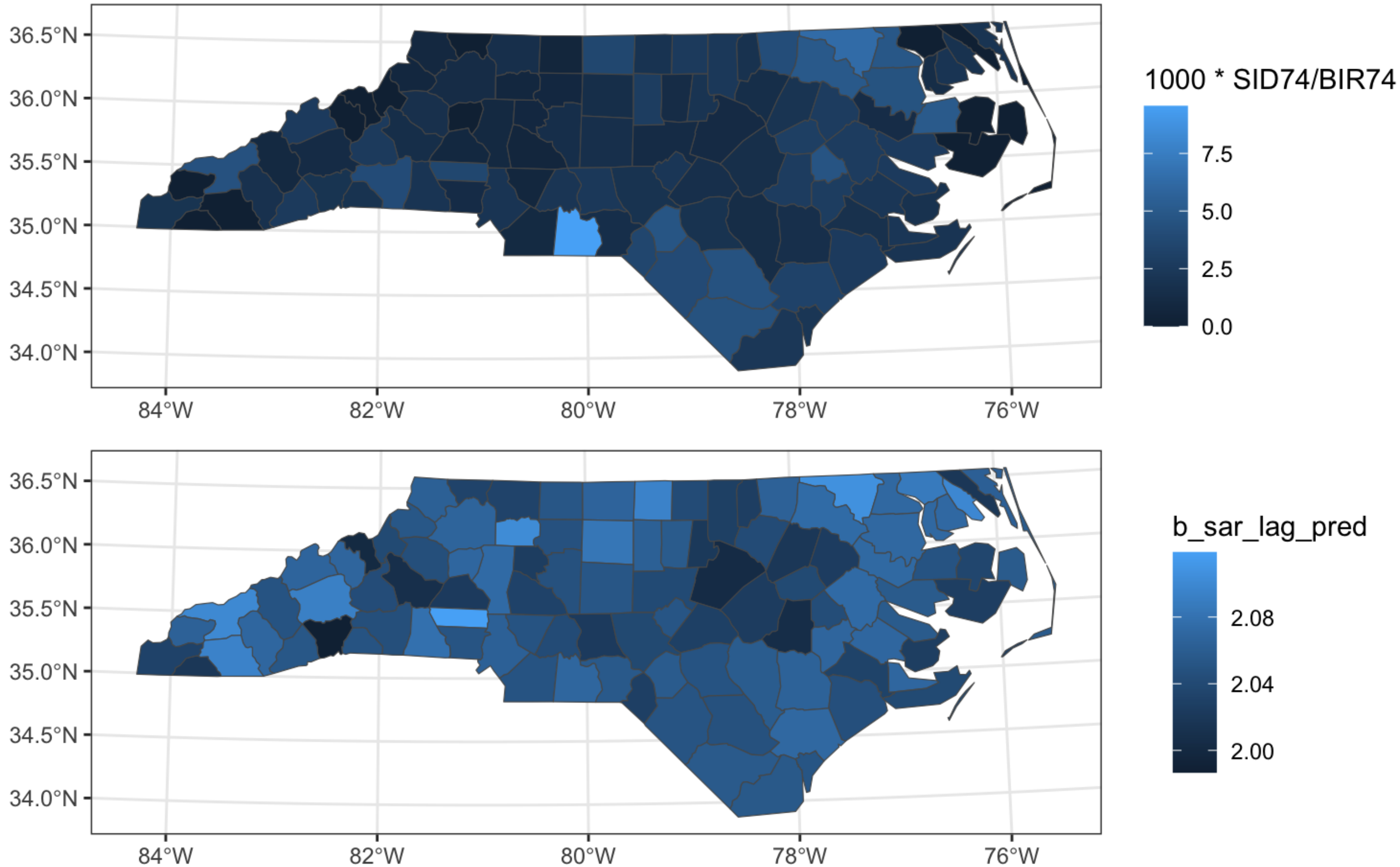
```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: 1000 * SID74/BIR74 ~ 1 + sar(W, type = "lag")
   Data: nc (Number of observations: 100)
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 2;
         total post-warmup draws = 4000

Correlation Structures:
       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
lagsar     0.39      0.12     0.14     0.61 1.00     2561     2779

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     1.27      0.29     0.74     1.84 1.00     2489     2675

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.49      0.11     1.29     1.73 1.00     3224     3138

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Diagnostics

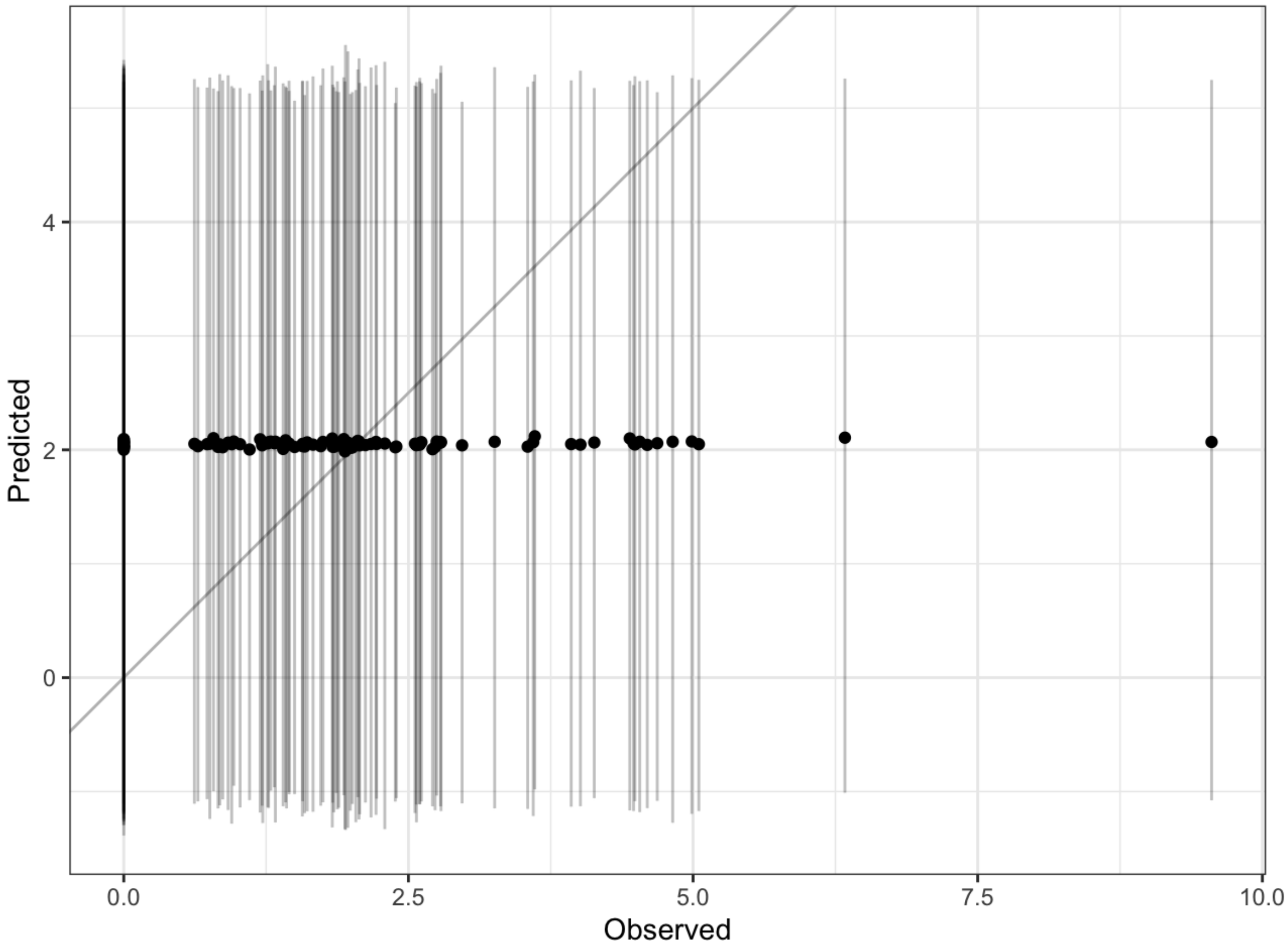
```
plot(b_sar_lag)
```



Predictions



Observed vs predicted



Correcting `predict()`

If instead we use \boldsybol$X\boldsymbol{\beta} + \phi\boldsymbol{W}\boldsymbol{y}$, we get the following:

```r
p = b_sar_lag |>
  tidybayes::spread_draws(b_Intercept, lagsar) |>
  filter(.chain == 1) |>
  mutate(
    y_cond = map2(
      b_Intercept, lagsar,
      ~ .x + .y * W %*% (1000 * (nc$SID74 / nc$BIR74))
    )
  ) |>
  pull(y_cond) |>
  do.call(cbind, args = _)

tibble(
    y_cond_mean = apply(p, 1, mean),
    y_cond_q025 = apply(p, 1, quantile, 0.025),
    y_cond_q975 = apply(p, 1, quantile, 0.975)
```