Lecture 19

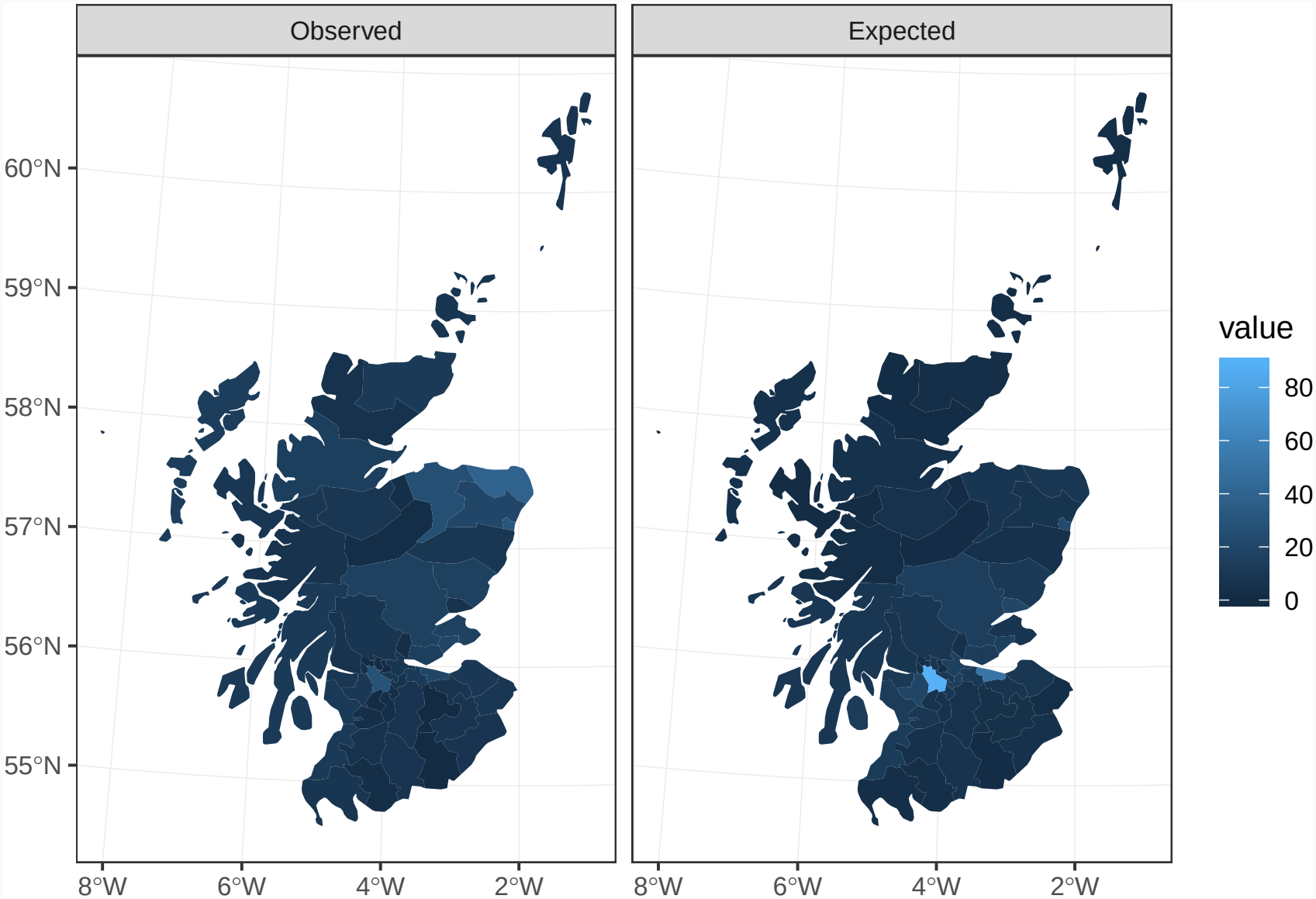
Spatial GLM + Point Reference Spatial Data

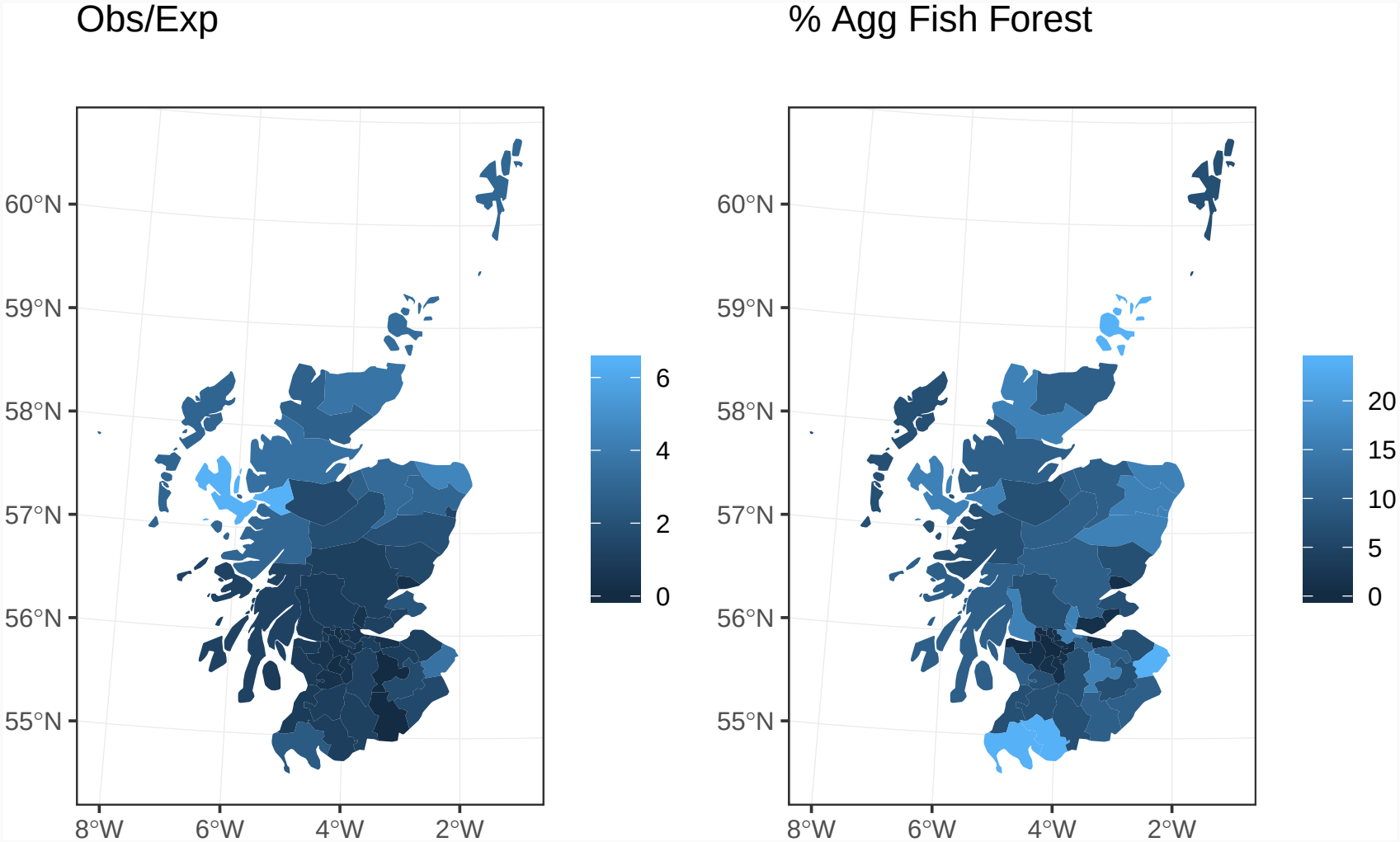
Colin Rundel

11/09/2017

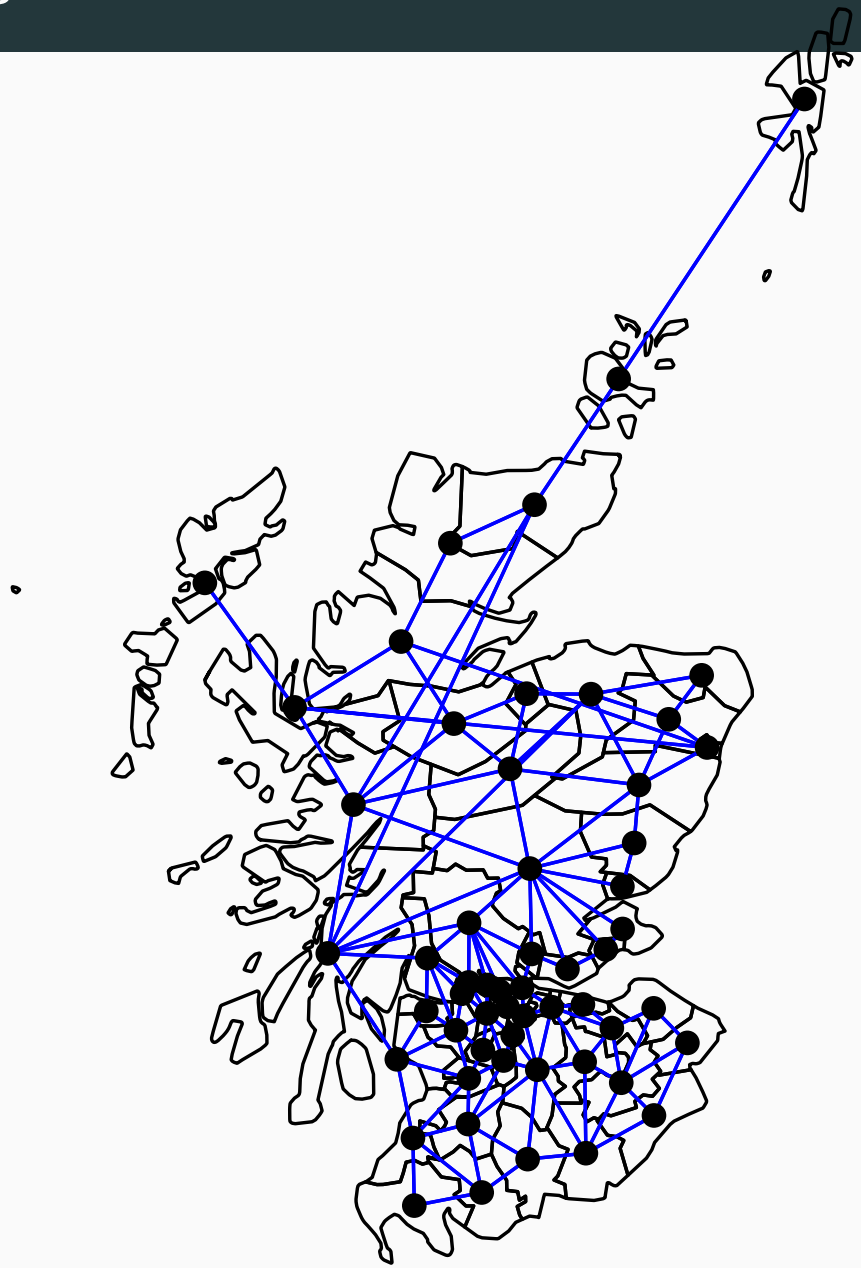
Spatial GLM Models

Scottish Lip Cancer Data




Obs/Exp
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
6
4
2
0
% Agg Fish Forest
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
20
15
10
5
0

Neighborhood / weight matrix



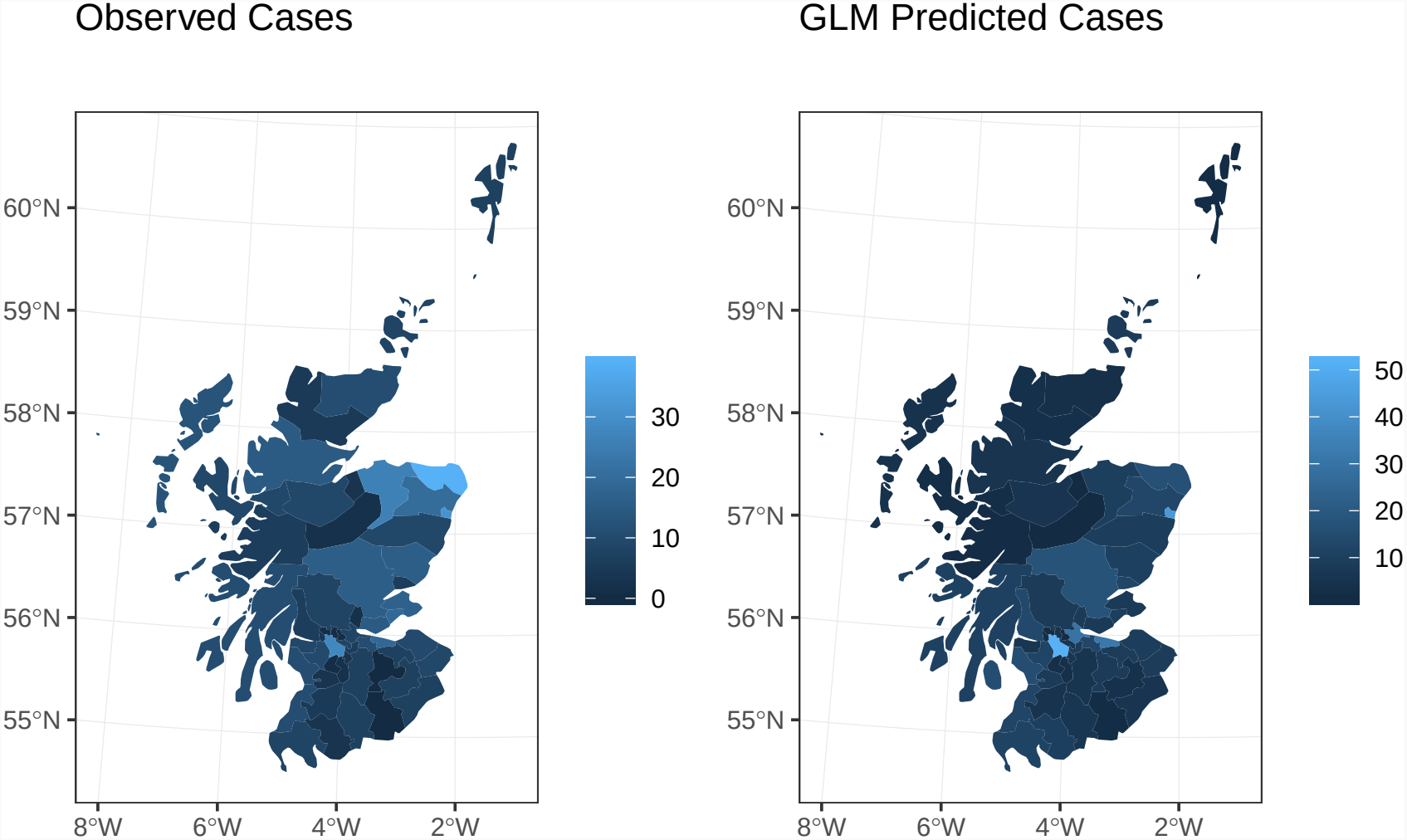
Moran's I

```
spdep::moran.test(lip_cancer$Observed, listw)
##
##  Moran I test under randomisation
##
## data:  lip_cancer$Observed
## weights: listw
##
## Moran I statistic standard deviate = 4.5416, p-value = 2.792e-06
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.311975396      -0.018181818       0.005284831

spdep::moran.test(lip_cancer$Observed / lip_cancer$Expected, listw)
##
##  Moran I test under randomisation
##
## data:  lip_cancer$Observed/lip_cancer$Expected
## weights: listw
##
## Moran I statistic standard deviate = 8.2916, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.589795225      -0.018181818       0.005376506
```

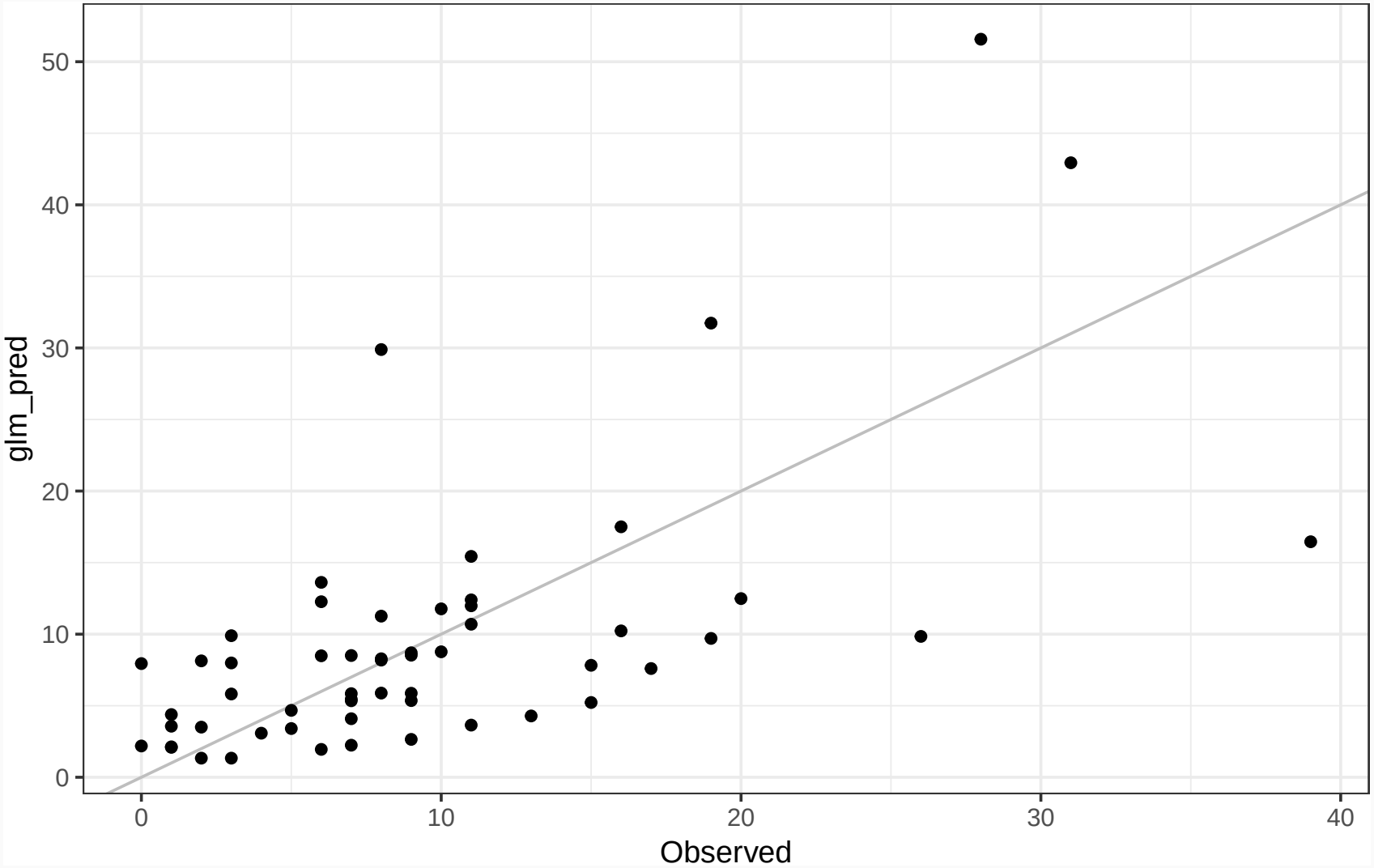
GLM

```
l = glm(Observed ~ offset(log(Expected)) + pcaff,
        family="poisson", data=lip_cancer)

summary(l)
##
## Call:
## glm(formula = Observed ~ offset(log(Expected)) + pcaff, family = "poisson",
##     data = lip_cancer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7632  -1.2156   0.0967   1.3362   4.7130
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.542268   0.069525   -7.80 6.21e-15 ***
## pcaff        0.073732   0.005956   12.38  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 380.73  on 55  degrees of freedom
## Residual deviance: 238.62  on 54  degrees of freedom
## AIC: 450.6
##
## Number of Fisher Scoring iterations: 5
```

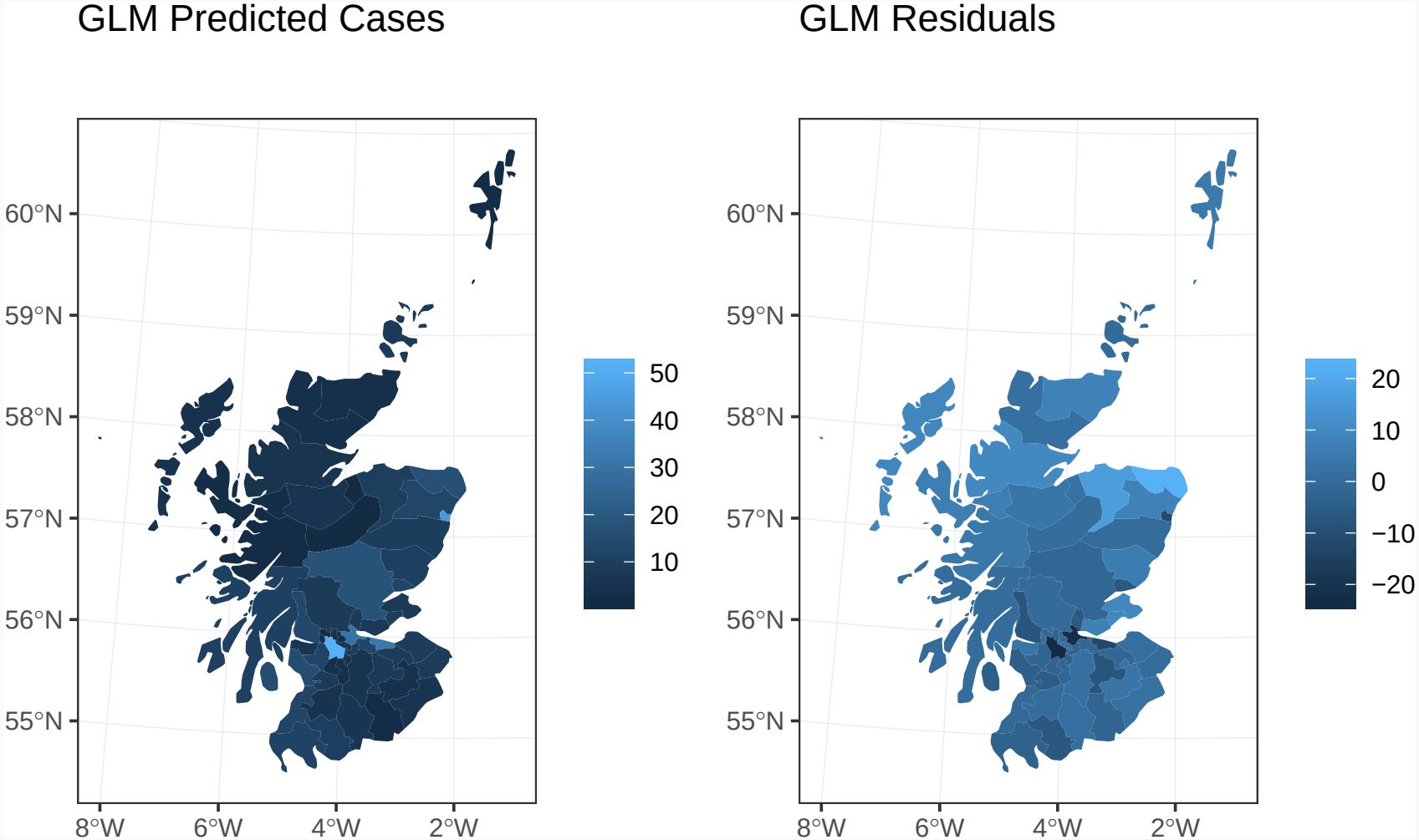
GLM Fit


Observed Cases
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
30
20
10
0
GLM Predicted Cases
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
50
40
30
20
10

GLM Fit



GLM Residuals



Model Results

```
#RMSE
lip_cancer$glm_resid %>% .^2 %>% mean() %>% sqrt()
## [1] 7.480889

#Moran's I
spdep::moran.test(lip_cancer$glm_resid, listw)
##
##  Moran I test under randomisation
##
## data:  lip_cancer$glm_resid
## weights: listw
##
## Moran I statistic standard deviate = 4.8186, p-value = 7.228e-07
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.333403223      -0.018181818       0.005323717
```

A hierachical model for lip cancer

We have observed counts of lip cancer for 56 districts in Scotland. Let y_i represent the number of lip cancer for district i.

$$y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \log(E_i) + x_i\beta + \omega_i$$

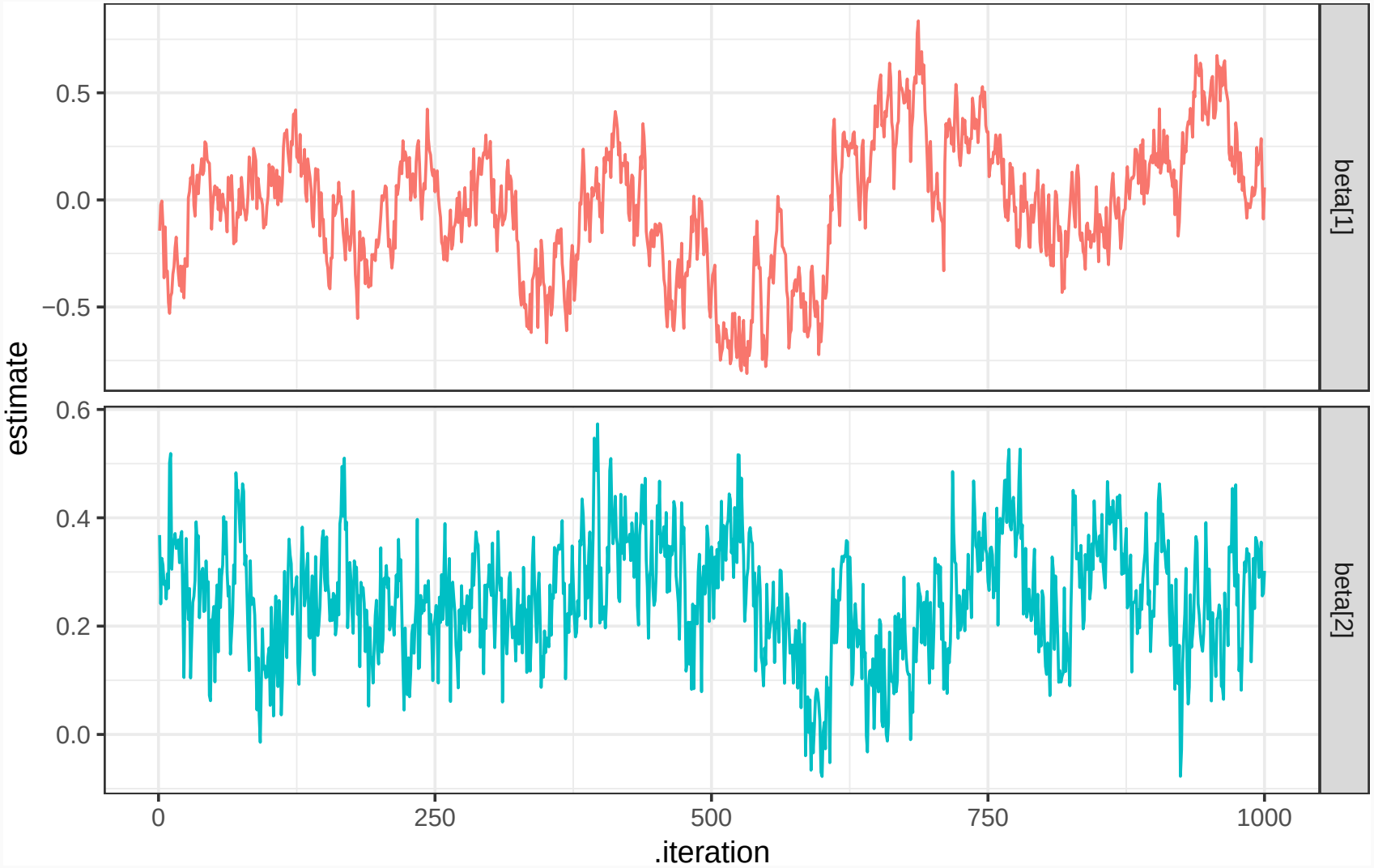
$$\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0},\ \sigma^2(\mathbf{D} - \phi\,\mathbf{W})^{-1})$$

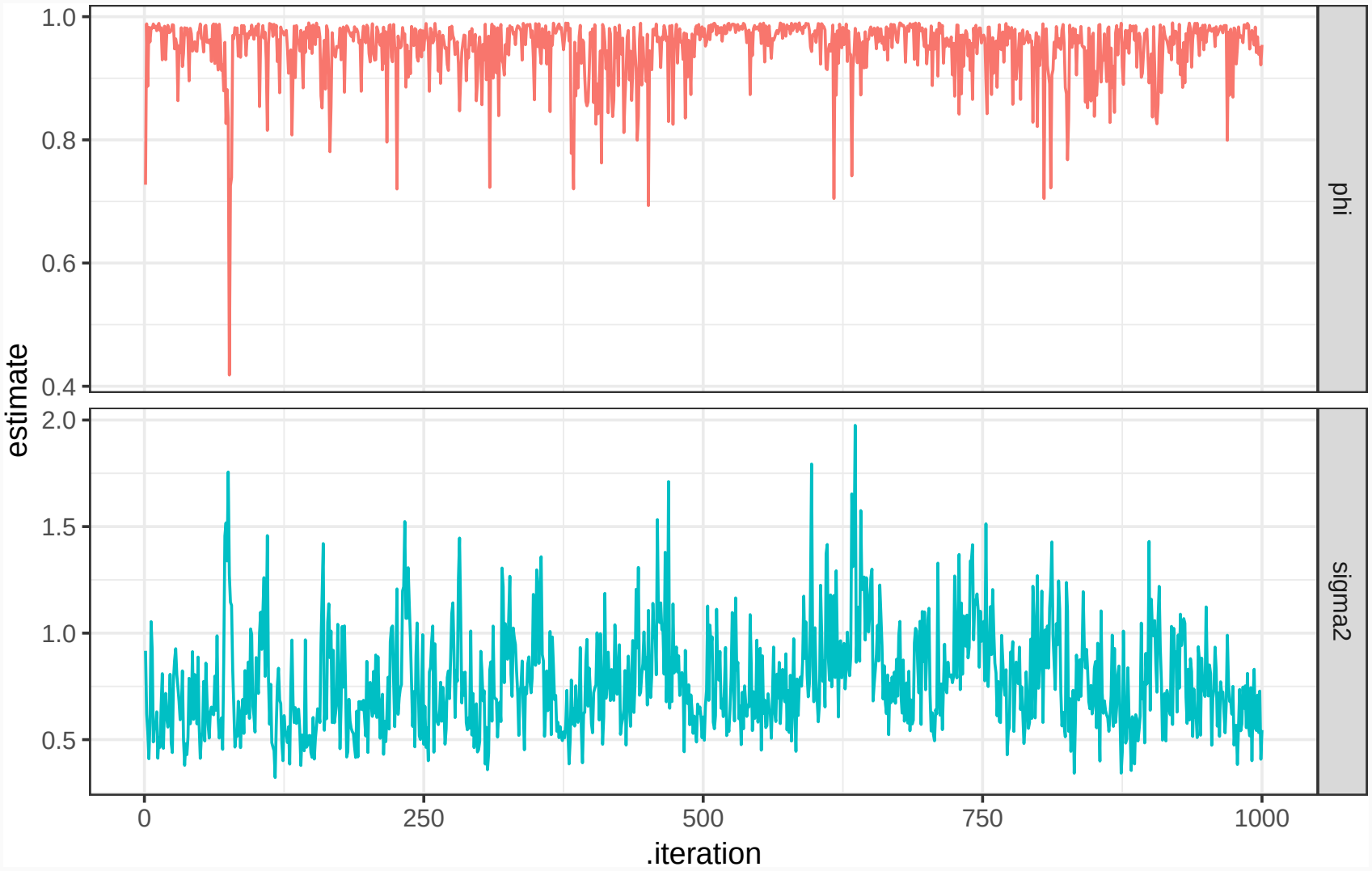
where E_i is the expected counts for each region (and serves as an offet).

Data prep & CAR model

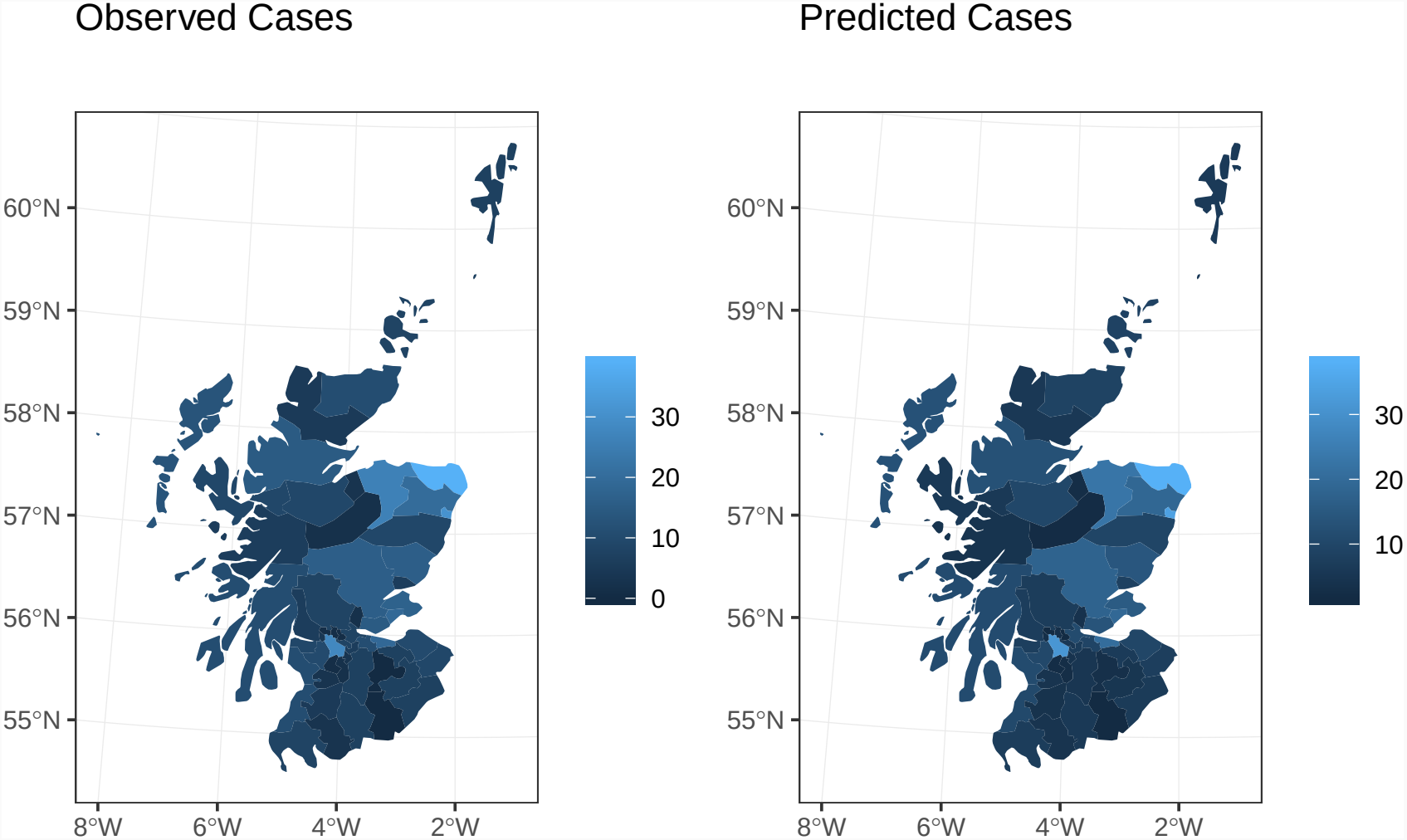
```
D = diag(rowSums(W))
X = model.matrix(~scale(lip_cancer$pcaff))
log_offset = log(lip_cancer$Expected)
y = lip_cancer$Observed


car_model = "model{
  for(i in 1:length(y)) {
    y[i] ~ dpois(lambda[i])
    y_pred[i] ~ dpois(lambda[i])
    log(lambda[i]) = log_offset[i] + X[i,] %*% beta + omega[i]
  }

  for(i in 1:2) {
    beta[i] ~ dnorm(0,1)
  }

  omega ~ dmnorm(rep(0,length(y)), tau * (D - phi*W))
  sigma2 = 1/tau
  tau ~ dgamma(2, 2)
  phi ~ dunif(0,0.99)
}"
```

CAR Results

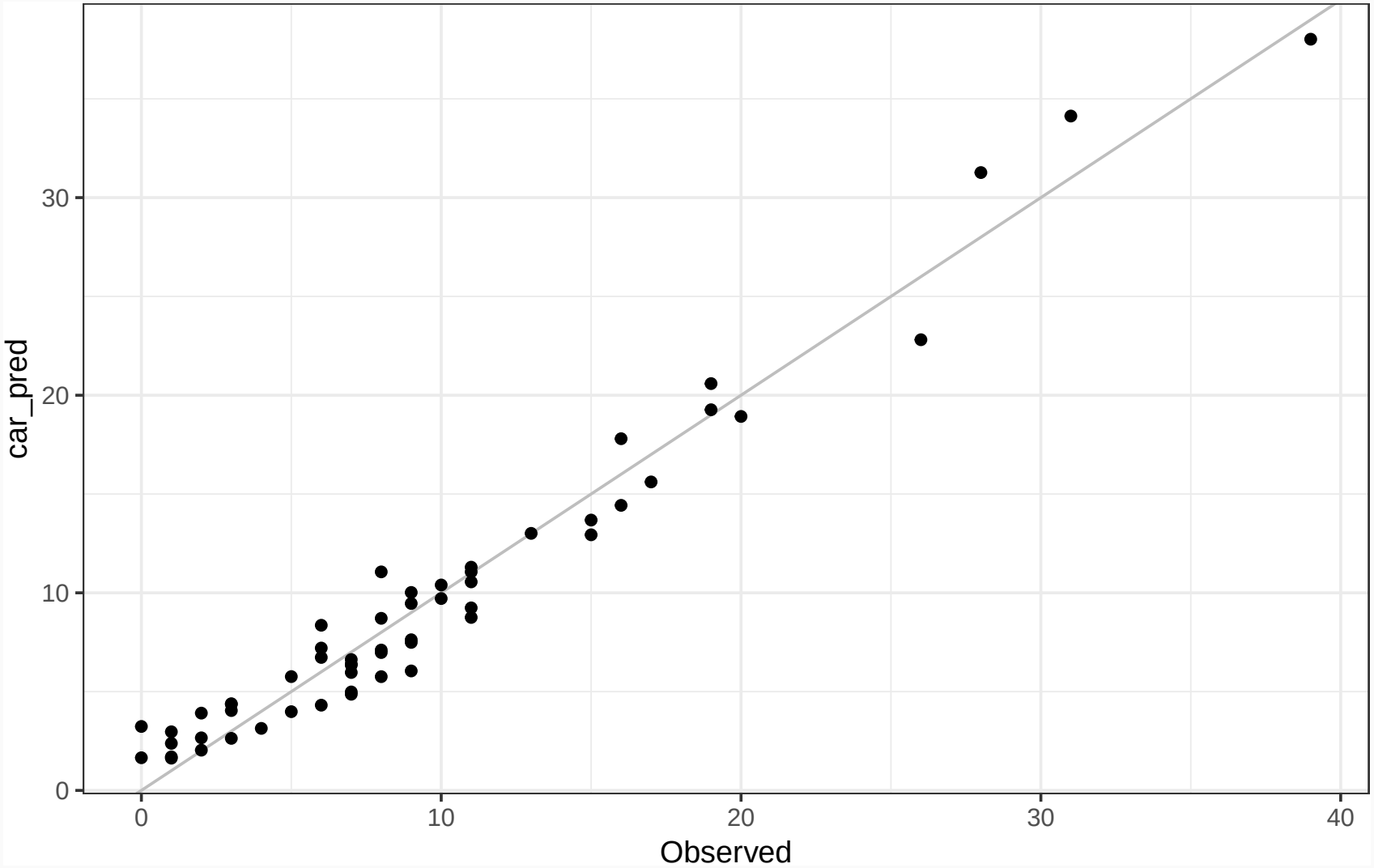

estimate
0.5
0.0
−0.5
0.6
0.4
0.2
0.0
0
250
500
750
1000
.iteration
beta[1]
beta[2]


estimate
1.0
0.8
0.6
0.4
phi
2.0
1.5
1.0
0.5
sigma2
0
250
500
750
1000
.iteration

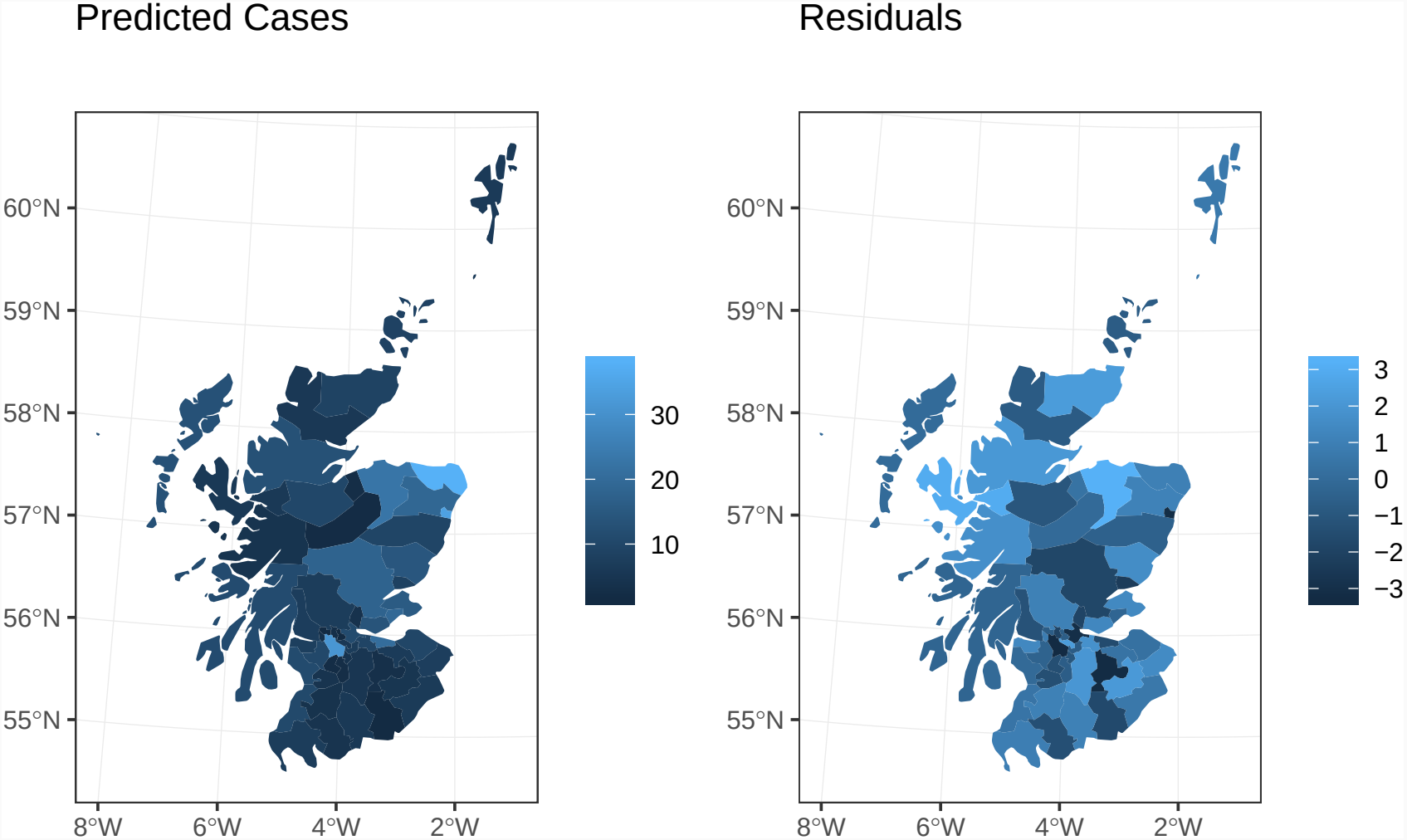
CAR Predictions



CAR Predictions



CAR Residuals



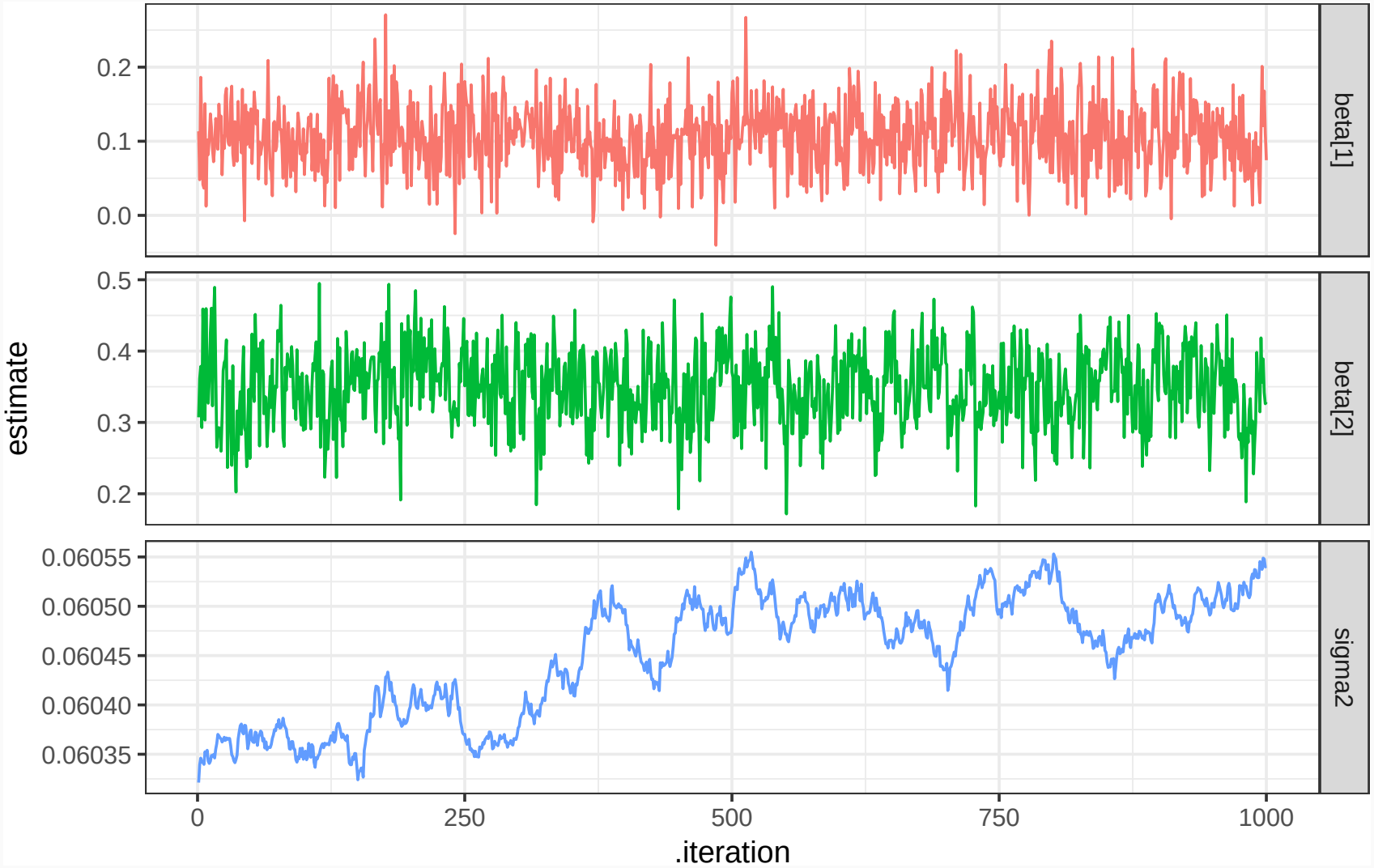
CAR Results

```
#RMSE
lip_cancer$car_resid %>% .^2 %>% mean() %>% sqrt()
## [1] 1.586241

#Moran's I
spdep::moran.test(lip_cancer$car_resid, listw)
##
##  Moran I test under randomisation
##
## data:  lip_cancer$car_resid
## weights: listw
##
## Moran I statistic standard deviate = 1.0633, p-value = 0.1438
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.061804482      -0.018181818       0.005658742
```

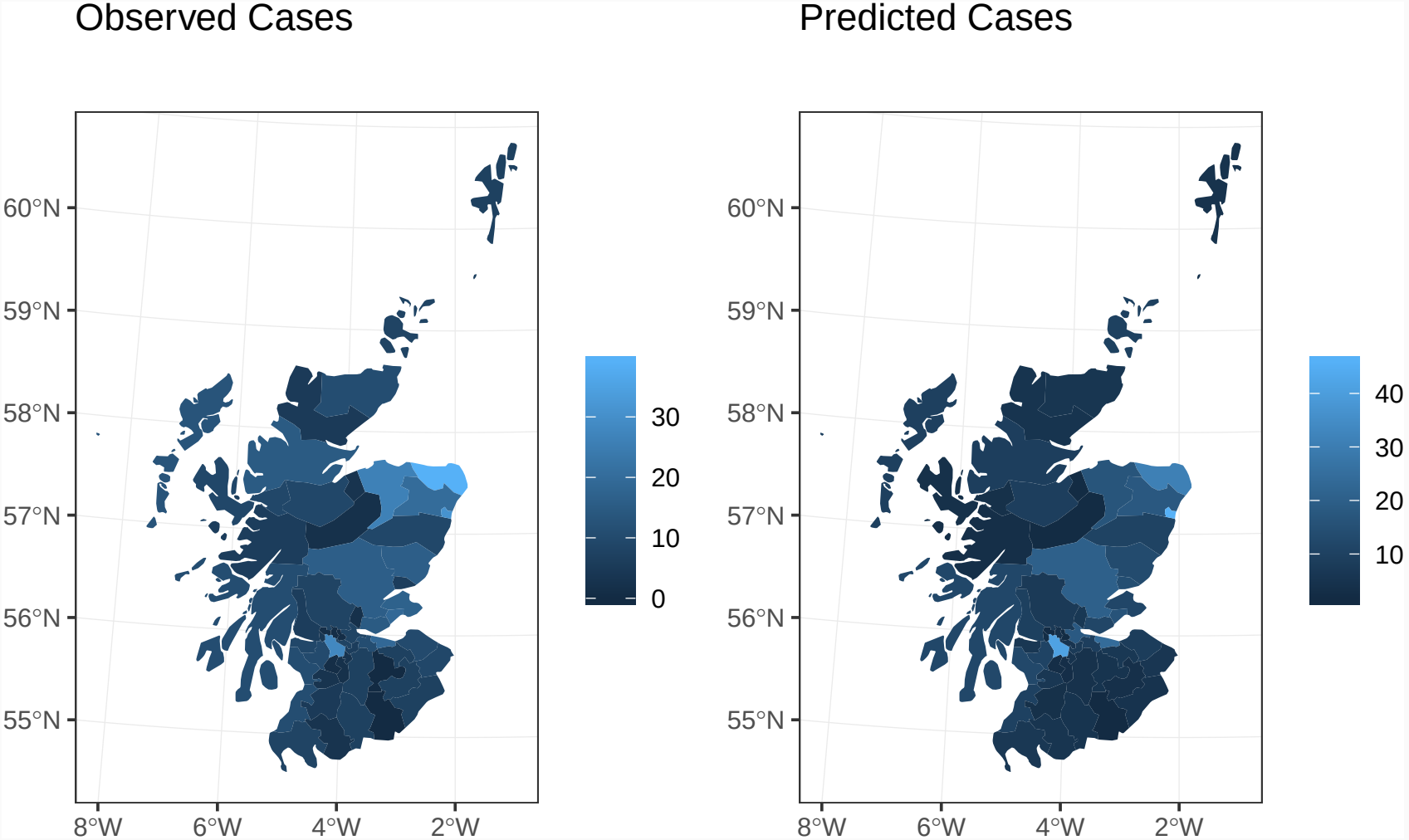
Intrinsic Autoregressive Model

```
iar_model = "model{
  for(i in 1:length(y)) {
    y[i] ~ dpois(lambda[i])
    y_pred[i] ~ dpois(lambda[i])
    log(lambda[i]) = log_offset[i] + X[i,] %*% beta + omega[i]
  }

  for(i in 1:2) {
    beta[i] ~ dnorm(0,1)
  }

  omega_free ~ dmnorm(rep(0,length(y)), tau * (D - W))
  omega = omega_free - mean(omega_free)
  sigma2 = 1/tau
  tau ~ dgamma(2, 2)
}"
```

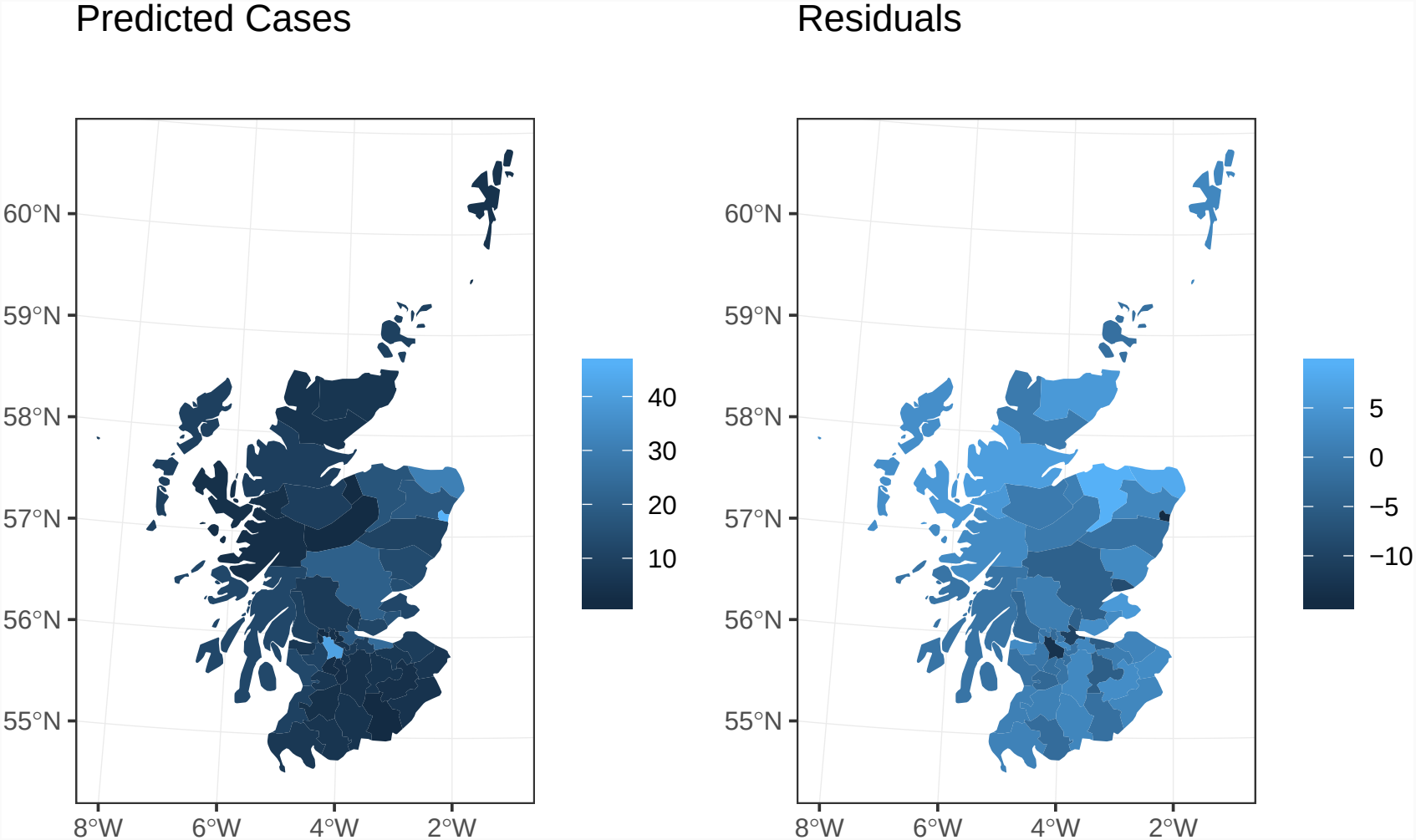
Model Parameters



Predictions



Residuals


Predicted Cases
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
40
30
20
10
Residuals
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
5
0
−5
−10

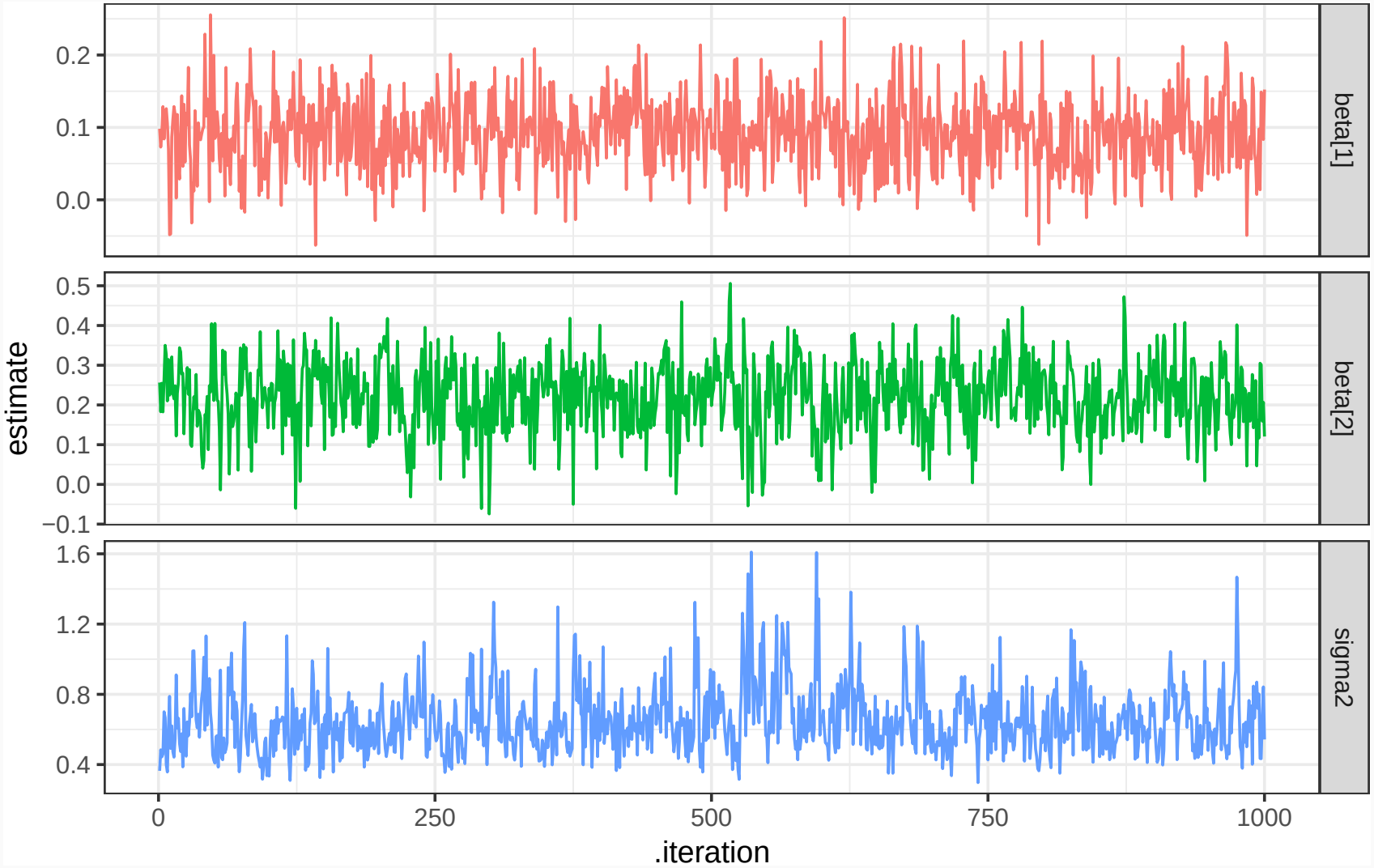
IAR Results

```
#RMSE
lip_cancer$iar_resid %>% .^2 %>% mean() %>% sqrt()
## [1] 4.473069

#Moran's I
spdep::moran.test(lip_cancer$iar_resid, listw)
##
##  Moran I test under randomisation
##
## data:  lip_cancer$iar_resid
## weights: listw
##
## Moran I statistic standard deviate = 2.6377, p-value = 0.004174
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.175216401      -0.018181818       0.005375965
```

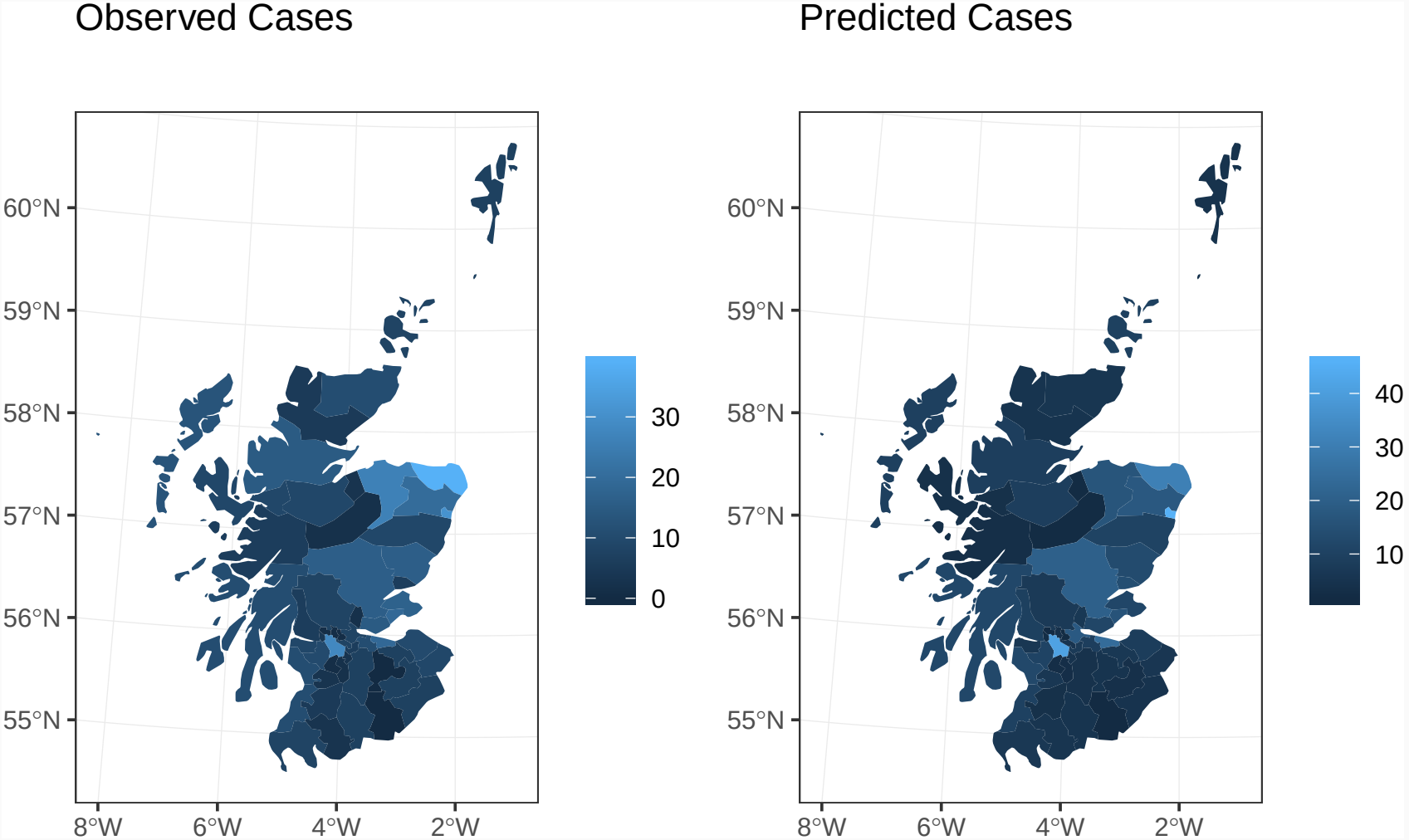
Intrinsic Autoregressive Model - Reparameterized

```
iar_model2 = "model{
  for(i in 1:length(y)) {
    y[i] ~ dpois(lambda[i])
    y_pred[i] ~ dpois(lambda[i])
    log(lambda[i]) = log_offset[i] + X[i,] %*% beta + sigma * omega[i]
  }

  for(i in 1:2) {
    beta[i] ~ dnorm(0,1)
  }

  omega_free ~ dmnorm(rep(0,length(y)), (D - W))
  omega = omega_free - mean(omega_free)
  sigma2 = 1/tau
  sigma = sqrt(sigma2)
  tau ~ dgamma(2, 2)
}"
```

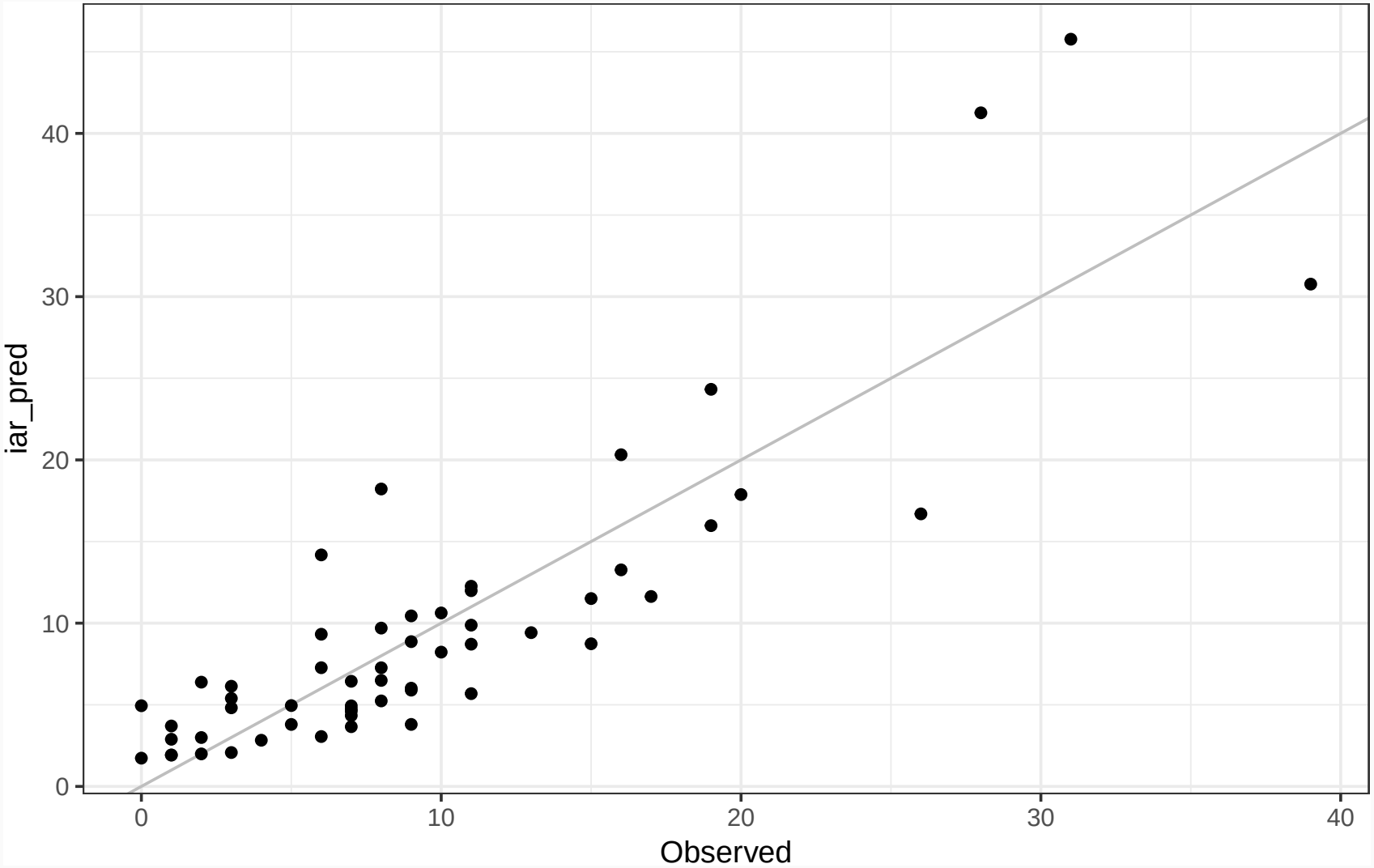
IAR(2) Parameters



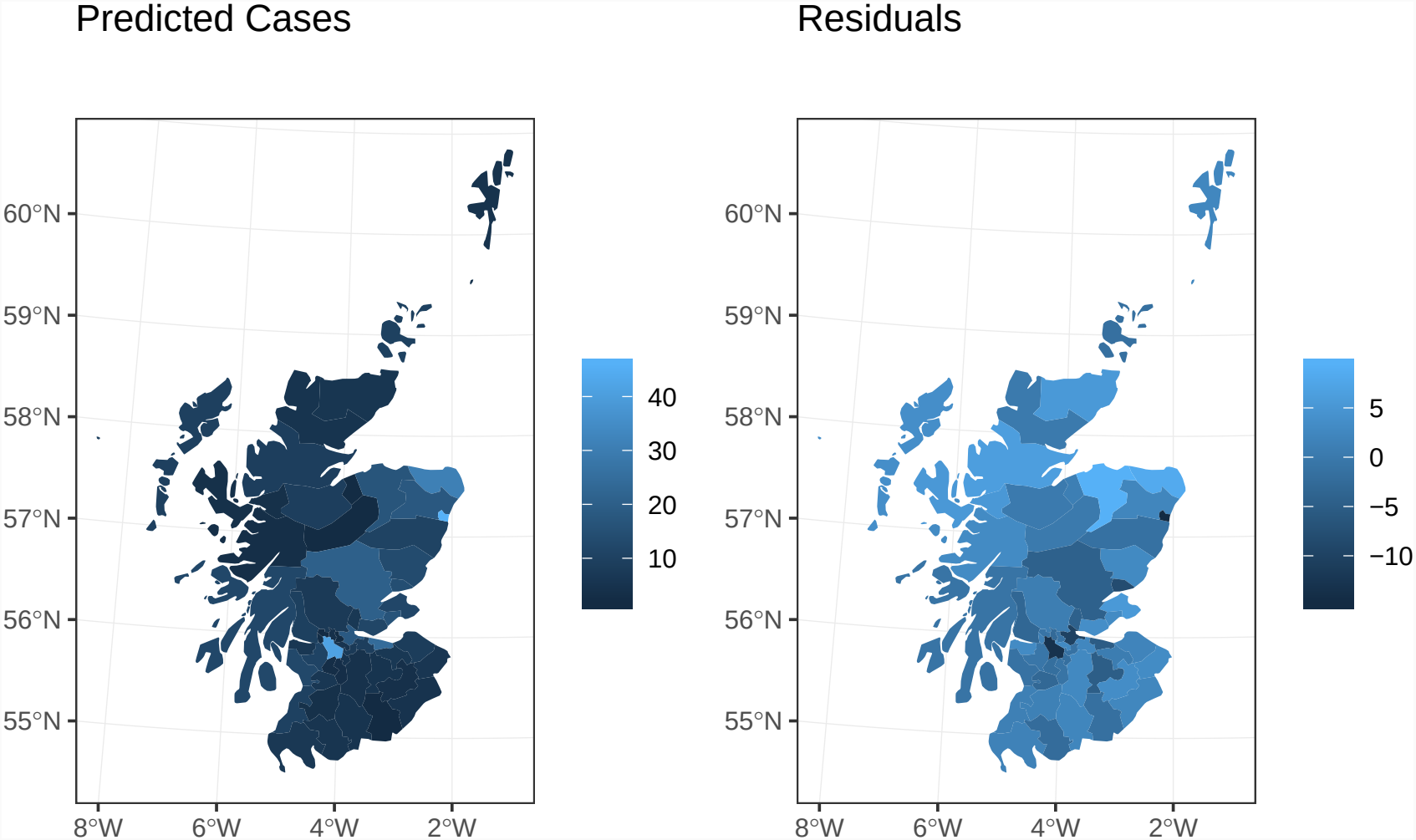
Predictions



Predictions (cont.)



Residuals


Predicted Cases
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
40
30
20
10
Residuals
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
5
0
−5
−10

IAR(2) Results

```
#RMSE
lip_cancer$iar2_resid %>% .^2 %>% mean() %>% sqrt()
## [1] 1.654235

#Moran's I
spdep::moran.test(lip_cancer$iar2_resid, listw)
##
##  Moran I test under randomisation
##
## data:  lip_cancer$iar2_resid
## weights: listw
##
## Moran I statistic standard deviate = 0.39186, p-value = 0.3476
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.011188088      -0.018181818       0.005617437
```

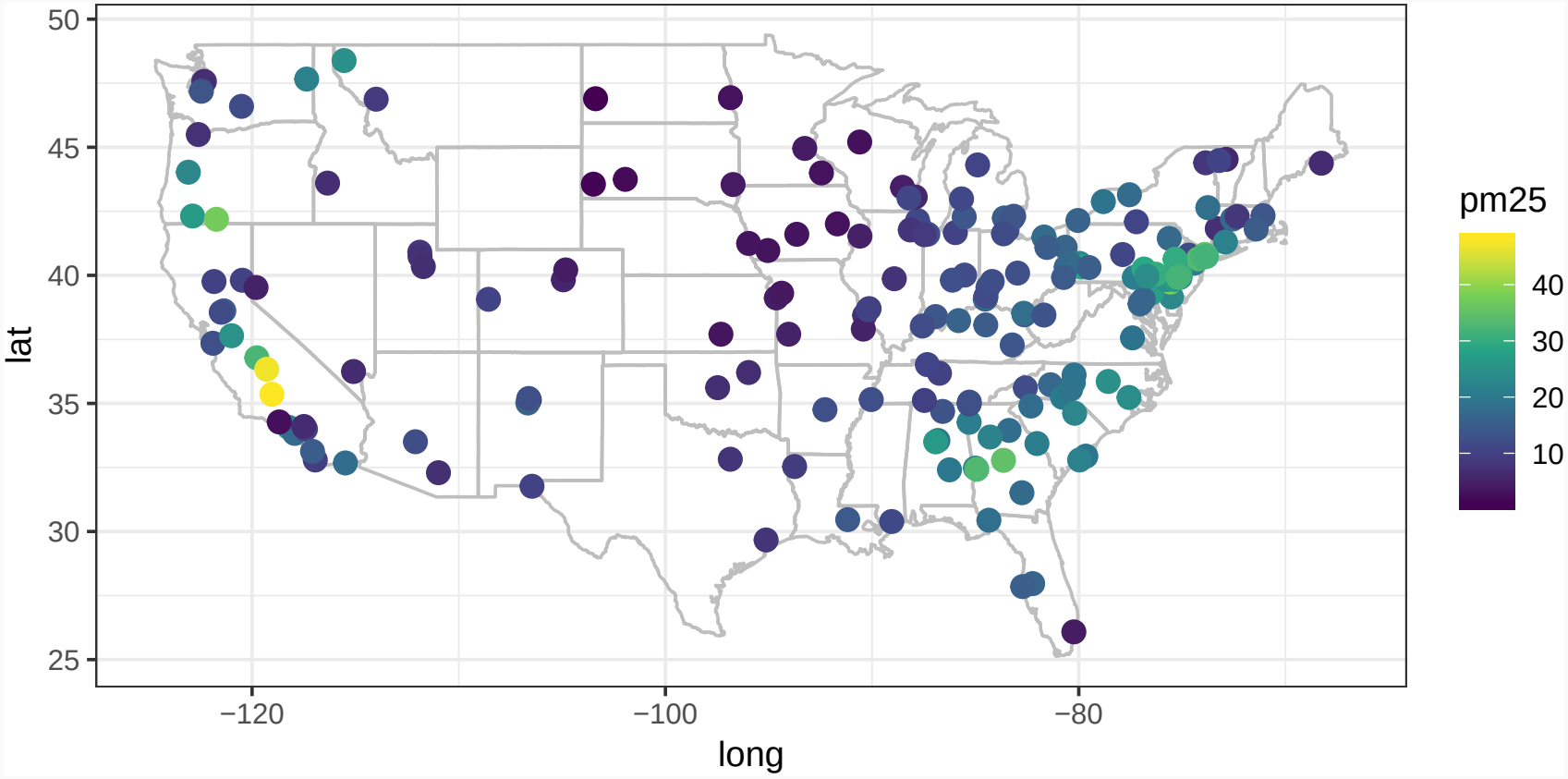
Overall Results

model	rmse	moran
glm	7.4809	0.3334
car	1.5862	0.0618
iar	4.4731	0.1752
iar2	1.6542	0.0112

Point Referenced Data

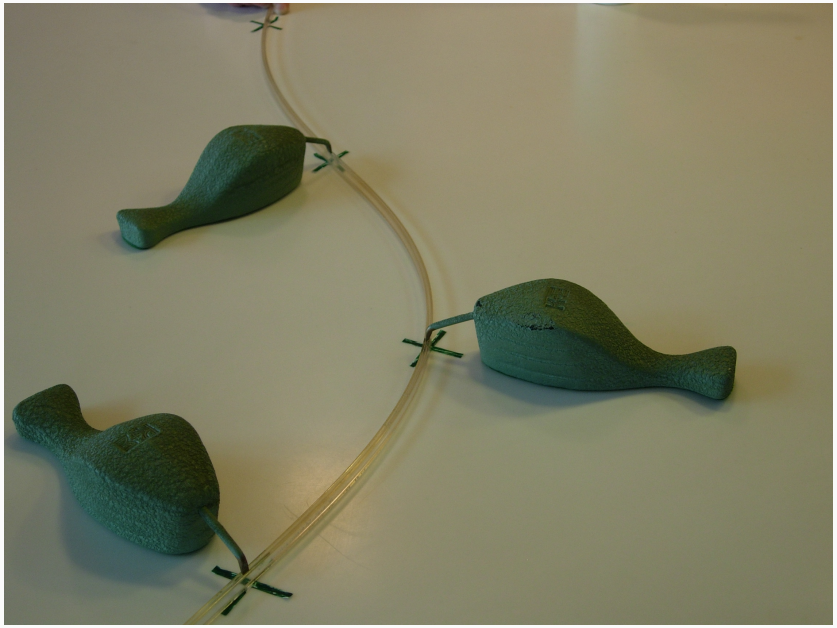
Example - PM2.5 from CSN

The Chemical Speciation Network are a series of air quality monitors run by EPA (221 locations in 2007). We'll look at a subset of the data from Nov 11th, 2007 (n=191) for just PM2.5.



```
csn
## # A tibble: 191 x 5
##        site longitude latitude date                pm25
##       <int>     <dbl>    <dbl> <dttm>              <dbl>
##  1 10730023     -86.8     33.6 2007-11-14 00:00:00  19.4
##  2 10732003     -86.9     33.5 2007-11-14 00:00:00  26.4
##  3 10890014     -86.6     34.7 2007-11-14 00:00:00  13.4
##  4 11011002     -86.3     32.4 2007-11-14 00:00:00  19.7
##  5 11130001     -85.0     32.5 2007-11-14 00:00:00  22.6
##  6 40139997    -112.      33.5 2007-11-14 00:00:00  12.3
##  7 40191028    -111.      32.3 2007-11-14 00:00:00   7.2
##  8 51190007     -92.3     34.8 2007-11-14 00:00:00  12.7
##  9 60070002    -122.      39.8 2007-11-14 00:00:00  10
## 10 60190008    -120.      36.8 2007-11-14 00:00:00  32.3
## # ... with 181 more rows
```

Aside - Splines



Splines in 1d - Smoothing Splines

These are a mathematical analogue to the drafting splines represented using a penalized regression model.

Splines in 1d - Smoothing Splines

These are a mathematical analogue to the drafting splines represented using a penalized regression model.

We want to find a function $f(x)$ that best fits our observed data $\mathbf{y} = y_1, \dots, y_n$ while being as *smooth* as possible.

$$\underset{f(x)}{\arg\min} \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2 + \lambda \int_{-\infty}^{\infty} f''(x)^2 \, dx$$

Interestingly, this minimization problem has an exact solution which is given by a mixture of weighted natural cubic splines (cubic splines that are linear in the tails) with knots at the observed data locations (xs).

Splines in 2d - Thin Plate Splines

Now imagine we have observed data of the form (x_i, y_i, z_i) where we wish to predict z_i given x_i and y_i for all i. We can naturally extend the smoothing spline model in two dimensions,

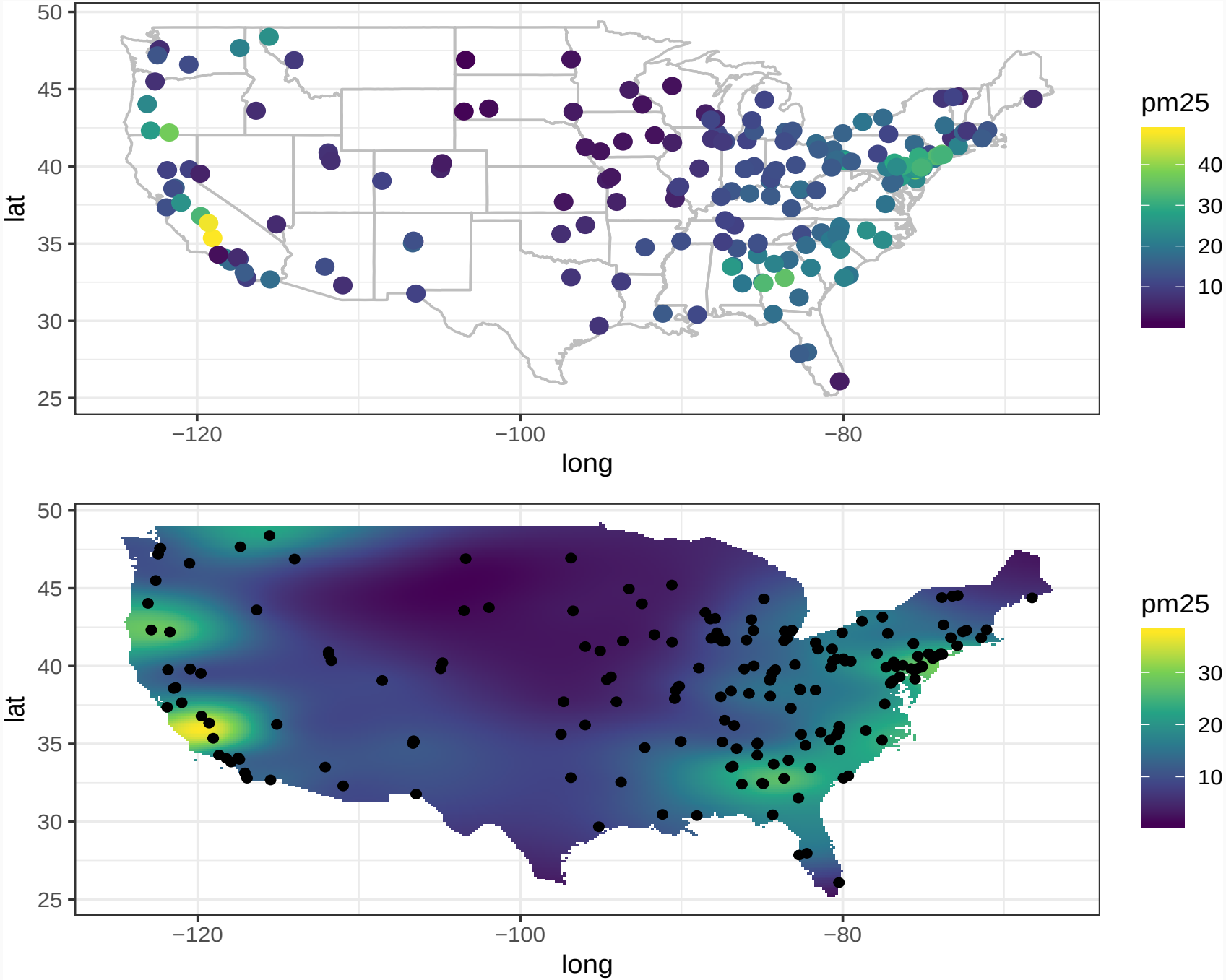
$$\underset{f(x,y)}{\arg\min} \sum_{i=1}^{n} (z_i - f(x_i, y_i))^2 + \lambda \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\frac{\partial^2 f}{\partial x^2} + 2 \frac{\partial^2 f}{\partial x \, \partial y} + \frac{\partial^2 f}{\partial y^2} \right) dx \, dy$$

The solution to this equation has a natural representation using a weighted sum of *radial basis functions* with knots at the observed data locations

$$f(x, y) = \sum_{i=1}^{n} w_i \, d(x_i, y_i)^2 \log d(x_i, y_i).$$

Fitting a TPS

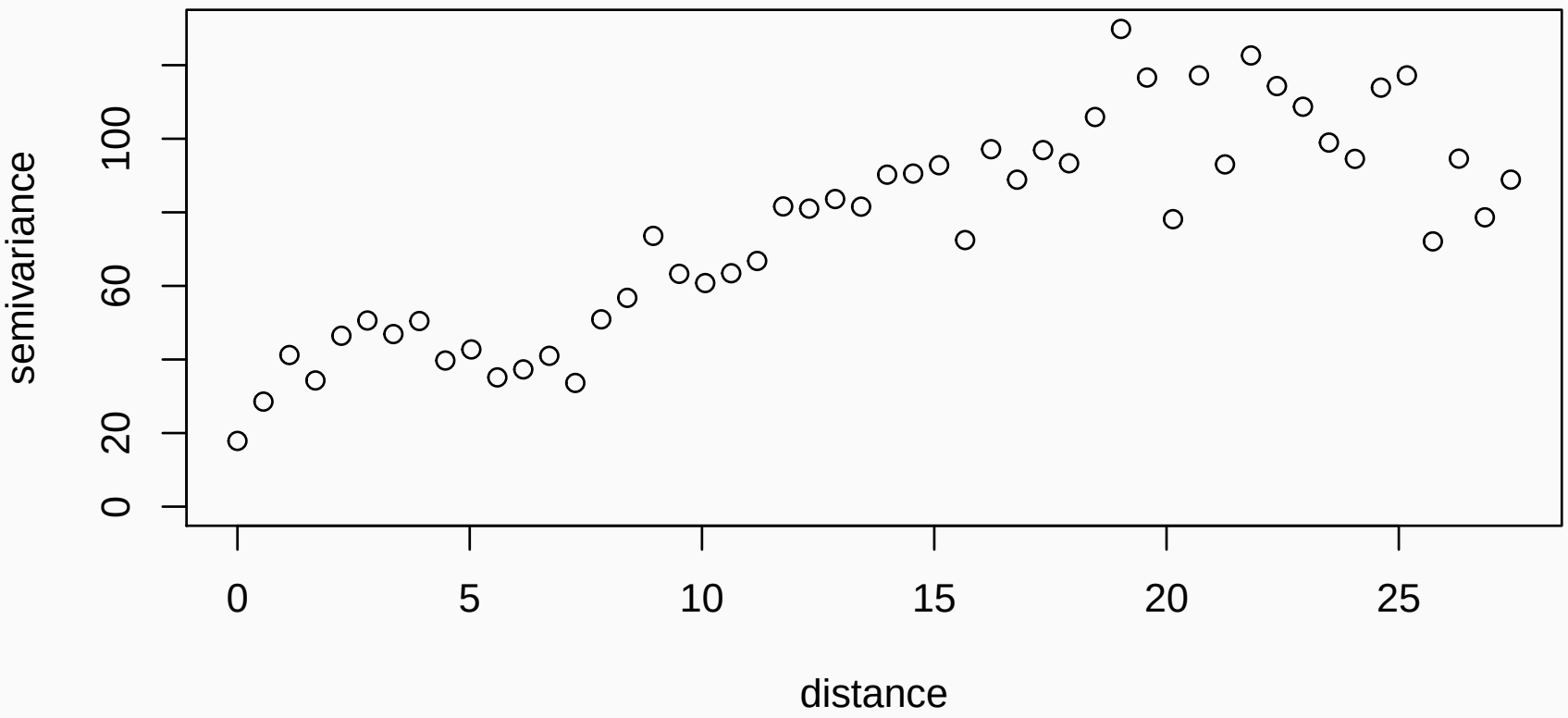
```
          select      long=          lat=          %>% as.matrix
tps = fields::Tps(x = coords, Y=csn$pm25)


data(wrld_simpl, package = "maptools")

r = raster::raster(nrows=200, ncol=400,
           xmn = min(csn$longitude)*1.05, xmx = max(csn$longitude)*0.95,
           ymn = min(csn$latitude )*0.95, ymx = max(csn$latitude )*1.05)

usa = raster::rasterize(wrld_simpl[wrld_simpl$NAME == "United States",], r)

cells = which(!is.na(usa[]))
pred_coords = raster::xyFromCell(r, cells)

pm25_pred = r
pm25_pred[cells] = predict(tps, pred_coords)

pm25_pred_df = as(pm25_pred, "SpatialPixelsDataFrame") %>%
  as.data.frame() %>%
  select(long=x, lat=y, pm25 = layer)
```


50
45
40
35
30
25
lat
−120
−100
−80
long
pm25
40
30
20
10
50
45
40
35
30
25
lat
−120
−100
−80
long
pm25
30
20
10

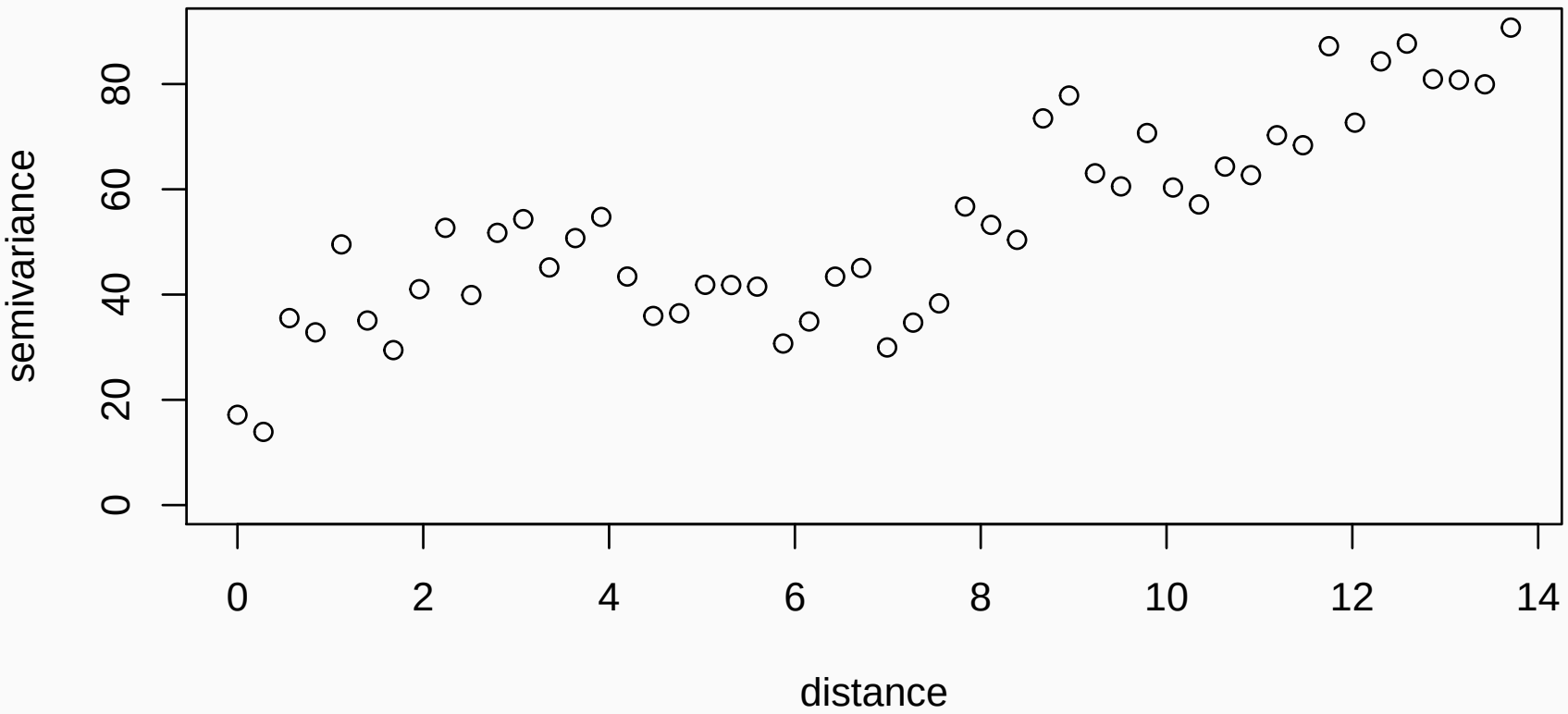
Gaussin Process Models / Kriging

Variogram

```
coords = csn %>% select(latitude, longitude) %>% as.matrix()
d = fields::rdist(coords)

geoR::variog(coords = coords, data = csn$pm25, messages = FALSE,
       uvec = seq(0, max(d)/2, length.out=50)) %>% plot()
```

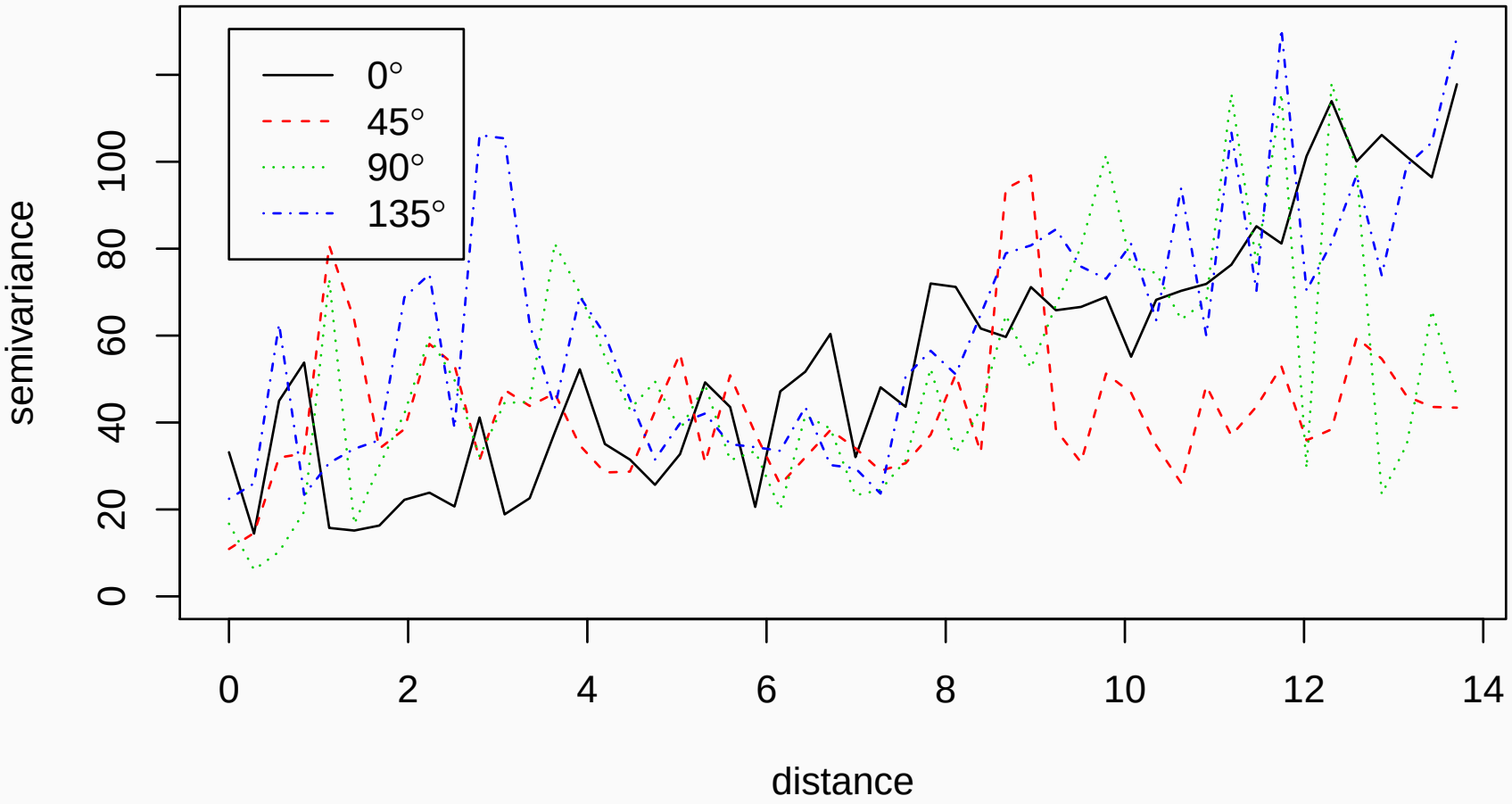


```
geoR::variog(coords = coords, data = csn$pm25, messages = FALSE,
       uvec = seq(0, max(d)/4, length.out=50)) %>% plot()
```



Isotropy / Anisotropy

```
v4 = geoR::variog4(coords = coords, data = csn$pm25, messages = FALSE,
                   uvec = seq(0, max(d)/4, length.out = 50))
plot(v4)
```



GP Spatial Model

If we assume that our data is *stationary* and *isotropic* then we can use a Gaussian Process model to fit the data. We will assume an exponential covariance structure.

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu},\, \Sigma)$$

$$\{\Sigma\}_{ij} = \sigma^2 \exp(-r \left\| s_i - s_j \right\|) + \sigma_n^2 \, 1_{i=j}$$

GP Spatial Model

If we assume that our data is *stationary* and *isotropic* then we can use a Gaussian Process model to fit the data. We will assume an exponential covariance structure.

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu},\, \Sigma)$$
$$\{\Sigma\}_{ij} = \sigma^2 \exp(-r \left\| s_i - s_j \right\|) + \sigma_n^2 \, 1_{i=j}$$

we can also view this as a spatial random effects model where

$$y(\mathbf{s}) = \mu(\mathbf{s}) + w(\mathbf{s}) + \epsilon(\mathbf{s})$$
$$w(\mathbf{s}) \sim \mathcal{N}(0, \Sigma')$$
$$\epsilon(s_i) \sim \mathcal{N}(0, \sigma_n^2)$$
$$\{\Sigma'\}_{ij} = \sigma^2 \exp(-r \left\| s_i - s_j \right\|)$$

Fitting with spBayes

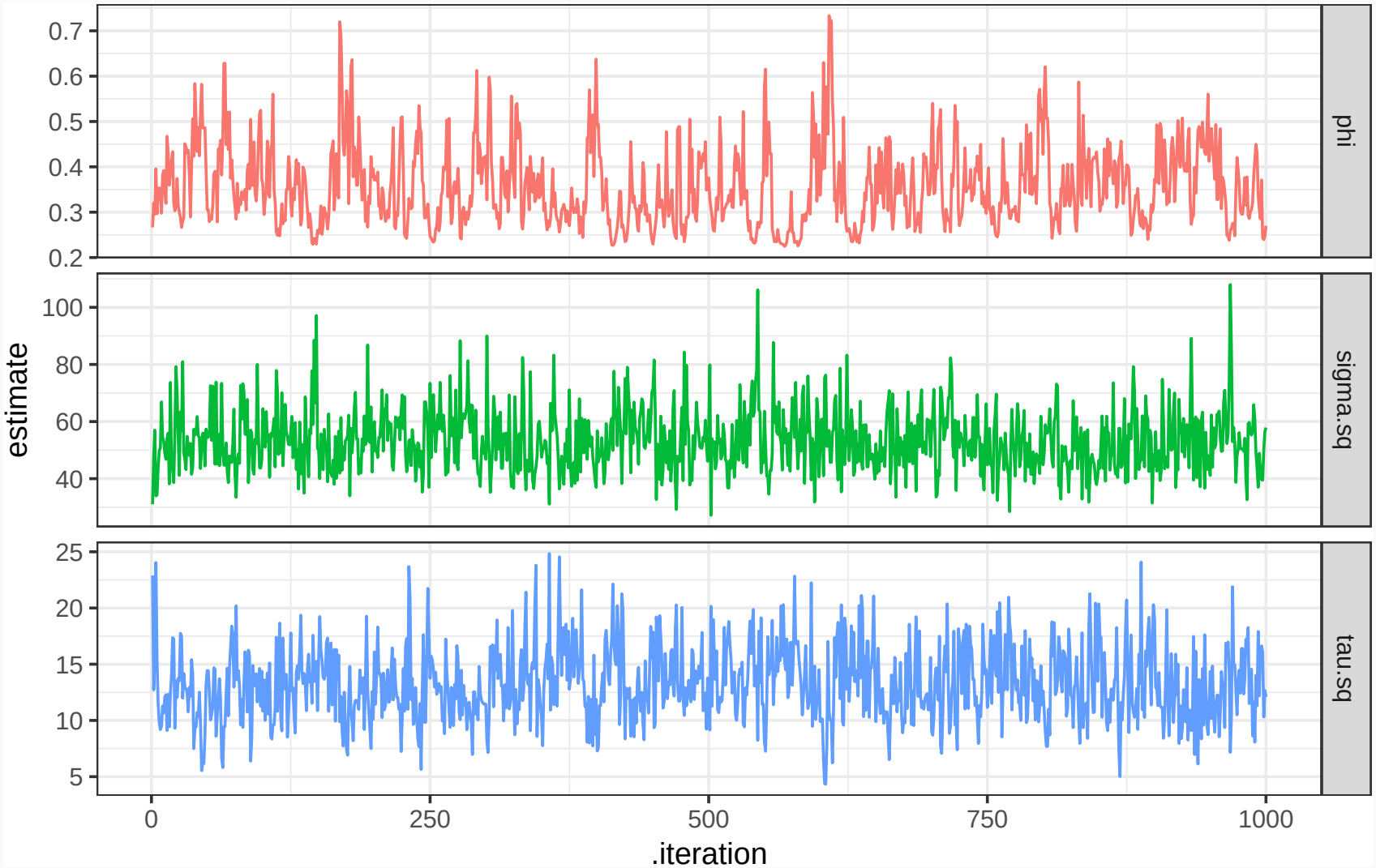
```
n = nrow(csn)
n_samp = 20000
coords = select(csn, longitude, latitude) %>% as.matrix()
max_range = max(dist(coords)) / 4


starting = list(phi = 3/3, sigma.sq = 33, tau.sq = 17)
tuning = list("phi"=0.1, "sigma.sq"=0.1, "tau.sq"=0.1)
priors = list(
  beta.Norm = list(0, 1000),
  phi.Unif = c(3/max_range, 3/(0.5)),
  sigma.sq.IG = c(2, 2),
  tau.sq.IG = c(2, 2)
)
```

```
m = spBayes::spLM(pm25 ~ 1, data = csn, coords = coords, starting = starting, priors = priors,
        cov.model = "exponential", n.samples = n_samp, tuning = tuning,
        n.report = n_samp/2)
## ----------------------------------------
##  General model description
## ----------------------------------------
## Model fit with 191 observations.
##
## Number of covariates 1 (including intercept if specified).
##
## Using the exponential spatial correlation model.
##
## Number of MCMC samples 20000.
##
## Priors and hyperpriors:
##  beta normal:
##  mu: 0.000
##  cov:
##  1000.000
##
##  sigma.sq IG hyperpriors shape=2.00000 and scale=2.00000
##  tau.sq IG hyperpriors shape=2.00000 and scale=2.00000
##  phi Unif hyperpriors a=0.21888 and b=6.00000
## ----------------------------------------------------
##      Sampling
## ----------------------------------------------------
## Sampled: 10000 of 20000, 50.00%
## Report interval Metrop. Acceptance rate: 32.56%
## Overall Metrop. Acceptance rate: 32.56%
## ----------------------------------------------------
## Sampled: 20000 of 20000, 100.00%
## Report interval Metrop. Acceptance rate: 32.82%
## Overall Metrop. Acceptance rate: 32.69%
## ----------------------------------------------------
```

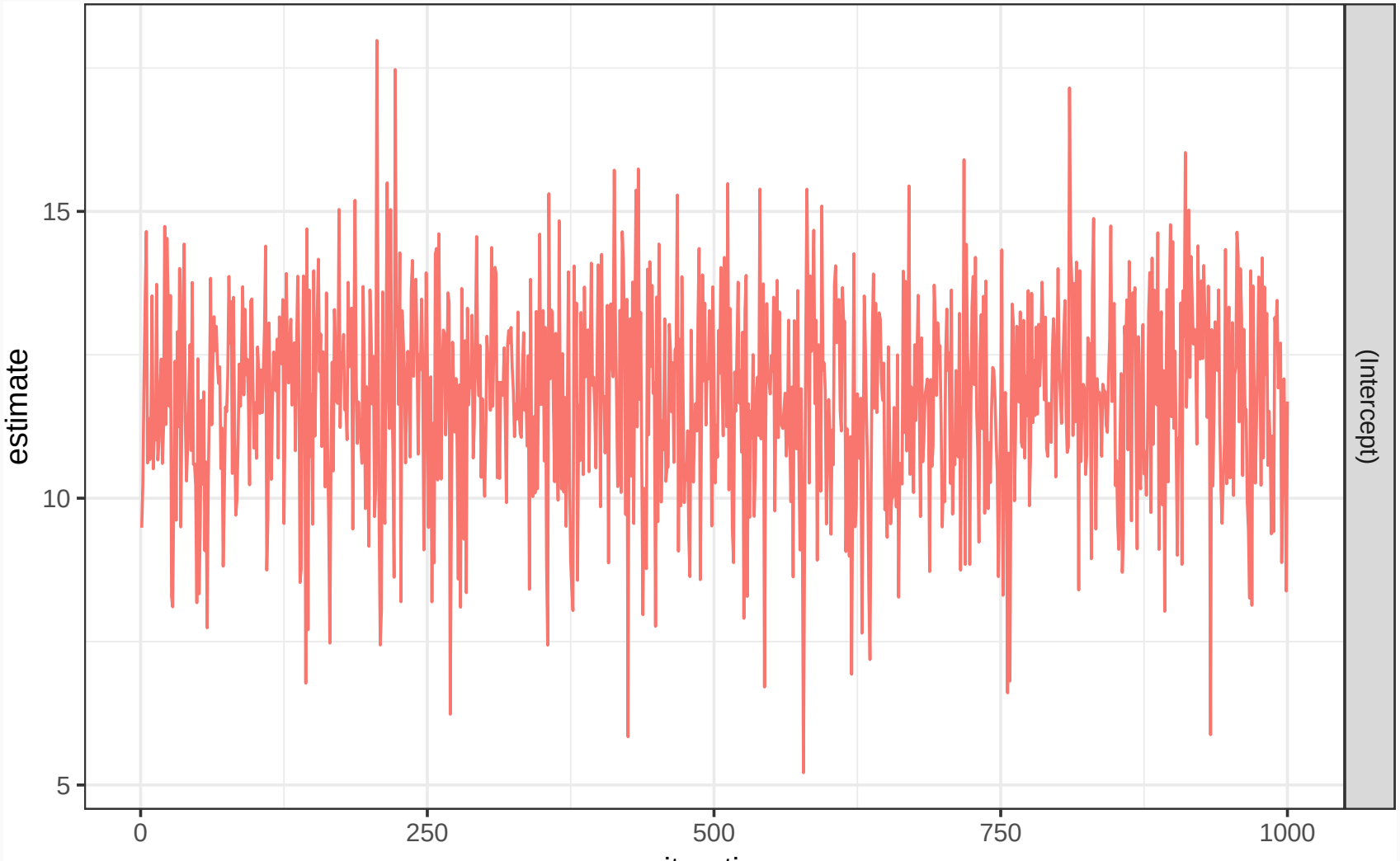
```
m = spBayes::spRecover(m, start=n_samp/2+1, thin = (n_samp/2)/1000)
## -------------------------------------------------
##      Recovering beta and w
## -------------------------------------------------
## Sampled: 99 of 1000, 9.90%
## Sampled: 199 of 1000, 19.90%
## Sampled: 299 of 1000, 29.90%
## Sampled: 399 of 1000, 39.90%
## Sampled: 499 of 1000, 49.90%
## Sampled: 599 of 1000, 59.90%
## Sampled: 699 of 1000, 69.90%
## Sampled: 799 of 1000, 79.90%
## Sampled: 899 of 1000, 89.90%
## Sampled: 999 of 1000, 99.90%
```

Parameter values

```
theta = m$p.theta.recover.samples %>%
  tidybayes::gather_samples(sigma.sq, tau.sq, phi)
```

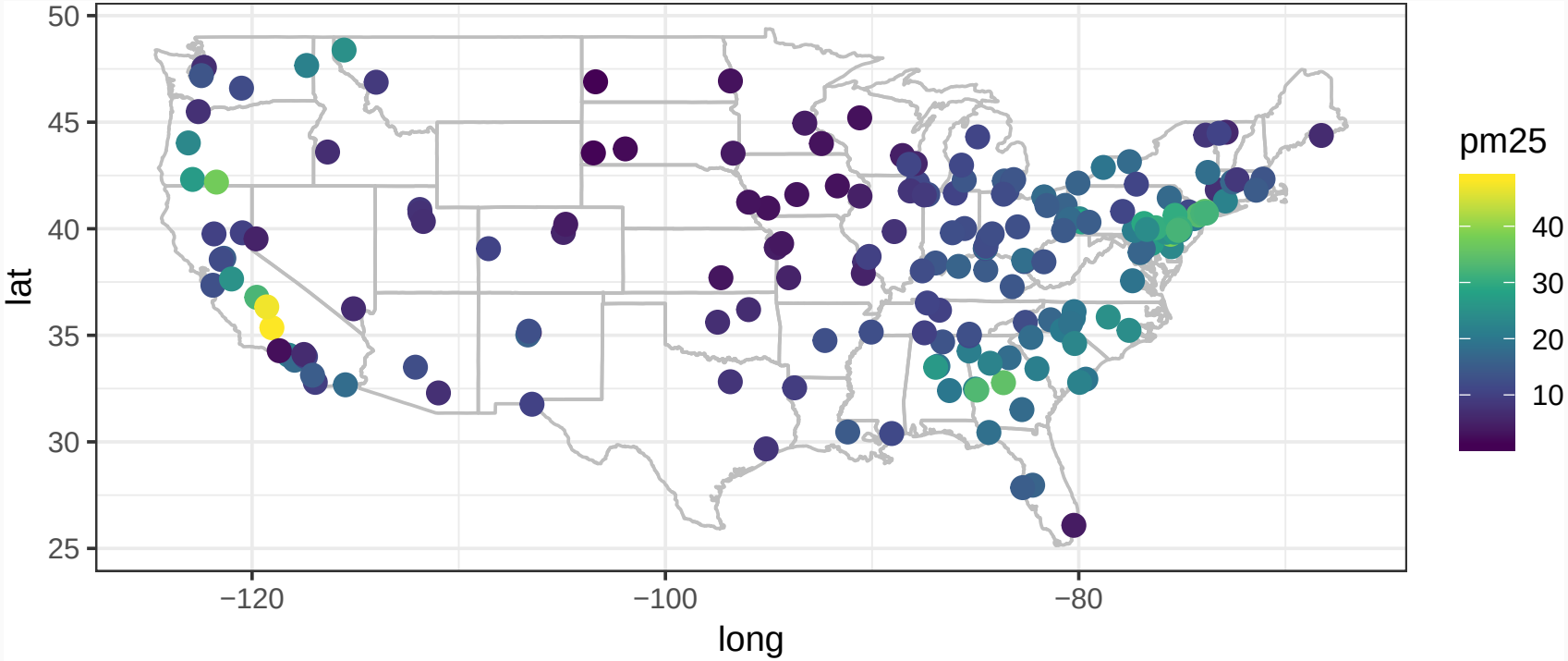


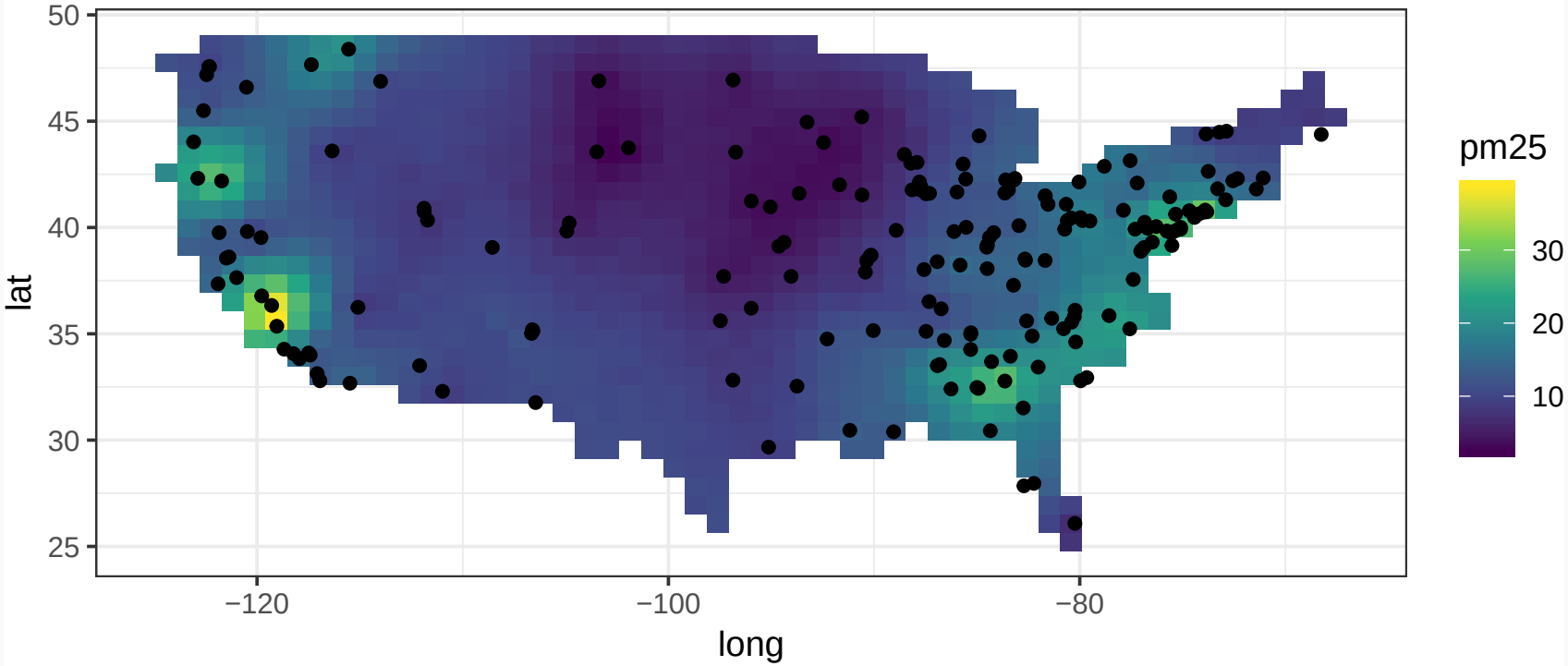
```
m$p.beta.recover.samples %>%
  tidybayes::gather_samples(`(Intercept)`) %>%
  ggplot(aes(x=.iteration, y=estimate, color=term)) +
    geom_line() +
    facet_grid(term~., scales = "free_y") +
    guides(color=FALSE)
```



Predictions

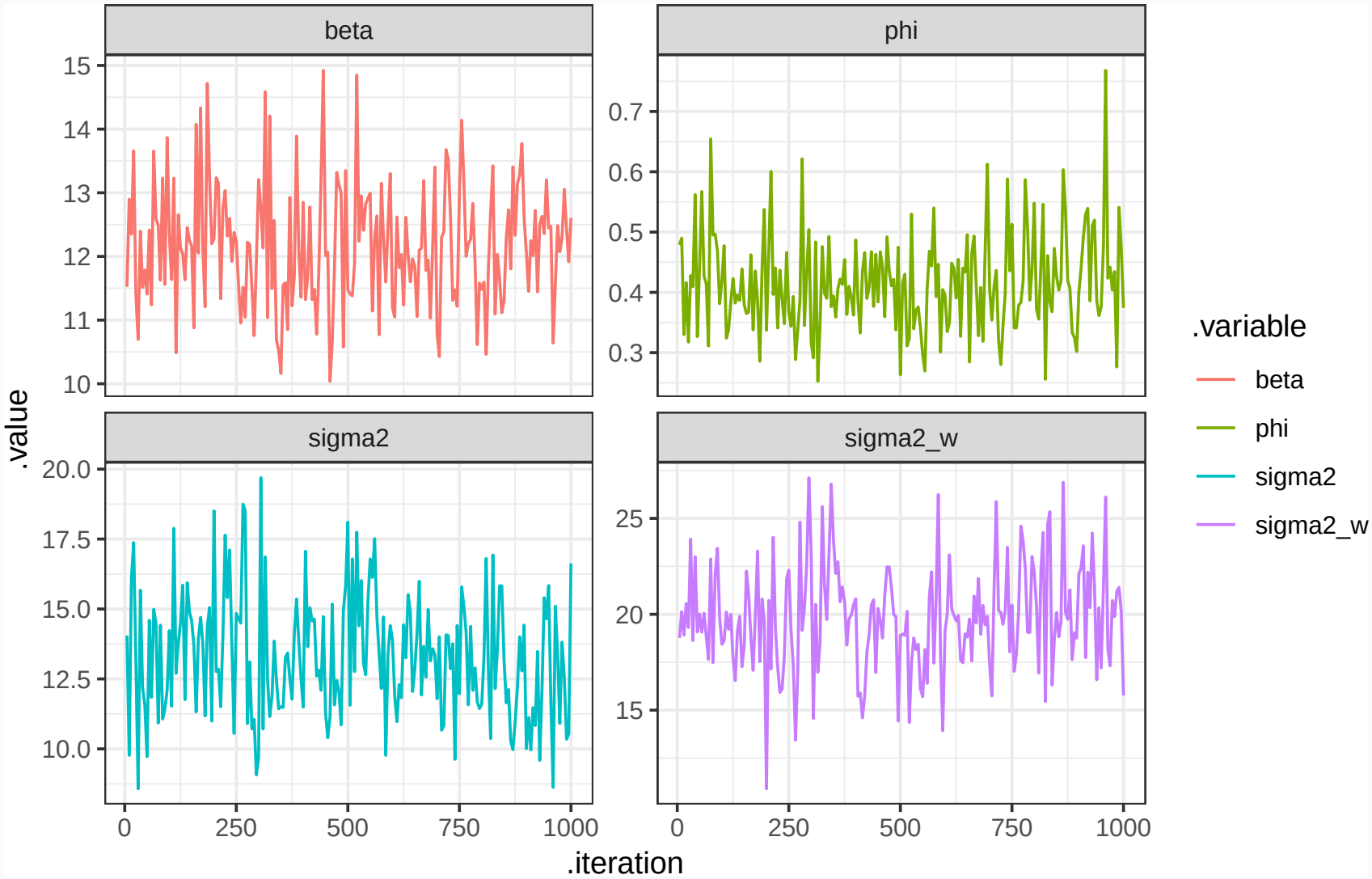
```
m_pred = spBayes::spPredict(m, pred_coords, pred.covars = matrix(1, nrow=nrow(pred_coords)),
                    start=n_samp/2+1, thin=(n_samp/2)/1000)
## ----------------------------------------
##  General model description
## ----------------------------------------
## Model fit with 191 observations.
##
## Prediction at 900 locations.
##
## Number of covariates 1 (including intercept if specified).
##
## Using the exponential spatial correlation model.
##
## ------------------------------------------------
##      Sampling
## ------------------------------------------------
## Sampled: 100 of 1000, 9.90%
## Sampled: 200 of 1000, 19.90%
## Sampled: 300 of 1000, 29.90%
## Sampled: 400 of 1000, 39.90%
## Sampled: 500 of 1000, 49.90%
## Sampled: 600 of 1000, 59.90%
## Sampled: 700 of 1000, 69.90%
## Sampled: 800 of 1000, 79.90%
## Sampled: 900 of 1000, 89.90%
## Sampled: 1000 of 1000, 99.90%
m_pred_summary = post_summary(t(m_pred$p.y.predictive.samples))
```


50
45
40
35
30
25
lat
−120
−100
−80
long
pm25
40
30
20
10


50
45
40
35
30
25
lat
−120
−100
−80
long
pm25
30
20
10

JAGs Model

```
gplm = "model{
  for(i in 1:length(y)){
    y[i] ~ dnorm(beta + w[i], tau)
    mu_w[i] = 0
  }

  for(i in 1:length(y)){
    for(j in 1:length(y)){
      Sigma_w[i,j] = sigma2_w * exp(-phi * d[i,j])
    }
  }
  w ~ dmnorm(mu_w, inverse(Sigma_w))

  beta ~ dnorm(0, 1/1000)
  sigma2_w ~ dgamma(2, 2)
  sigma2 ~ dgamma(2, 2)
  tau = 1/sigma2
  phi ~ dunif(3/14, 3/0.5)
}"
```


beta
phi
sigma2
sigma2_w
.value
.iteration
.variable
beta
phi
sigma2
sigma2_w
15
14
13
12
11
10
0.7
0.6
0.5
0.4
0.3
20.0
17.5
15.0
12.5
10.0
25
20
15
0
250
500
750
1000

Comparing Model Parameters