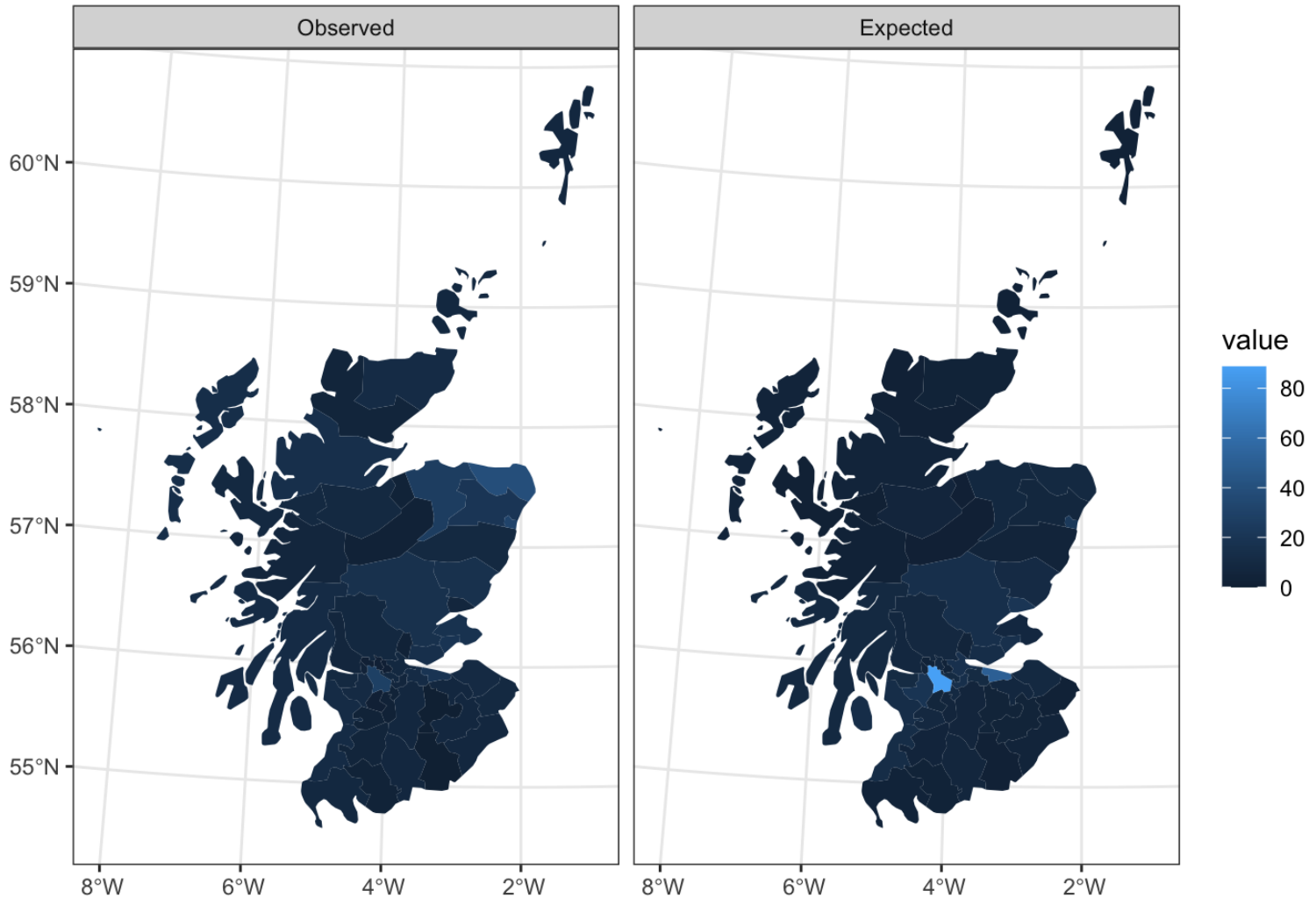
Spatial GLM + Point Reference Spatial Data

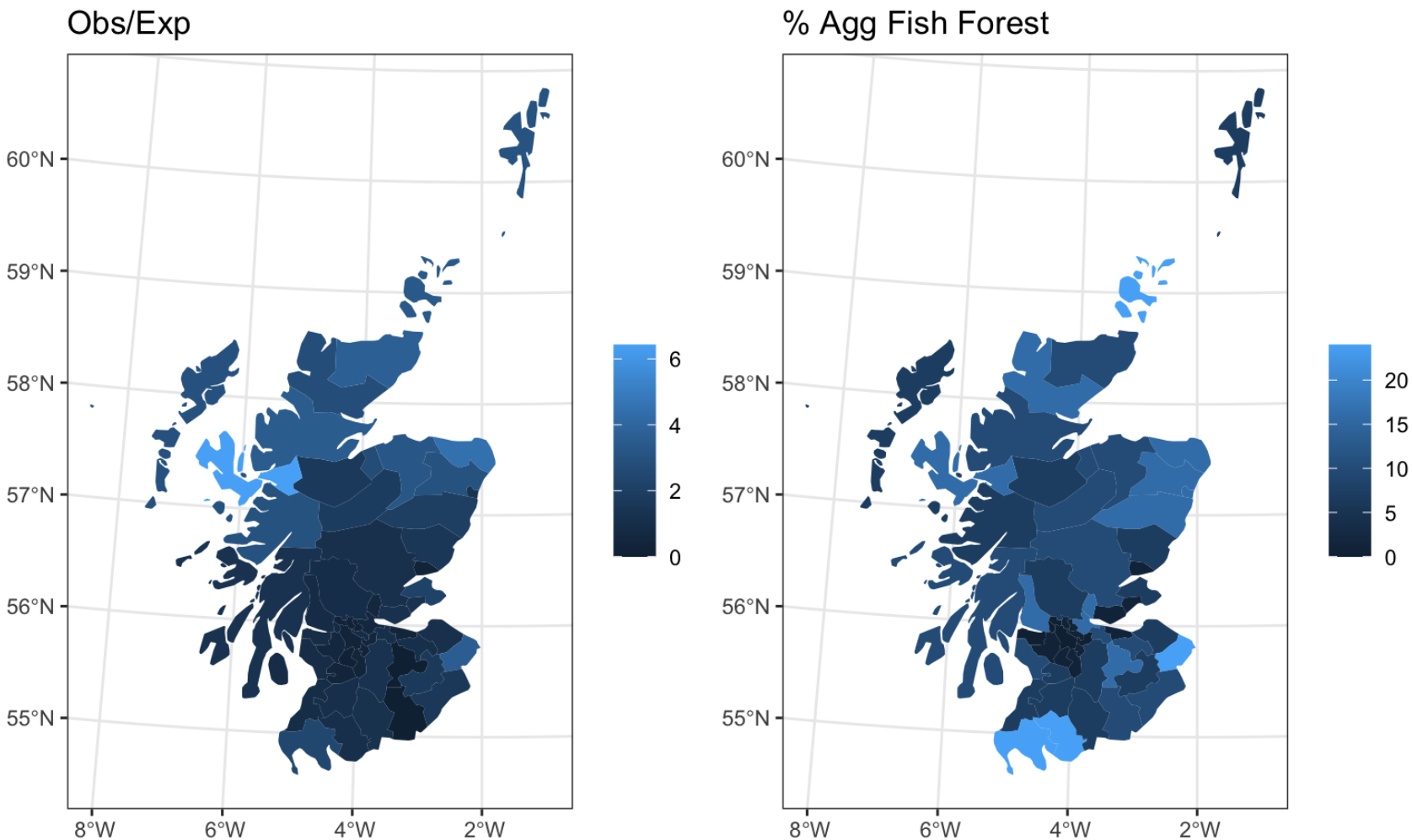
Lecture 20

Dr. Colin Rundel

Spatial GLM Models

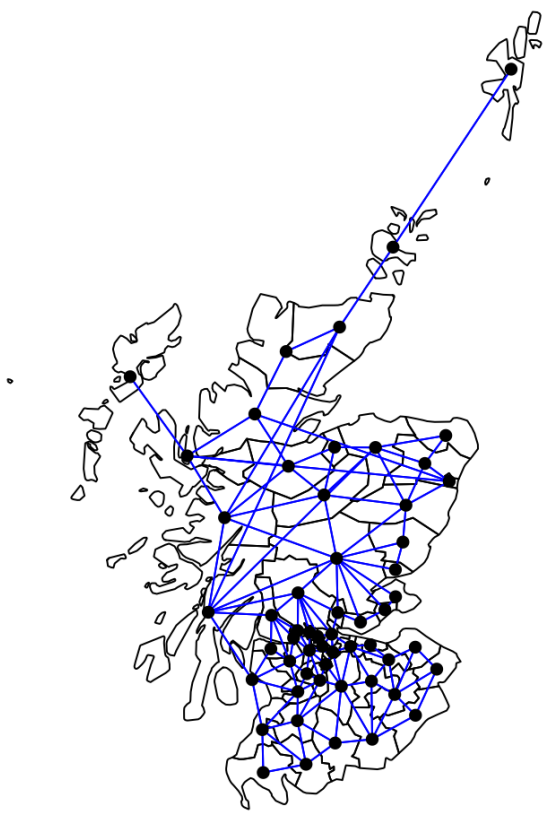
Scottish Lip Cancer Data




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

```
A = (st_distance(lip_cancer) |> unclass()) < 1e-6
diag(A) = 0
rownames(A) = lip_cancer$District
colnames(A) = lip_cancer$District

listw = spdep::mat2listw(A, style="B")
```



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 = 2.222,
p-value = 0.01314
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation
      0.161470014      -0.018181818
         Variance
      0.006537137
```

```
spdep::moran.test(
  lip_cancer$Observed / lip_cancer$Expected,
  listw
)
```

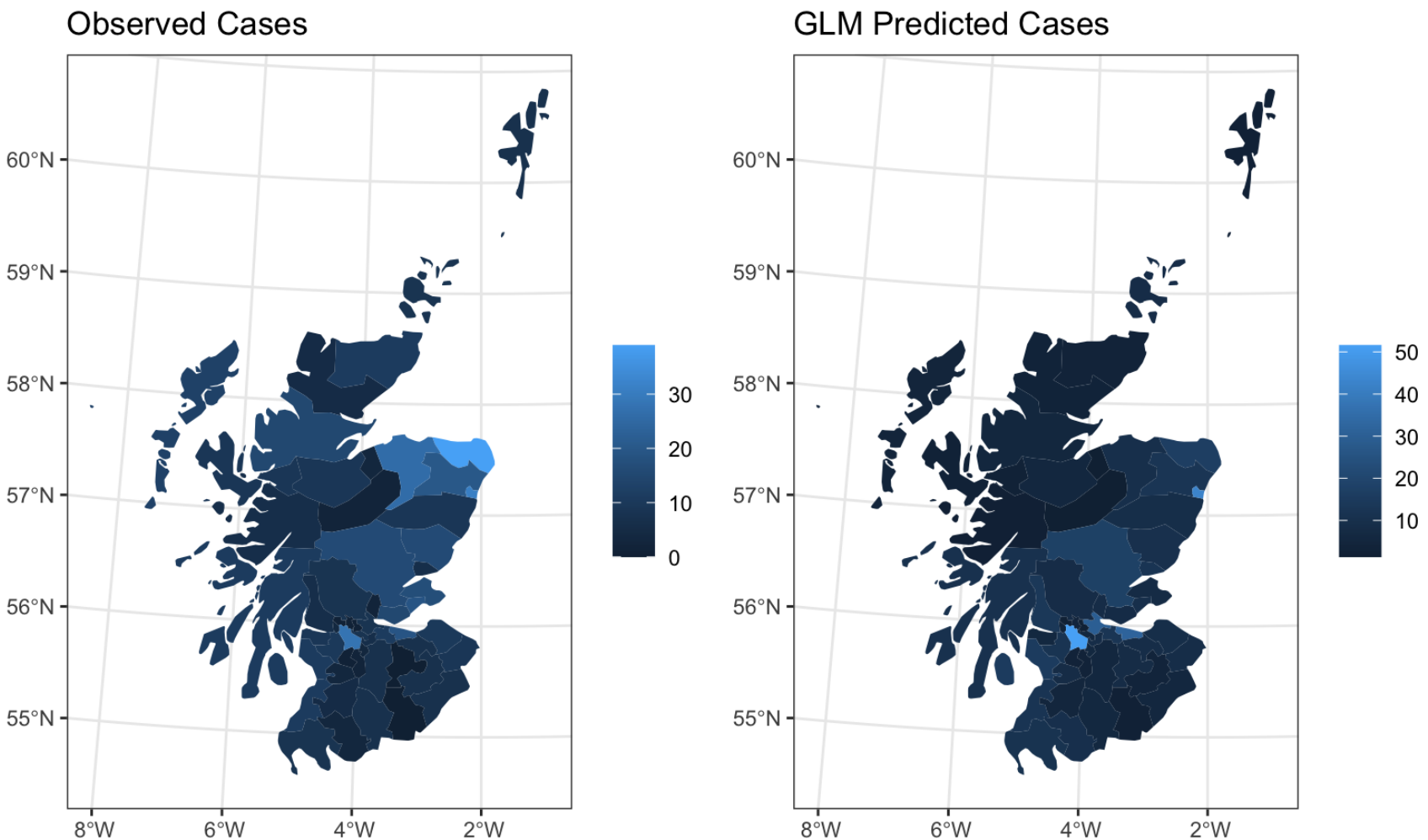
```
    Moran I test under randomisation

data:  lip_cancer$Observed/lip_cancer$Expected
weights: listw

Moran I statistic standard deviate = 6.3552,
p-value = 1.041e-10
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation
      0.500062930      -0.018181818
         Variance
      0.006649799
```

GLM

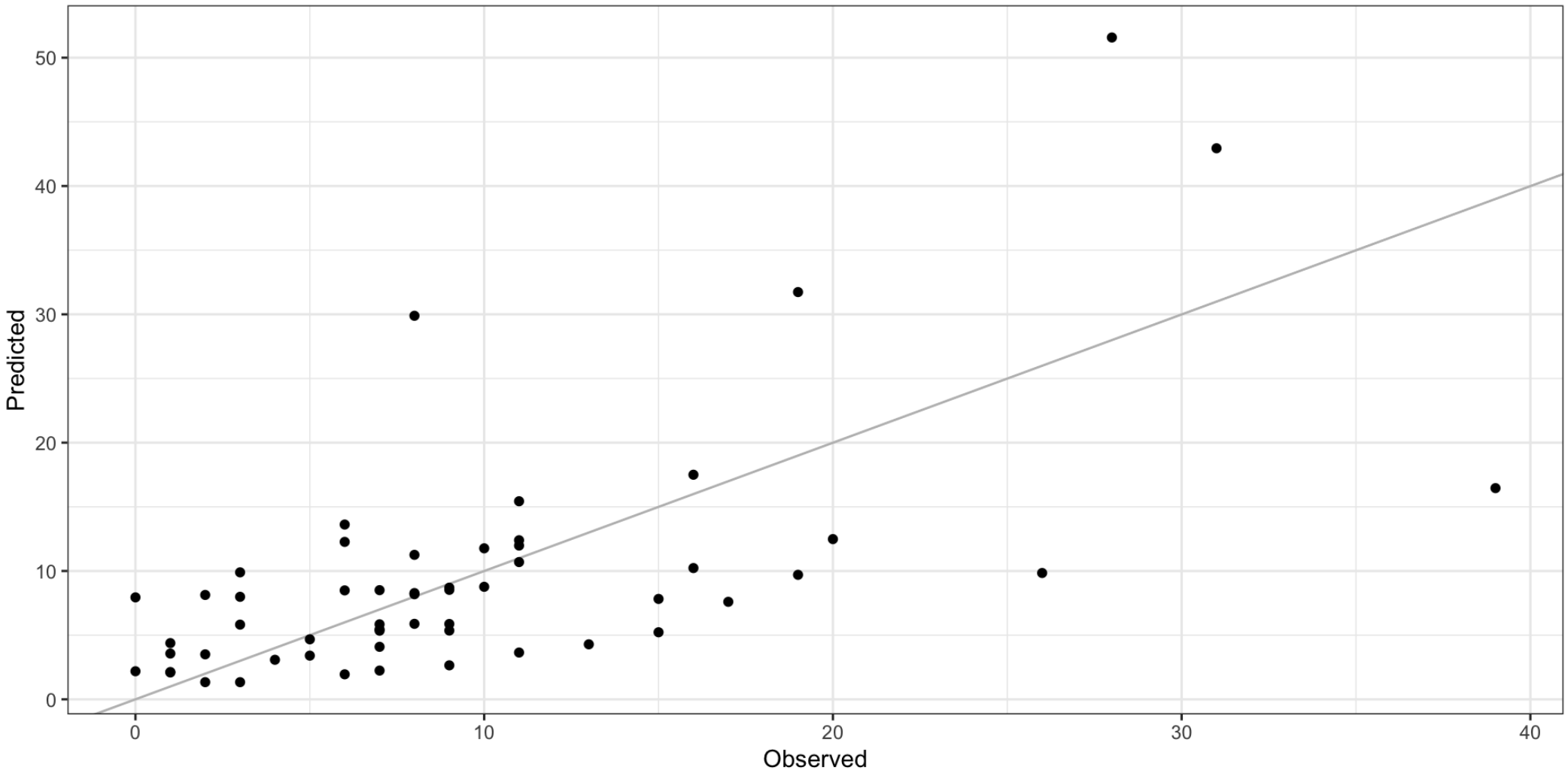
```r
l = glm(Observed ~ offset(log(Expected)) + pcaff,
        family="poisson", data=lip_cancer)
summary(l)
```

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

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
```

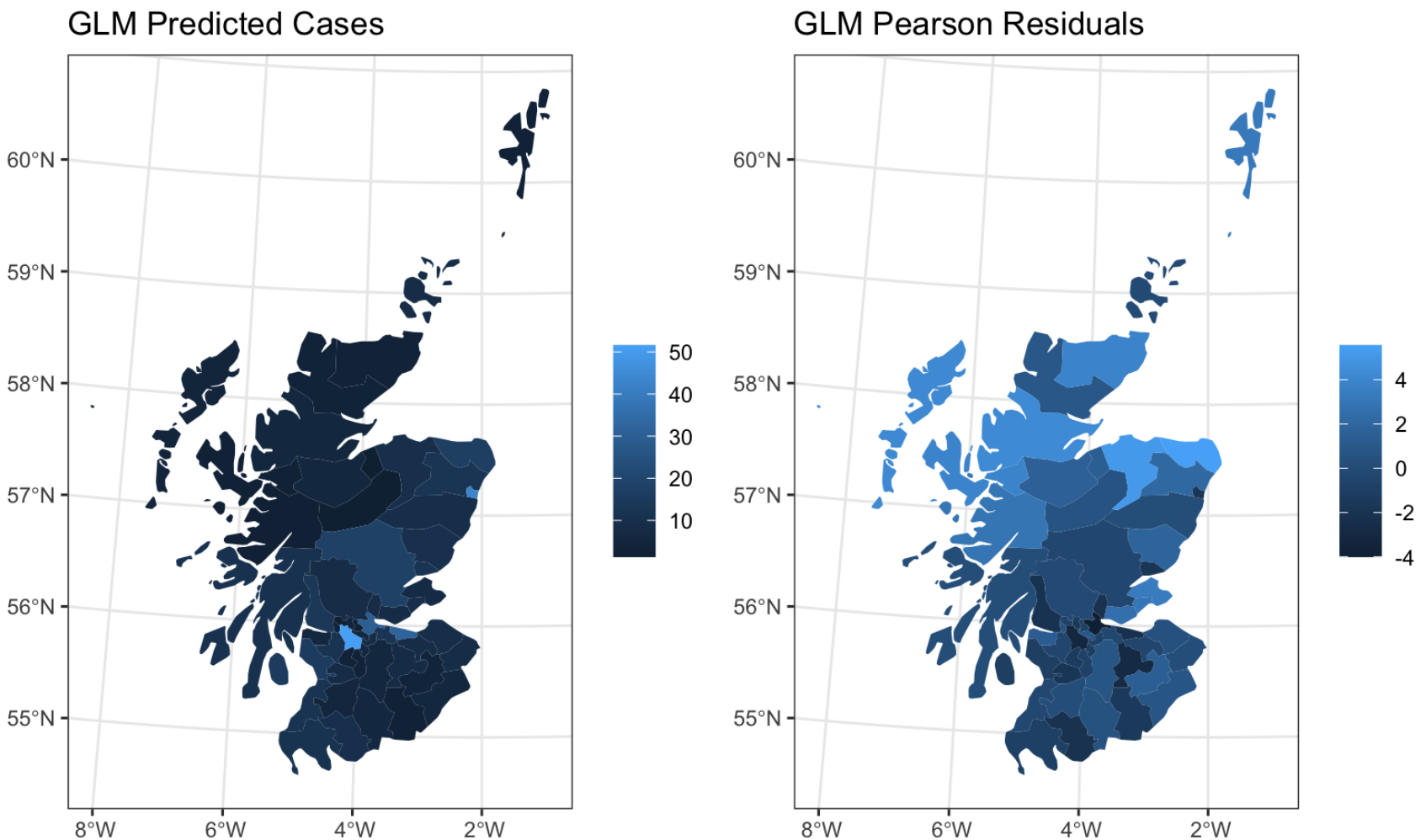
GLM Fit


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

```
1 yardstick::rmse_vec(lip_cancer_fit$Observed, lip_cancer_fit$.fitted)
```

```
[1] 7.480889
```

Moran's I

```
1 spdep::moran.test(lip_cancer_fit$.resid, listw)
```

```
    Moran I test under randomisation

data:  lip_cancer_fit$.resid
weights: listw

Moran I statistic standard deviate = 3.9739, p-value = 3.535e-05
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.31317350       -0.01818182        0.00695255
```

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.

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

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

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

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

CAR model

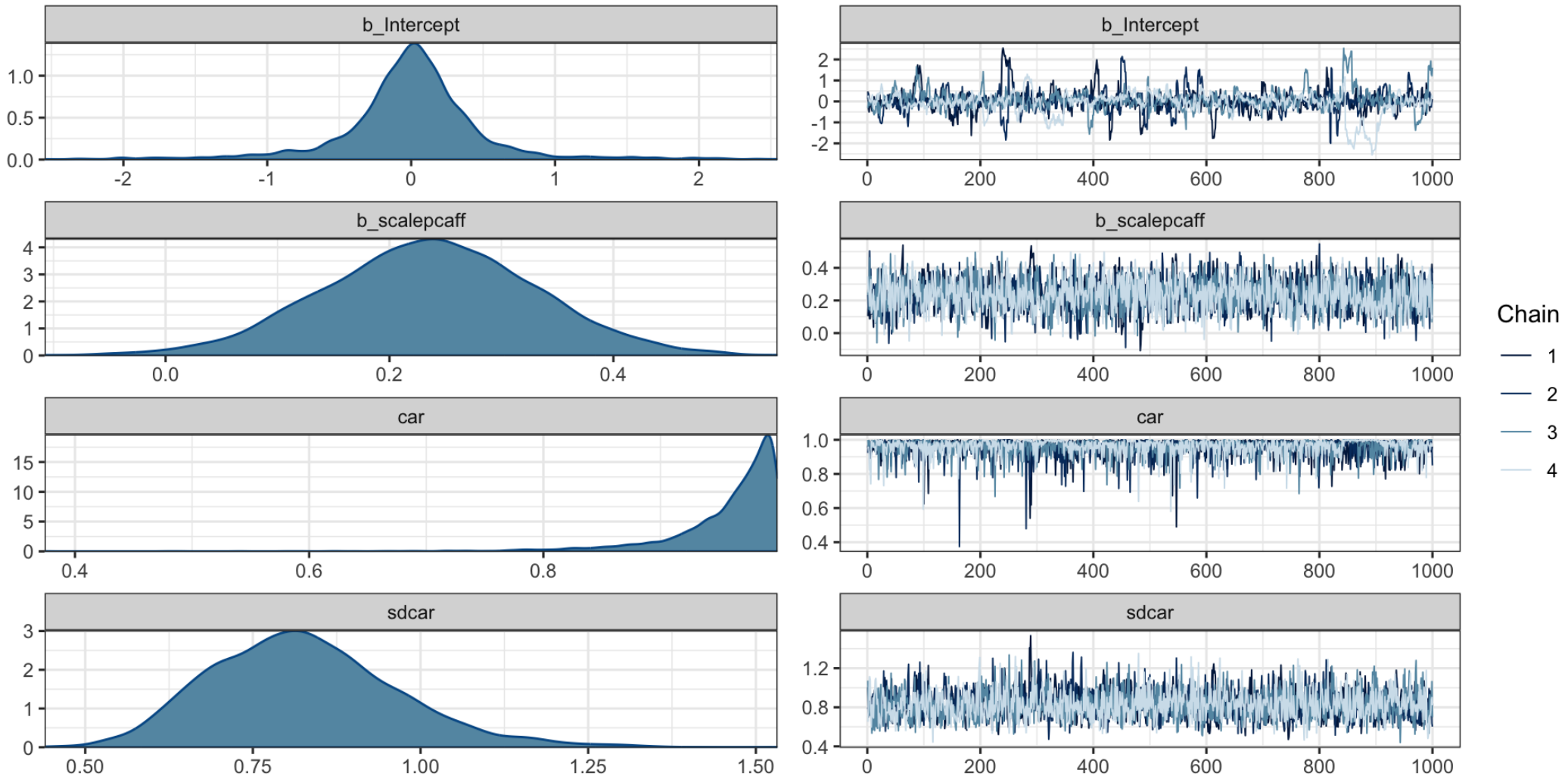
```
b_car = brms::brm(
  Observed~offset(log(Expected))+scale(pcaff)+car(A, gr=District),
  data=lip_cancer, data2=list(A=A),
  family = poisson, cores=4, iter=10000, thin=5
)
```

```
1 b_car
```

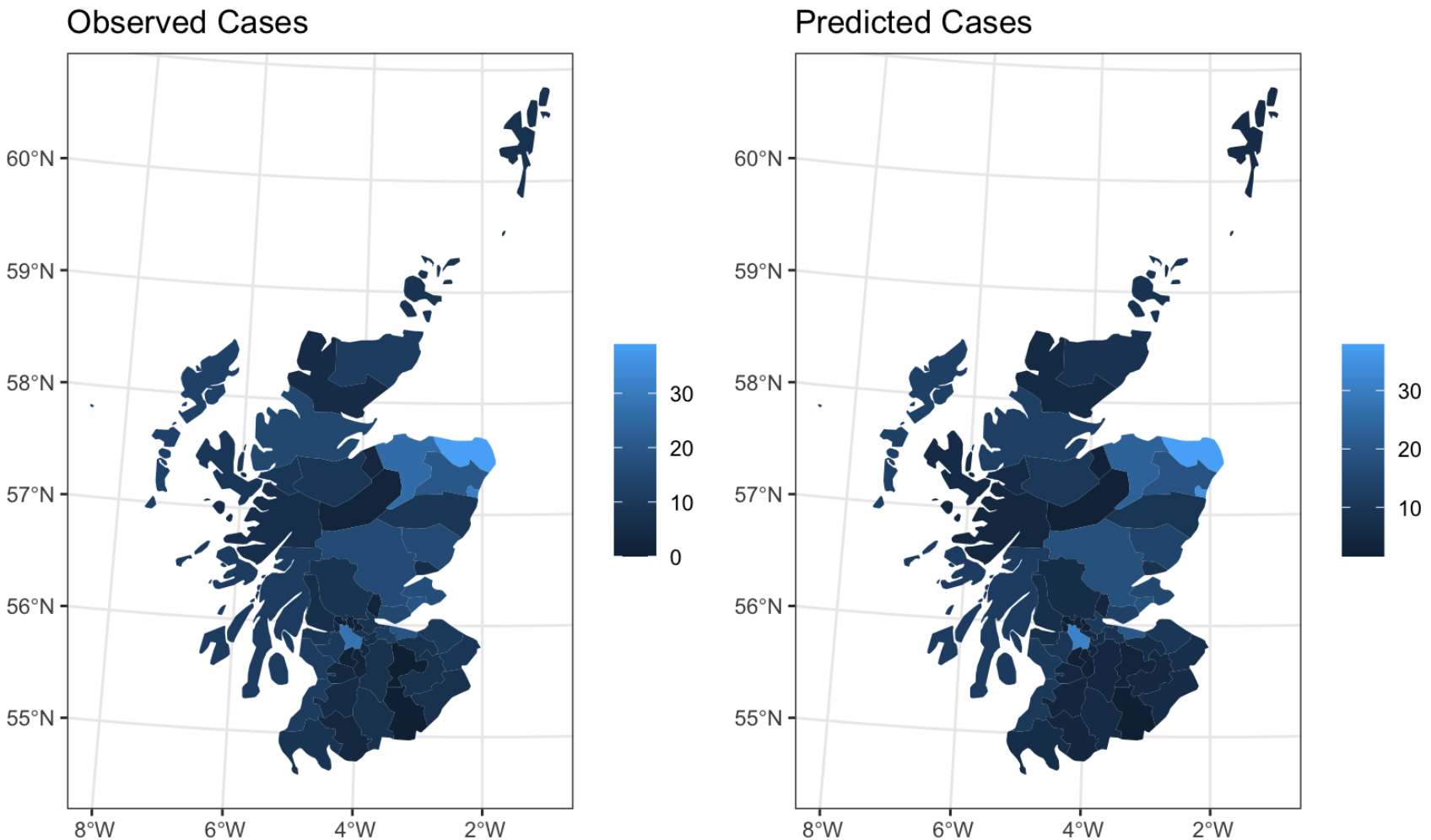
```
 Family: poisson
  Links: mu = log
Formula: Observed ~ offset(log(Expected)) + scale(pcaff) + car(A, gr = District)
   Data: lip_cancer (Number of observations: 56)
  Draws: 4 chains, each with iter = 10000; warmup = 5000; thin = 5;
         total post-warmup draws = 4000

Correlation Structures:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
car       0.96      0.05     0.82     1.00 1.00      717      636
sdcar     0.83      0.14     0.59     1.14 1.00     2122     2819

Population-Level Effects:
           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept      0.00      0.51    -1.15     1.17 1.01      324      179
scalepcaff     0.24      0.09     0.05     0.42 1.00     2272     2786

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

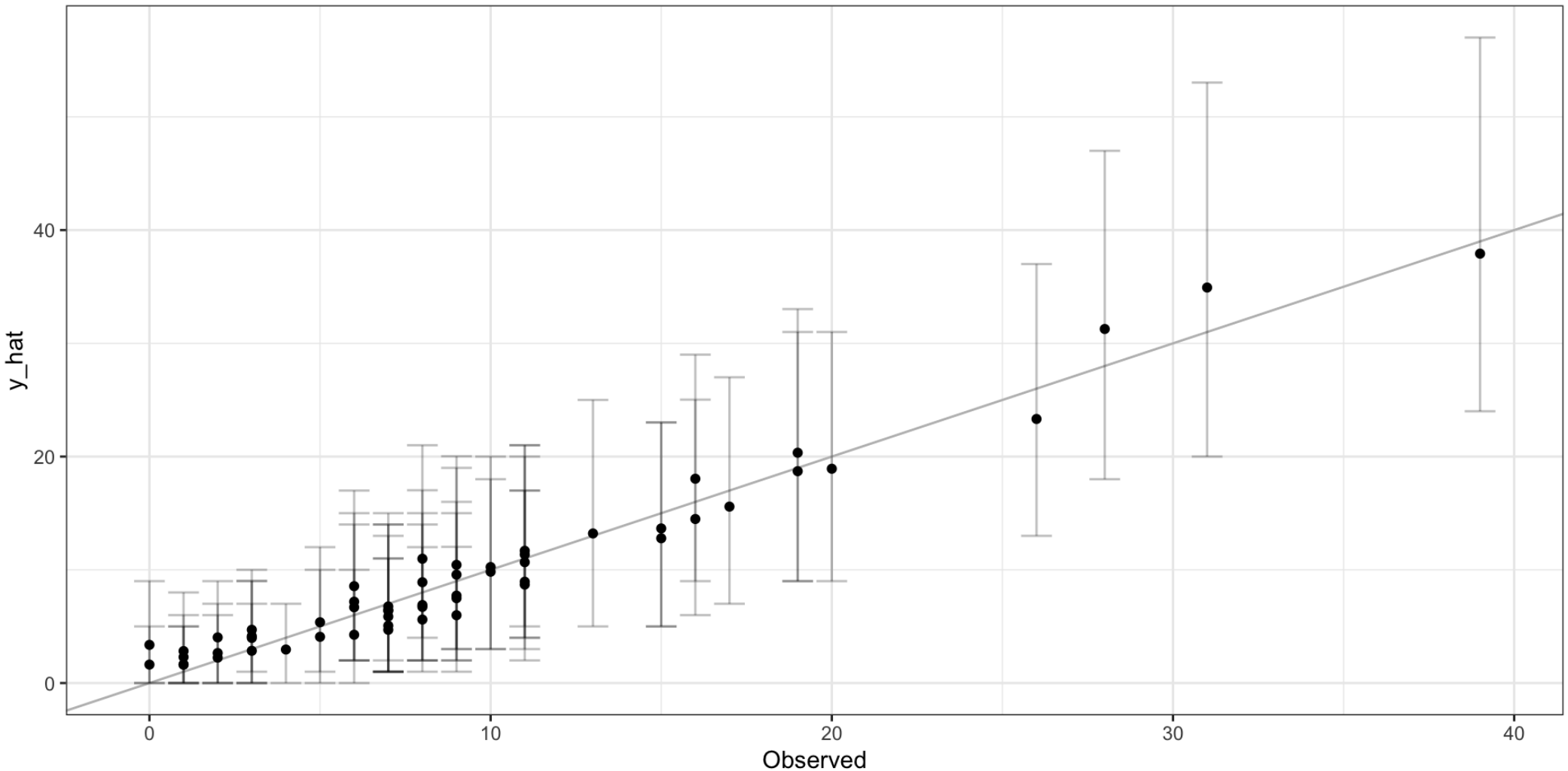
```
1 plot(b_car)
```



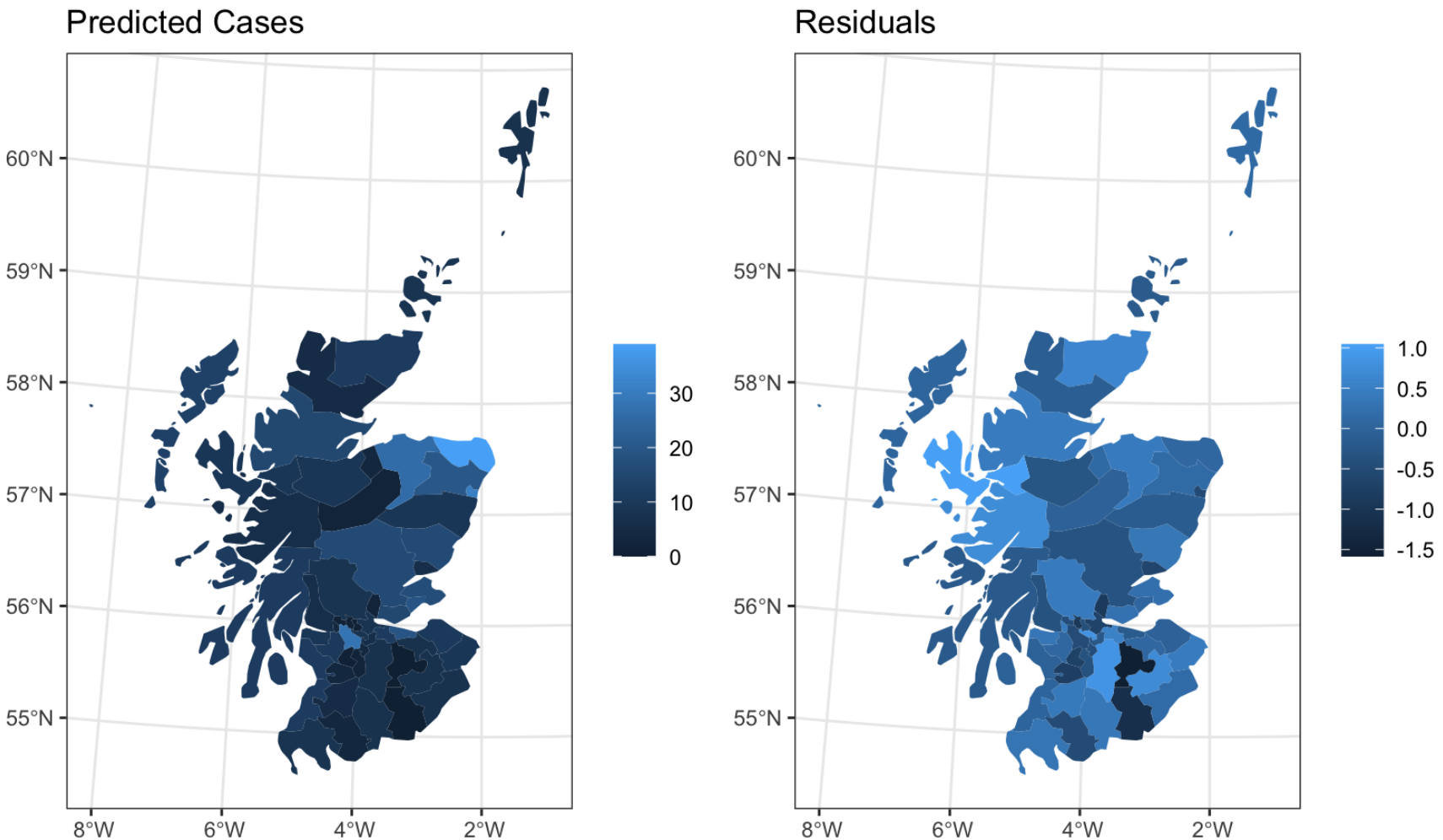
Predictions



Observed vs predicted



Residuals


Predicted 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
Residuals
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
1.0
0.5
0.0
-0.5
-1.0
-1.5

Results

RMSE

```
1 yardstick::rmse_vec(b_car_pred$Observed, b_car_pred$y_pred)
```

```
[1] 1.640872
```

Moran's I

```
1 spdep::moran.test(b_car_resid$resid, listw)
```

```
    Moran I test under randomisation

data:  b_car_resid$resid
weights: listw

Moran I statistic standard deviate = 1.5214, p-value = 0.06407
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.108093866      -0.018181818       0.006888589
```

IAR Model

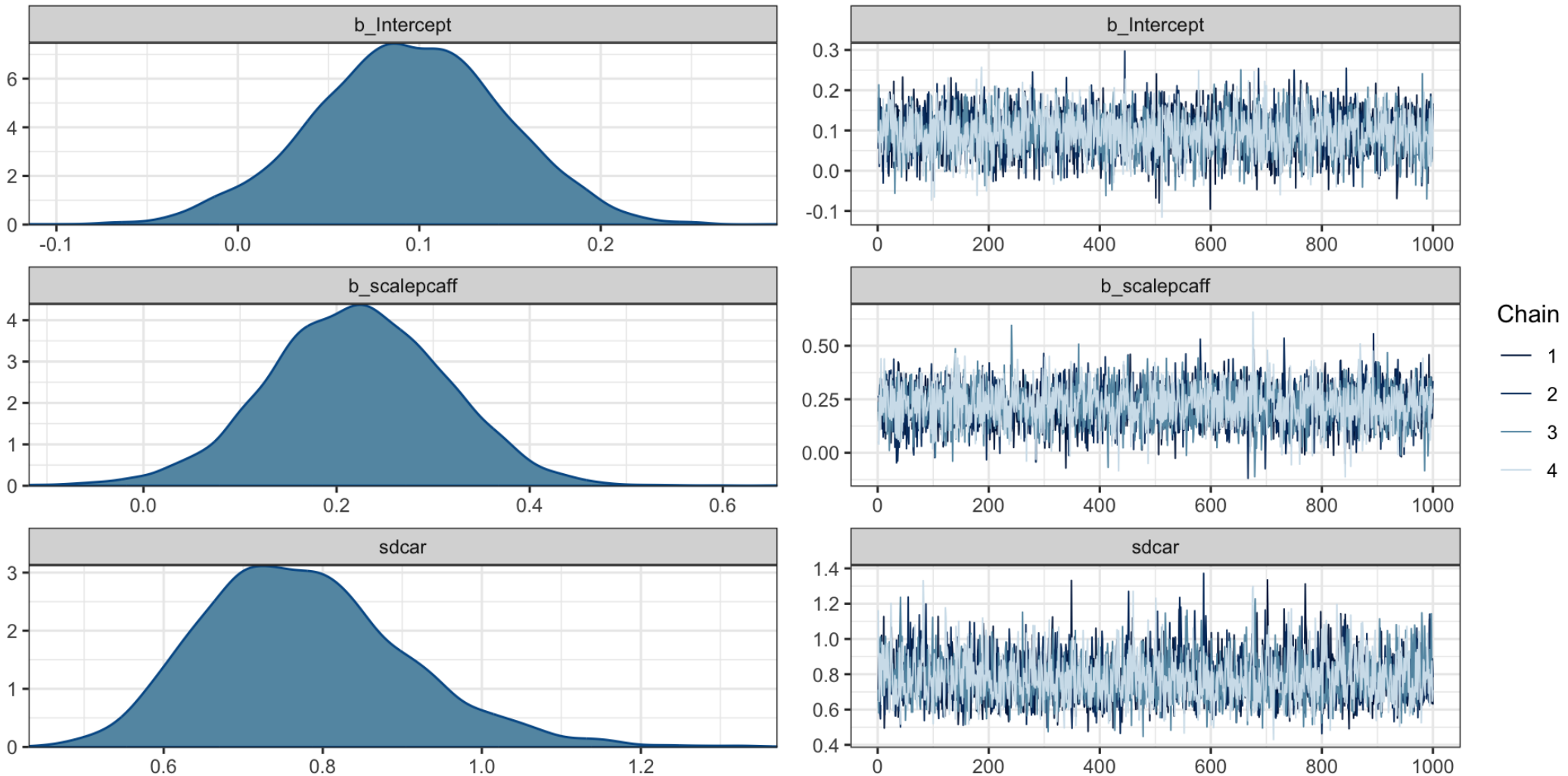
Intrinsic Autoregressive Model (IAR)

```
b_iar = brms::brm(
  Observed~offset(log(Expected))+scale(pcaff)+car(A, gr=District, type='
  data=lip_cancer, data2=list(A=A),
  family = poisson, cores=4, iter=10000, thin=5
)
```

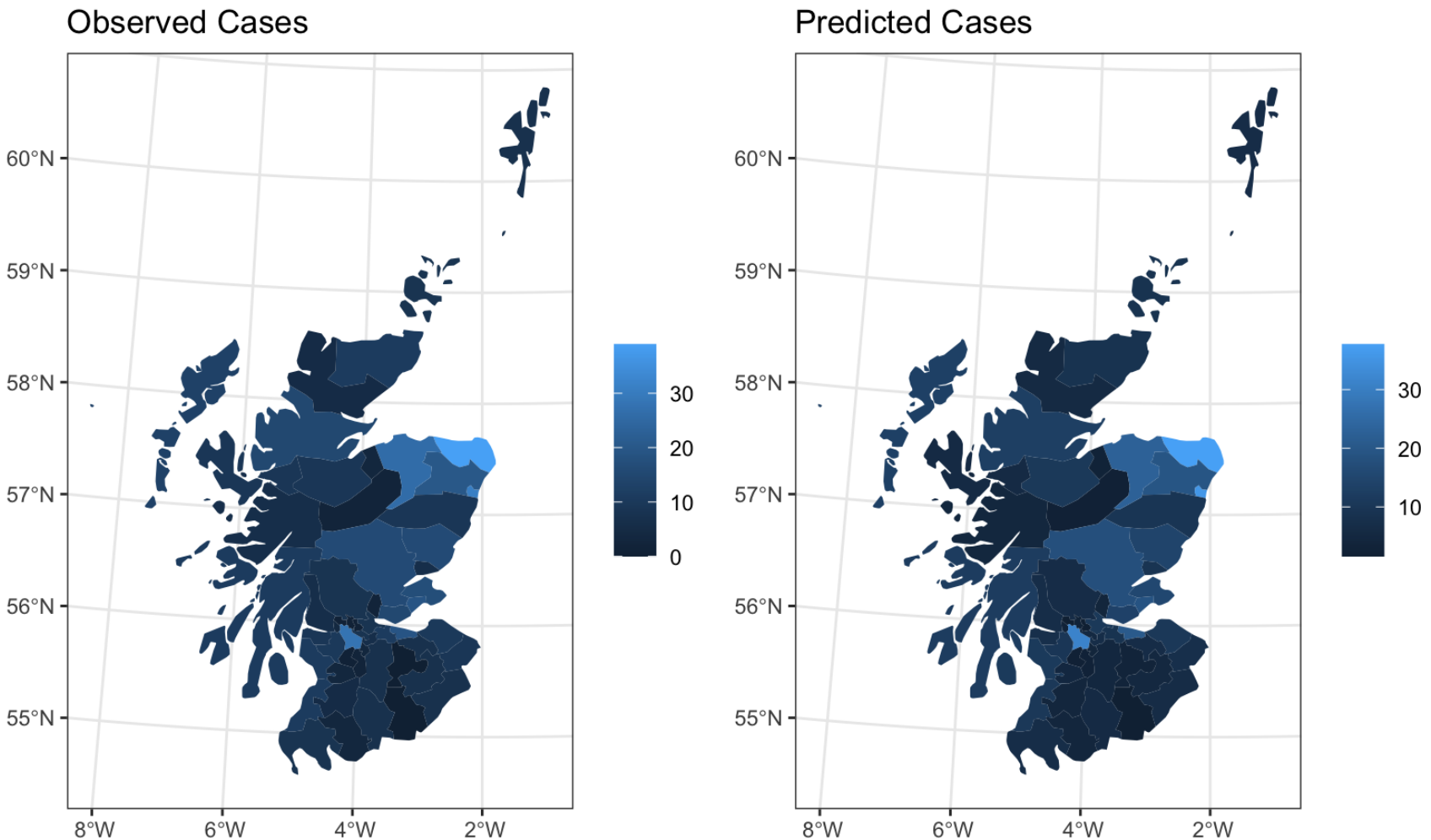
```
1 b_iar
```

```
 Family: poisson
  Links: mu = log
Formula: Observed ~ offset(log(Expected)) + scale(pcaff) + car(A, gr = District, type = "icar")
   Data: lip_cancer (Number of observations: 56)
  Draws: 4 chains, each with iter = 10000; warmup = 5000; thin = 5;
         total post-warmup draws = 4000

Correlation Structures:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sdcar     0.78      0.13     0.56     1.06 1.00     2827     3093

Population-Level Effects:
           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept      0.09      0.05    -0.01     0.19 1.00     3970     3968
scalepcaff     0.22      0.09     0.04     0.39 1.00     3854     3638

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).
```

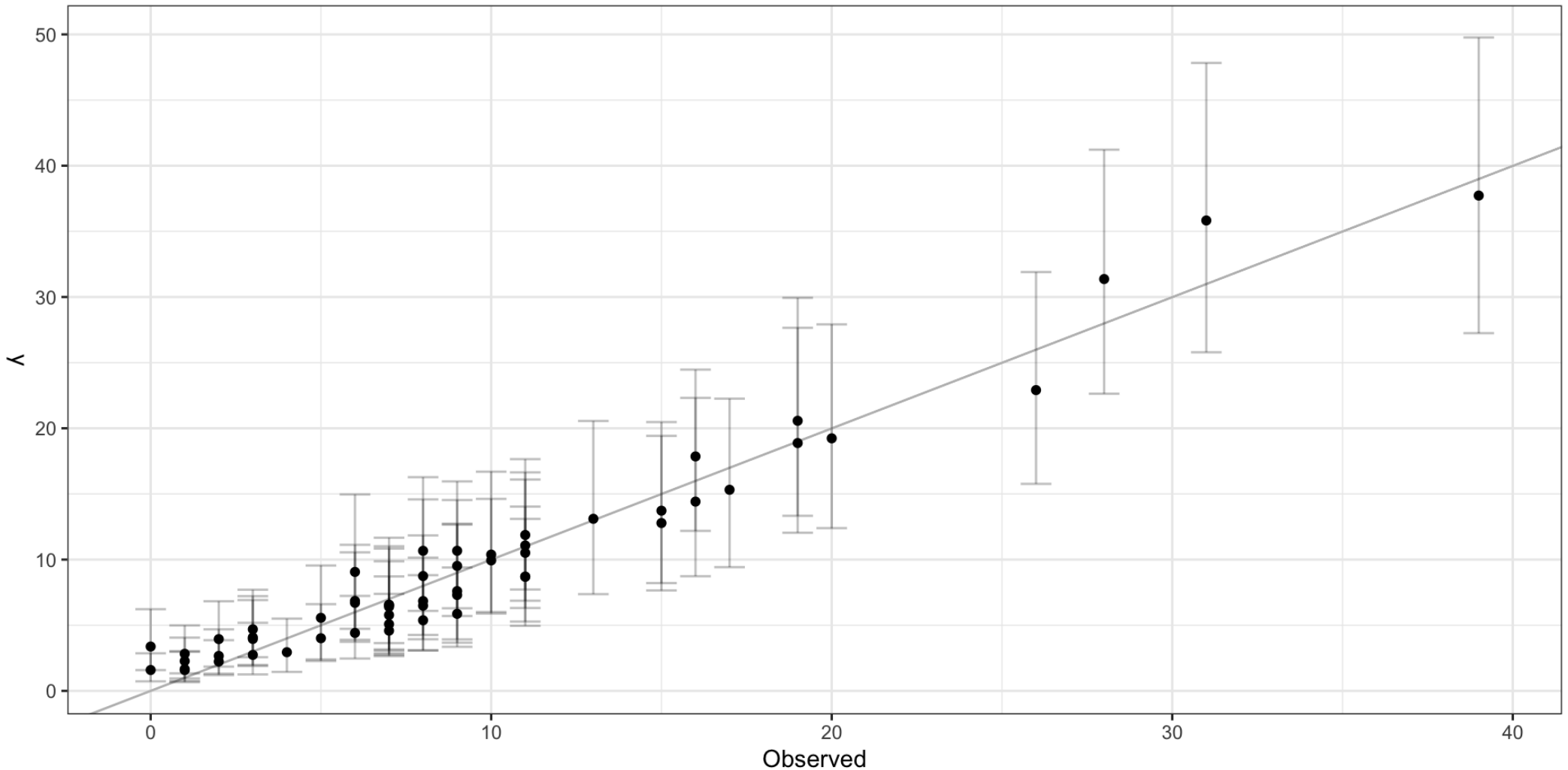
Diagnostics



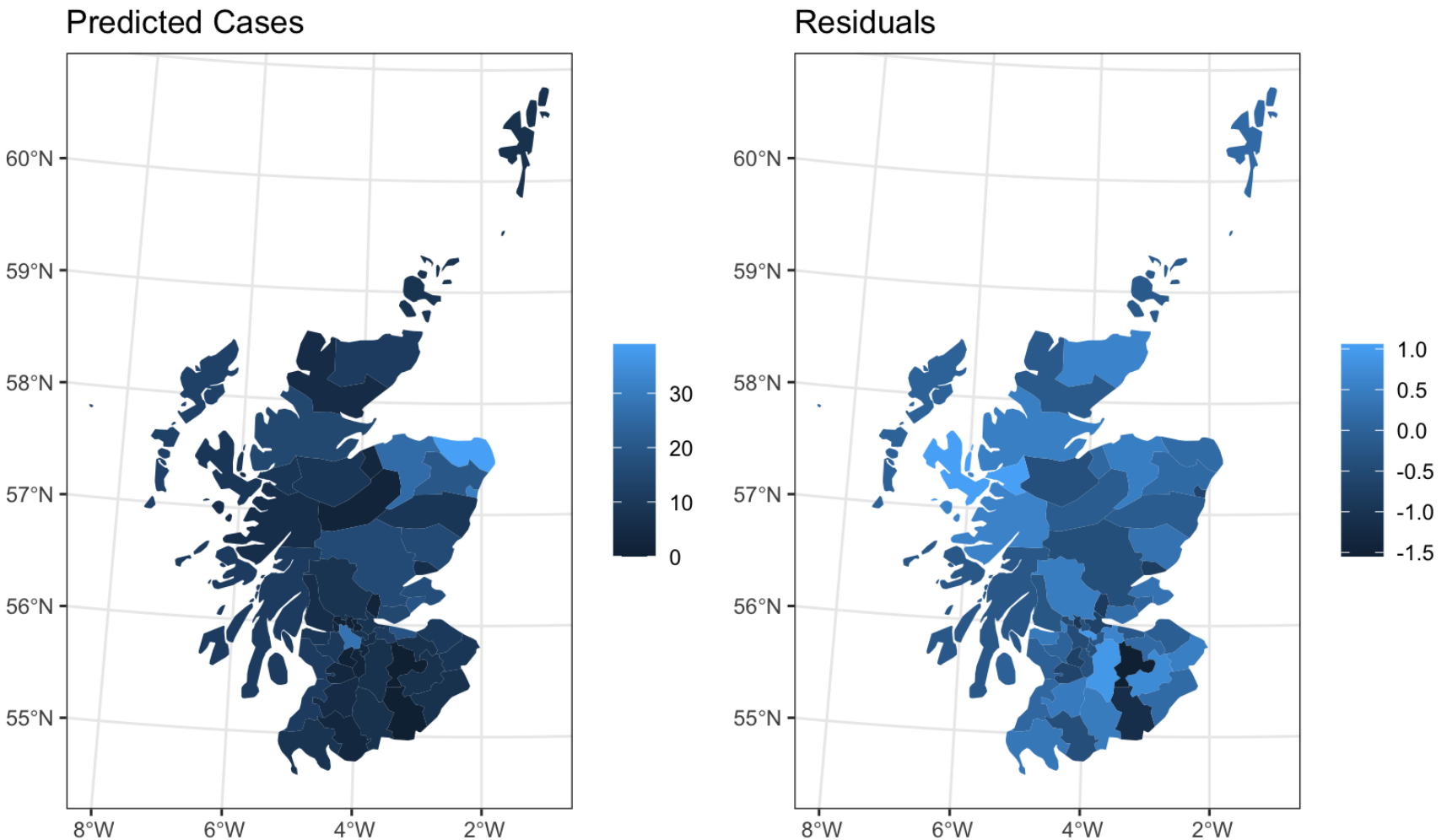
Predictions



Observed vs predicted



Residuals


Predicted 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
Residuals
60°N
59°N
58°N
57°N
56°N
55°N
8°W
6°W
4°W
2°W
1.0
0.5
0.0
-0.5
-1.0
-1.5

IAR Results

RMSE

```
1 yardstick::rmse_vec(b_iar_pred$Observed, b_iar_pred$y_pred)
```

```
[1] 1.732245
```

Moran's I

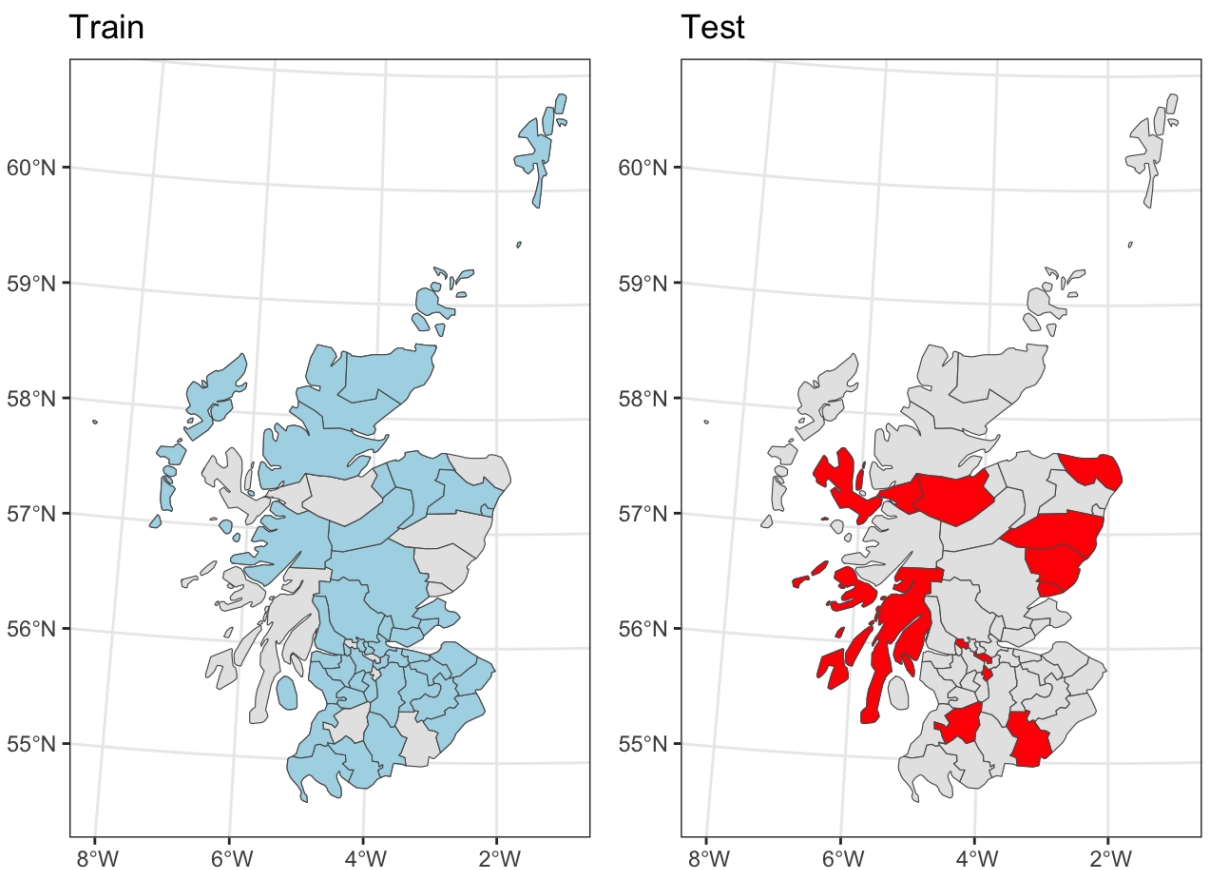
```
1 spdep::moran.test(b_iar_pred$resid, listw)
```

```
    Moran I test under randomisation

data:  b_iar_pred$resid
weights: listw

Moran I statistic standard deviate = 0.92508, p-value = 0.1775
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.058867250      -0.018181818       0.006937102
```

Out of sample predictions

Test / Train split

```
set.seed(202311091)
lip_cancer_train =
  slice_sample(lip_cancer, prop = 0.8, replace= FALSE)
lip_cancer_test = lip_cancer |>
  filter(! District %in% lip_cancer_train$District)
```



CAR Training Model

```
b_iar_train = brms::brm(
  Observed ~ offset(log(Expected)) + scale(pcaff) + car(A, gr=District,
  data=lip_cancer_train, data2=list(A=A),
  family = poisson, cores=4, iter=10000, thin = 5
)
```

Prediction

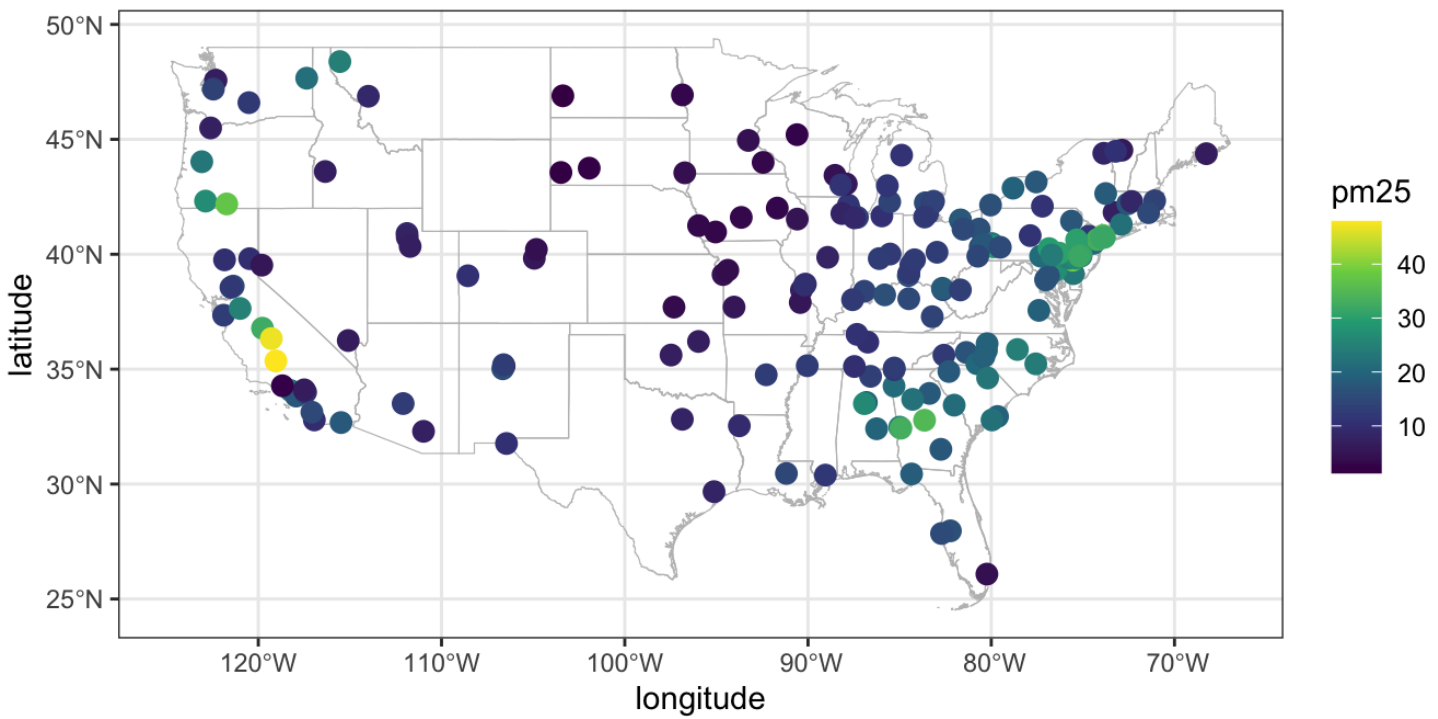
```
1 predict(b_iar_train, newdata=b_car_test)
```

```
Error in eval(expr, envir, enclos): object 'b_car_test' not found
```

Point Referenced Data

Example - PM2.5 from CSN

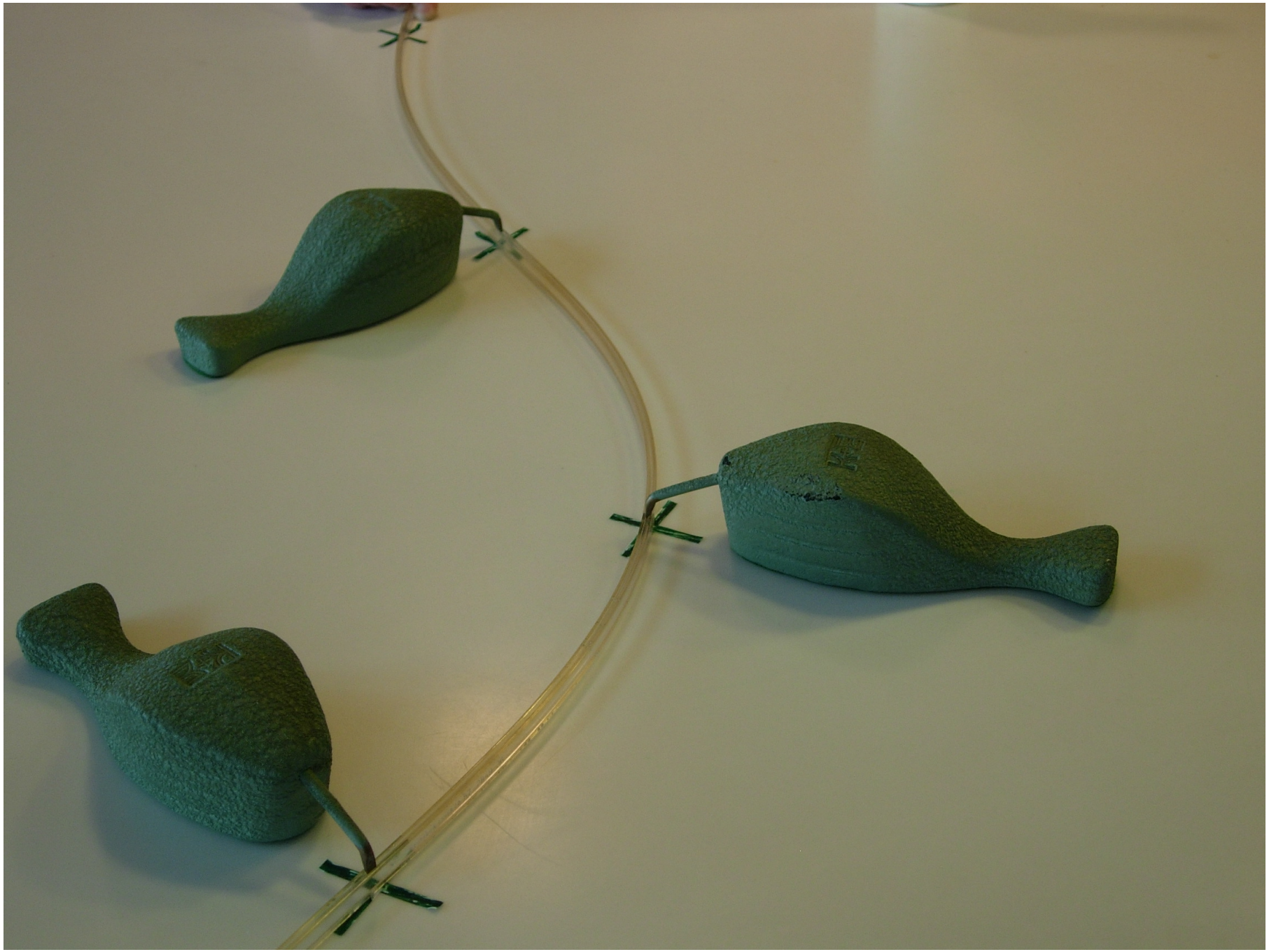
The Chemical Speciation Network are a series of air quality monitors run by EPA (221 locations in 2007). We'll look at a subset of the data from Nov 11th, 2007 (n=191) for just PM2.5.



```
1 csn
```

```
# A tibble: 191 × 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
# ℹ 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.

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

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

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 extend the smoothing spline model in two dimensions,

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

The solution to this equation has a natural representation using a weighted sum of *radial basis functions* with knots at the observed data locations ($\boldsymbol{x_i}$)

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

Prediction locations

```
r_usa = stars::st_rasterize(
  usa,
  stars::st_as_stars(st_bbox(usa),
    nx = 100, ny = 50, values=NA_real_)
)
```

```
plot(r_usa)
```

Fitting a TPS

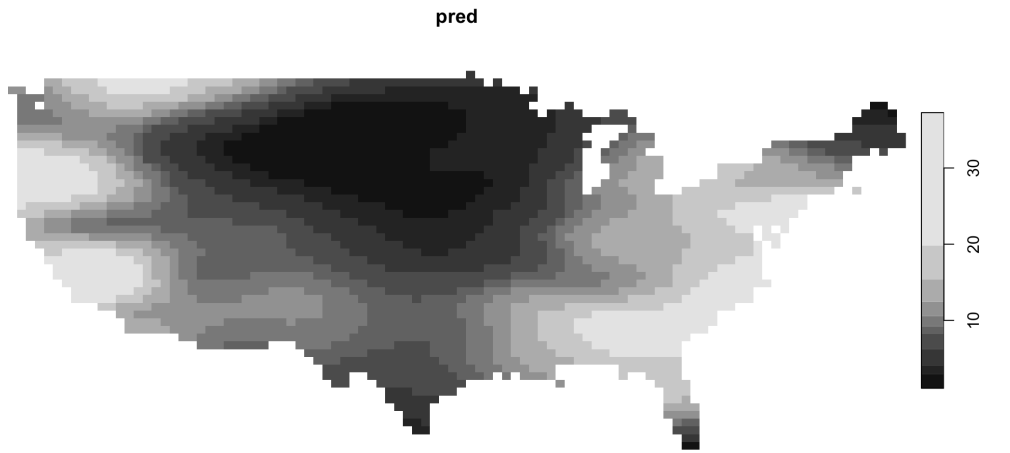
```
coords = select(csn, long=longitude, lat=latitude) |> as.matrix()
(tps = fields::Tps(x=coords, Y=csn$pm25, lon.lat=TRUE))
```

```
Call:
fields::Tps(x = coords, Y = csn$pm25, lon.lat = TRUE)

 Number of Observations:                 191
 Number of parameters in the null space 3
 Parameters for fixed spatial drift     3
 Model degrees of freedom:               64
 Residual degrees of freedom:            127
 GCV estimate for tau:                   4.461
 MLE for tau:                            4.286
 MLE for sigma:                          15.35
 lambda                                  1.2
 User supplied sigma                     NA
 User supplied tau^2                     NA
Summary of estimates:
              lambda      trA      GCV   tauHat -lnLike Prof converge
GCV         1.196496 63.97784 29.91791 4.460553     612.4247         5
GCV.model         NA       NA       NA       NA           NA        NA
GCV.one     1.196496 63.97784 29.91791 4.460553           NA         5
RMSE              NA       NA       NA       NA           NA        NA
pure error        NA       NA       NA       NA           NA        NA
```

Predictions

```
pred = r_usa |>
  as_tibble() |>
  filter(!is.na(ID)) |>
  select(long = x, lat = y)

tps_pred = pred |>
  mutate(
    pred = predict(tps, cbind(long, lat))
  ) |>
  stars::st_as_stars()
```

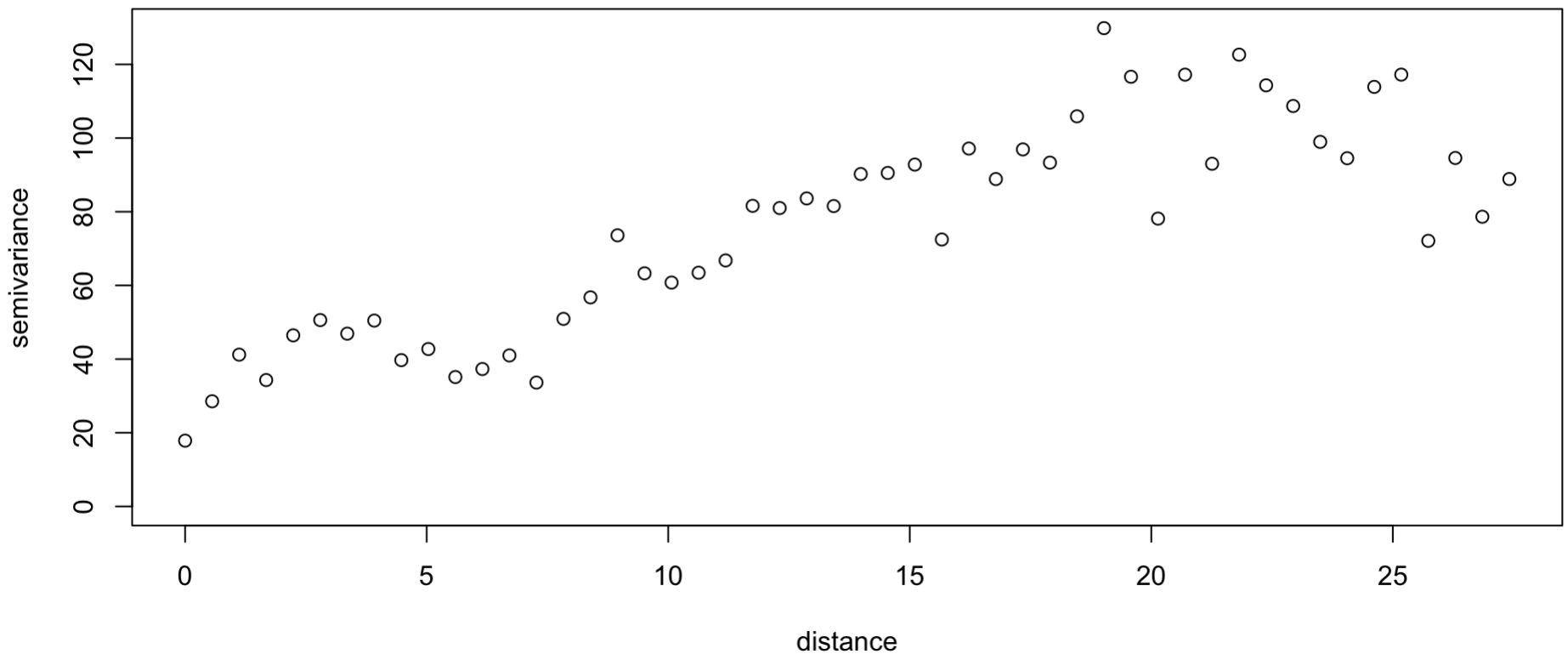
```
plot(tps_pred)
```



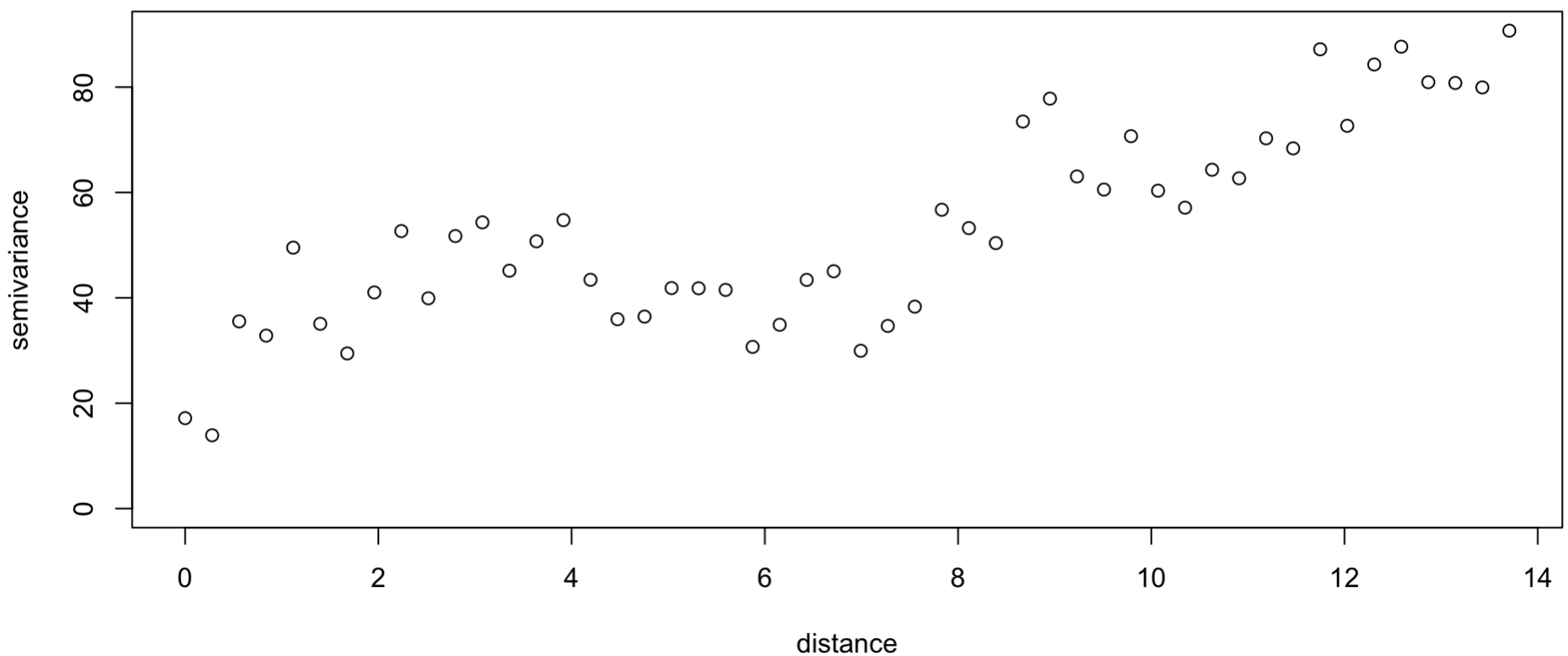
Gaussin Process Models / Kriging

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()
```

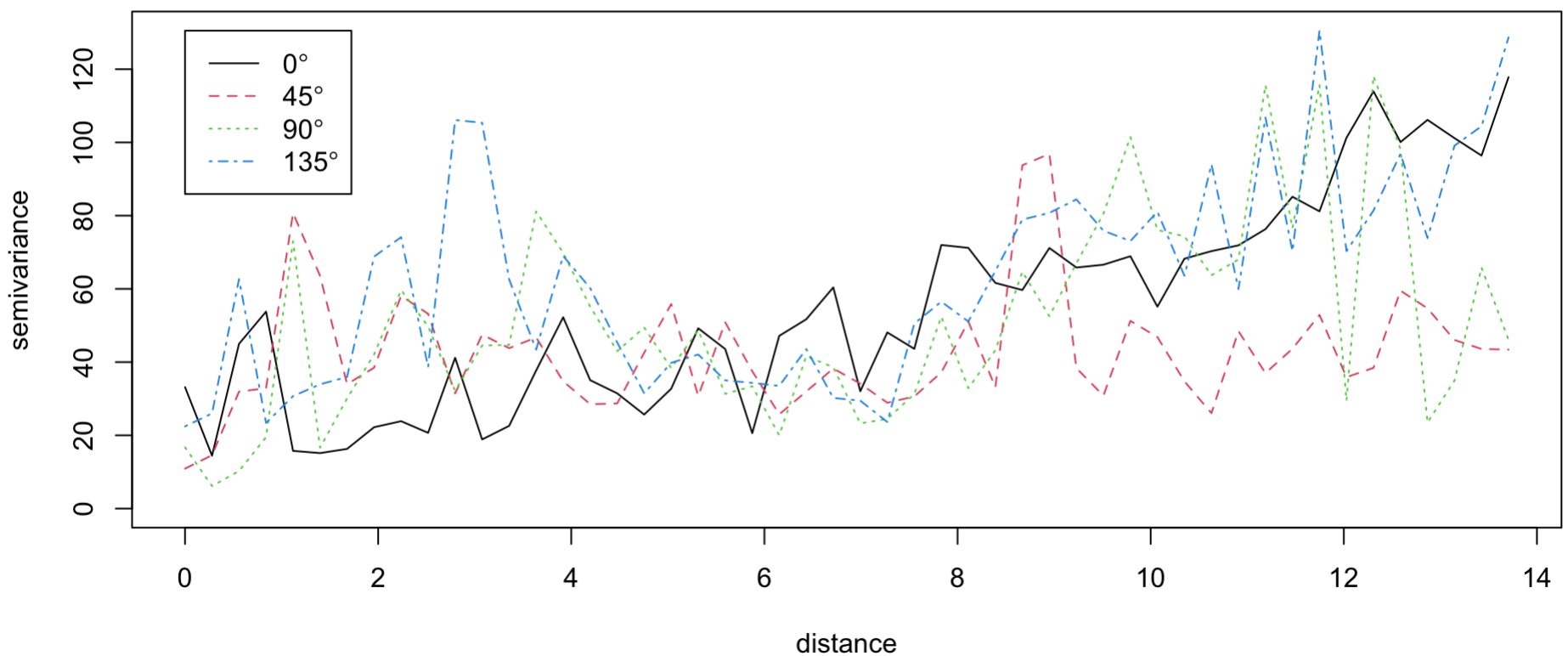

semivariance
0
20
40
60
80
100
120
0
5
10
15
20
25
distance

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



Isotropy / Anisotropy

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



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.

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

$$\{\Sigma\}_{ij} = \sigma^2 \exp(-l \left\| \boldsymbol{s}_i - \boldsymbol{s}_j \right\|) + \sigma_n^2 \, 1_{i=j}$$

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

$$y(\boldsymbol{s}) = \mu(\boldsymbol{s}) + w(\boldsymbol{s}) + \epsilon(\boldsymbol{s}) w(\boldsymbol{s}) \sim N(0, \Sigma') \epsilon(\boldsymbol{s}_i) \sim N(0, \sigma_n^2) \{\Sigma'\}_{ij} = \sigma^2 \, \mathrm{e}$$

Fitting with `gplm()`

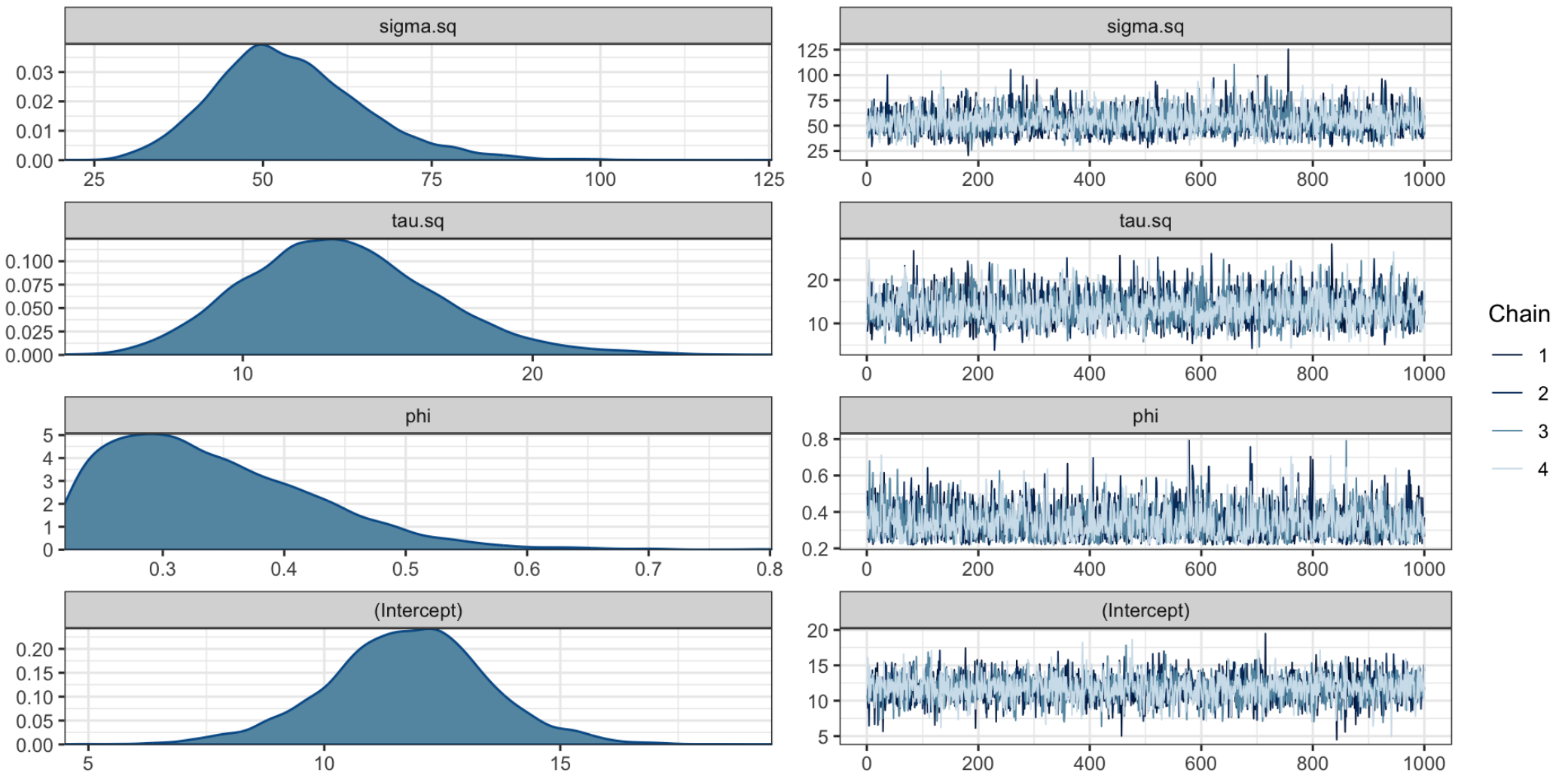
```
max_range = max(dist(csn[,c("longitude", "latitude")])) / 4

m = gplm(
  pm25~1, data = csn, coords=c("longitude", "latitude"),
  cov_model = "exponential",
  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(
    phi.Unif = c(3/max_range, 3/(0.5)),
    sigma.sq.IG = c(2, 2),
    tau.sq.IG = c(2, 2)
  ),
  thin=10
)
```

```
1 m
```

```
# A gplm model (spBayes spLM) with 4 chains, 4 variables, and 4000 iterations.
# 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 sigma.sq    54.0   52.9   11.3    10.4    37.7   74.0    1.00    2952.    3515.
2 tau.sq      13.4   13.2    3.33    3.20    8.28  19.2    1.00    2932.    3437.
3 phi          0.341  0.325  0.0848  0.0835  0.232  0.495  1.00    2776.    2836.
4 (Intercept) 11.8   11.8    1.71    1.59    8.87  14.5    1.00    4125.    3879.
```

Parameter values

```
1 plot(m)
```



Predictions - posterior draws

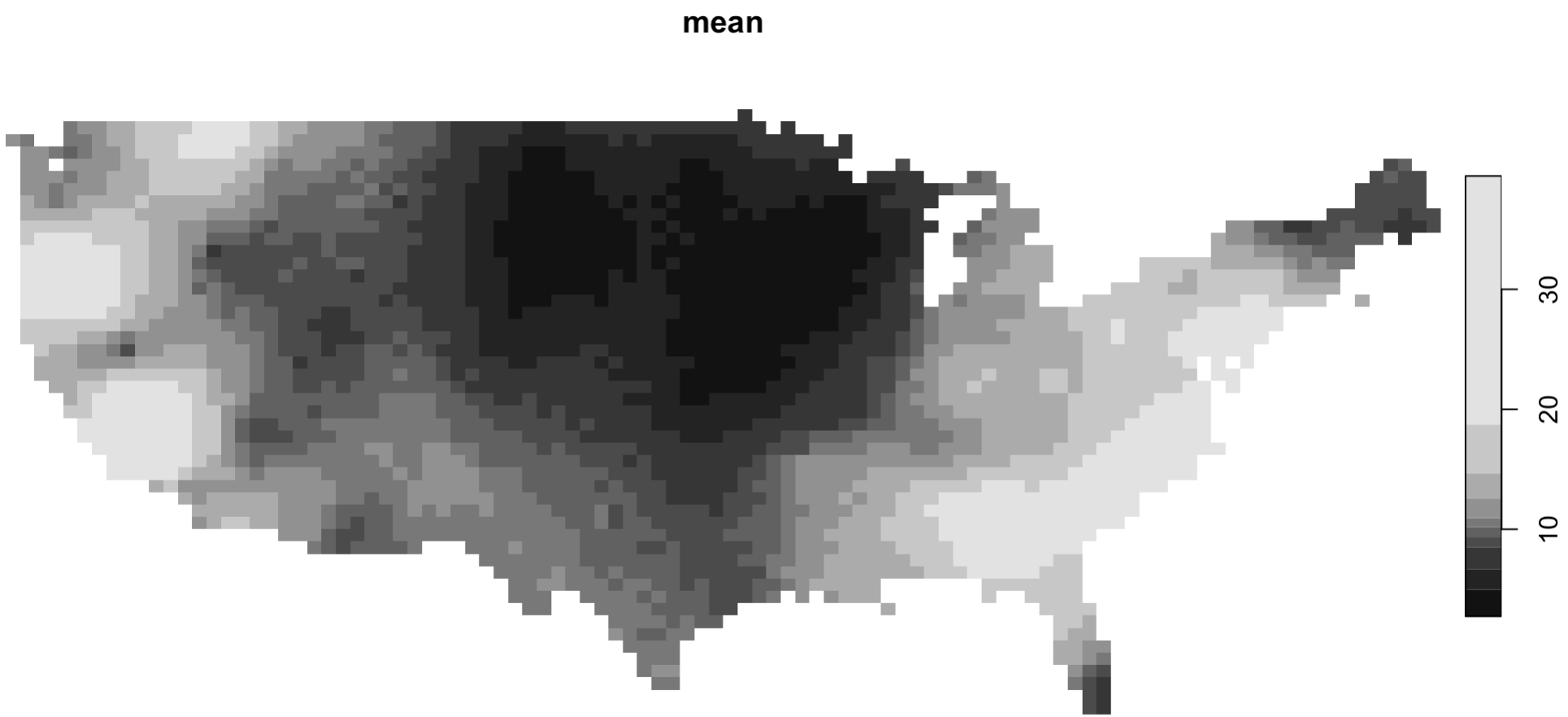
```r
(p = predict(m, newdata=pred, coords=c("longitude", "latitude")))
```

```
# A draws_matrix: 1000 iterations, 4 chains, and 2828 variables
    variable
draw  y[1]   y[2] y[3] y[4] y[5]  y[6] y[7] y[8]
  1  14.03 -4.073 15.0  4.8 -8.8  7.84   21  4.9
  2  11.71  0.052 10.2  5.8 11.3 14.58   20 10.7
  3  -3.37 17.307 18.4 20.2 23.7 28.46    9 20.4
  4   7.31  2.500  4.6  7.3 23.7 14.63   15 11.8
  5   0.47 10.014 10.4 17.2 14.6 11.17   10 10.0
  6   7.57 11.004 10.6  9.2 10.6 14.56   23 10.0
  7   7.16  6.791 12.8  5.0 22.4  0.88   16 20.1
  8  16.54  9.611  1.8 23.9 23.9 19.23   38 10.0
  9  16.03  3.135 23.7  1.1 12.4 13.10   34 20.1
  10 14.14  0.638 13.7  8.7 -4.9 11.37   18 13.5
# ... with 3990 more draws, and 2820 more variables
```

Predictions - raster

```
gp_pred = left_join(
  pred |>
    mutate(i = row_number()),
  tidybayes::gather_draws(p, y[i]) |>
    filter(.chain == 1) |>
    group_by(.chain, i) |>
    summarize(
      mean = mean(.value),
      med = median(.value),
      sd = sd(.value),
      .groups = "drop"
    ),
  by = "i"
)

gp_pred |>
  stars::st_as_stars() |>
  select(mean) |>
  plot()
```

Predictions - raster



```
gp_pred |>
  stars::st_as_stars() |>
  select(sd) |>
  plot()
```