

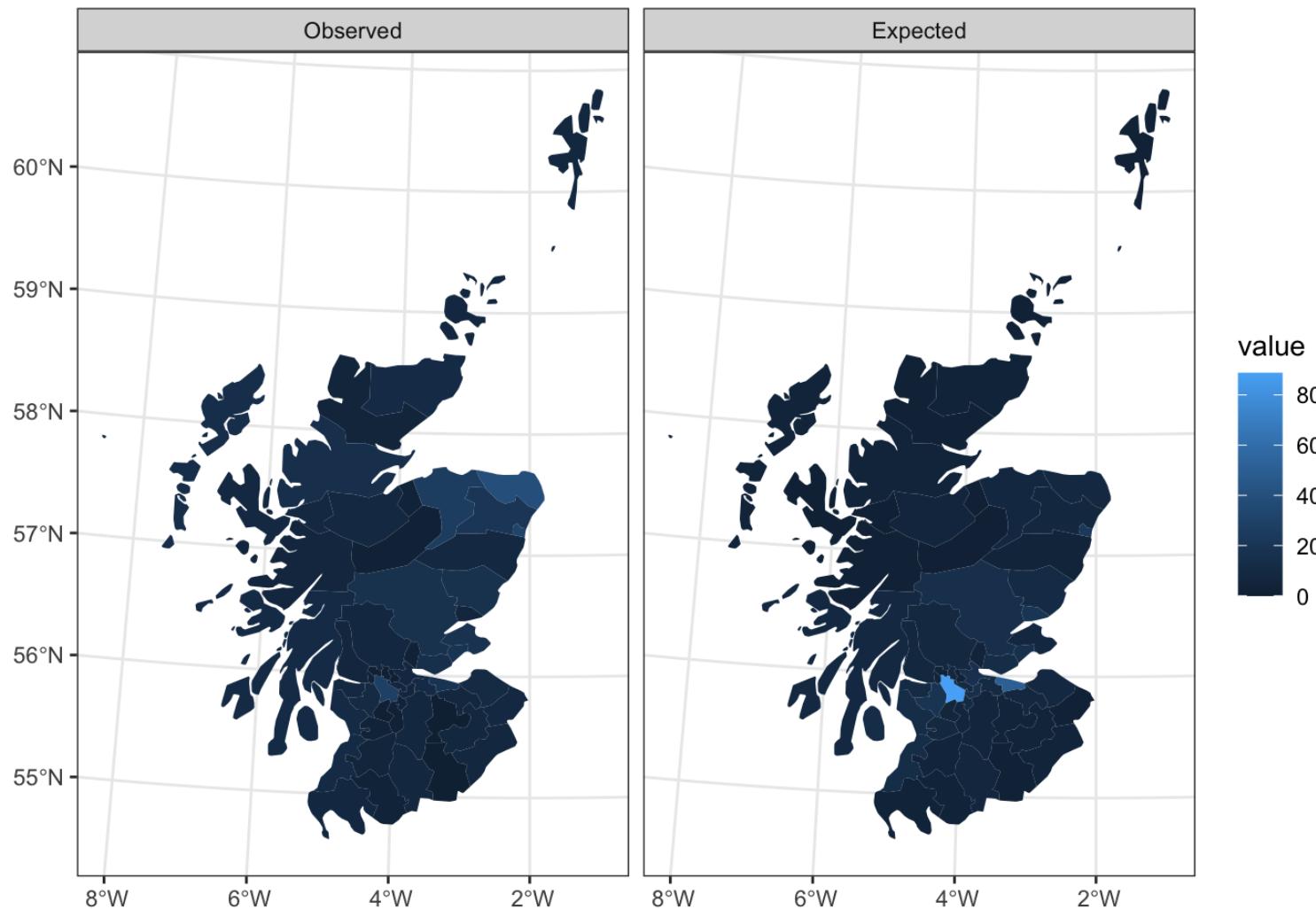
Spatial GLM + Point Reference Spatial Data

Lecture 20

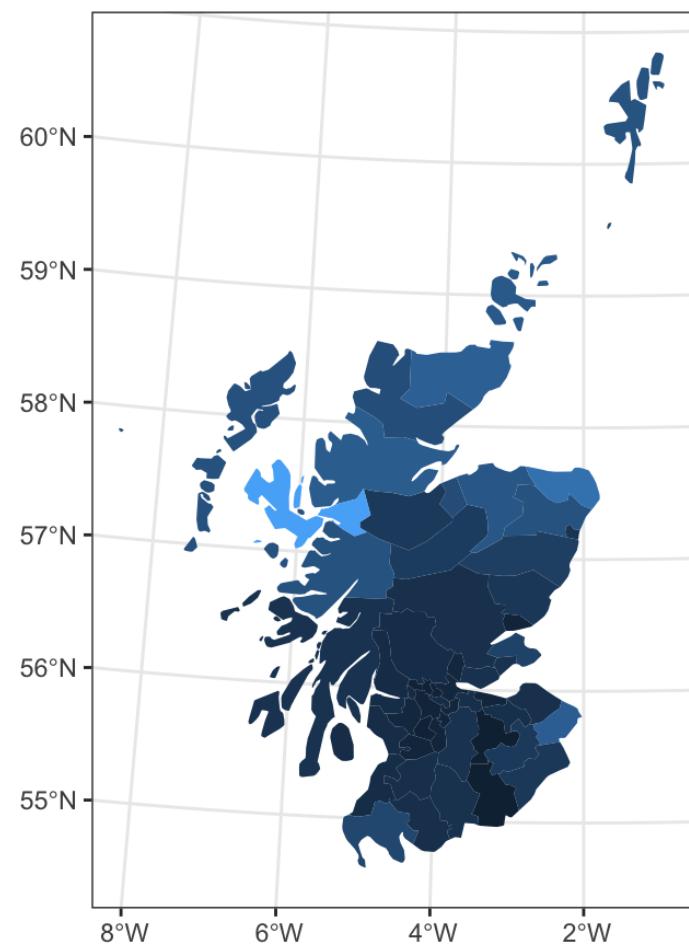
Dr. Colin Rundel

Spatial GLM Models

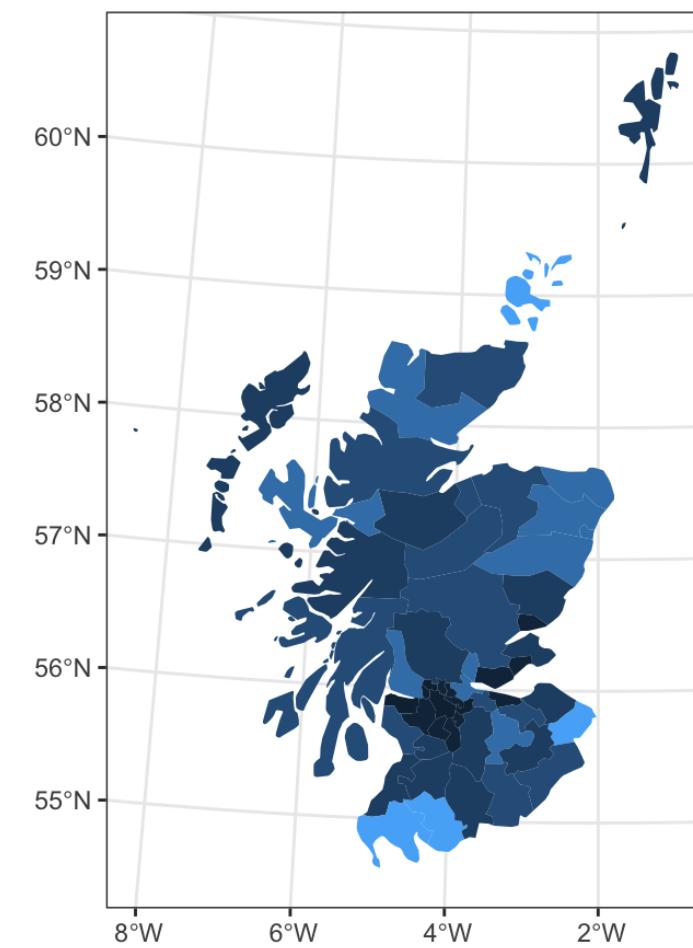
Scottish Lip Cancer Data



Obs/Exp

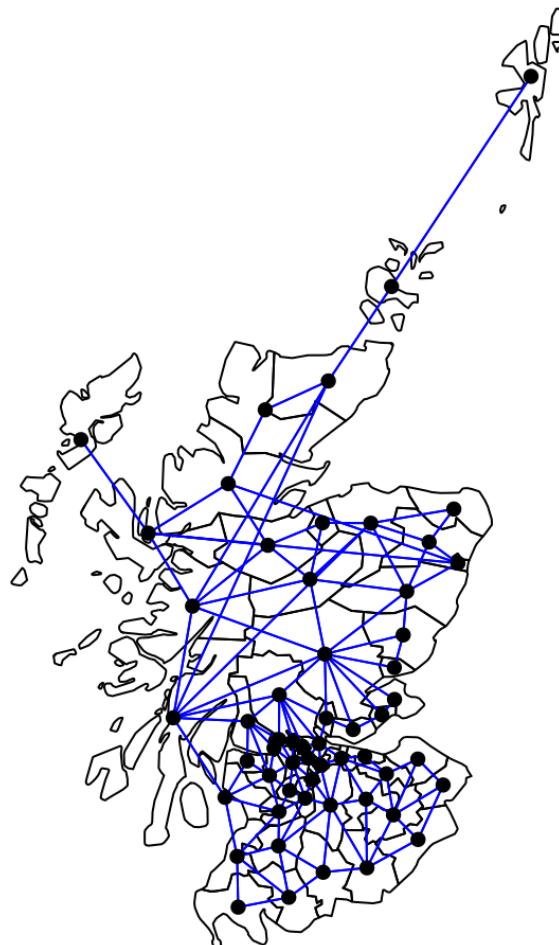


% Agg Fish Forest



Neighborhood / weight matrix

```
1 A = (st_distance(lip_cancer) |> unclass() < 1e-6; diag(A) = 0  
2 rownames(A) = lip_cancer$District; colnames(A) = lip_cancer$District  
3 listw = spdep::mat2listw(A, style="B")
```



Moran's I

```
1 spdep::moran.test(  
2   lip_cancer$Observed,  
3   listw  
4 )
```

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
```

```
1 spdep::moran.test(  
2   lip_cancer$Observed / lip_cancer$Expected,  
3   listw  
4 )
```

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

```
1 l = glm(Observed ~ offset(log(Expected)) + pcaff, family="poisson", data=lip_cancer)
2 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

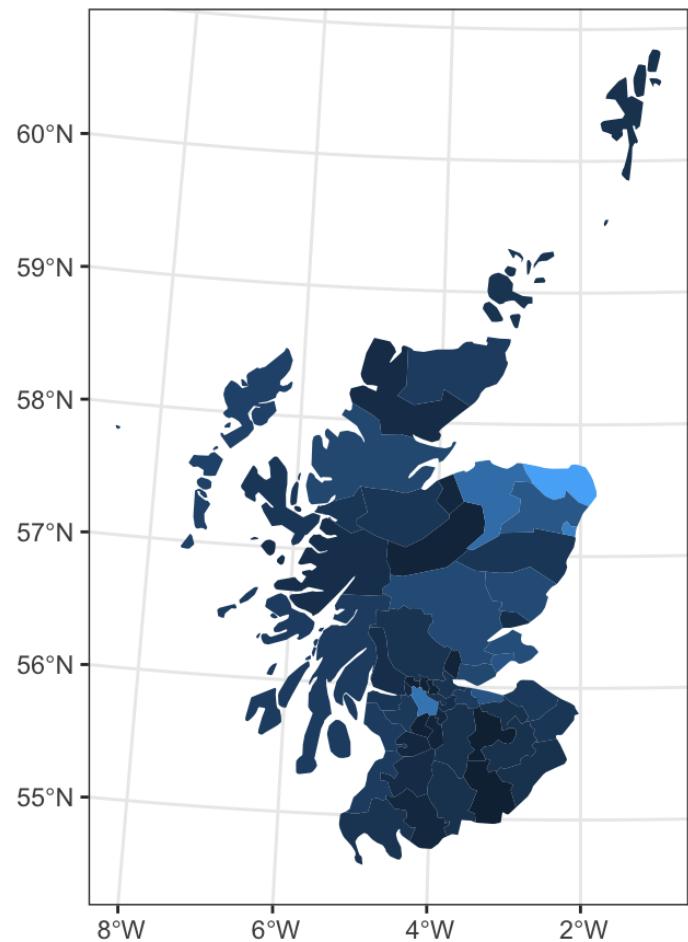
Residual deviance: 238.62 on 54 degrees of freedom

AIC: 450.6

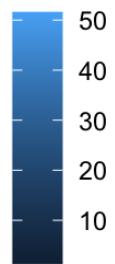
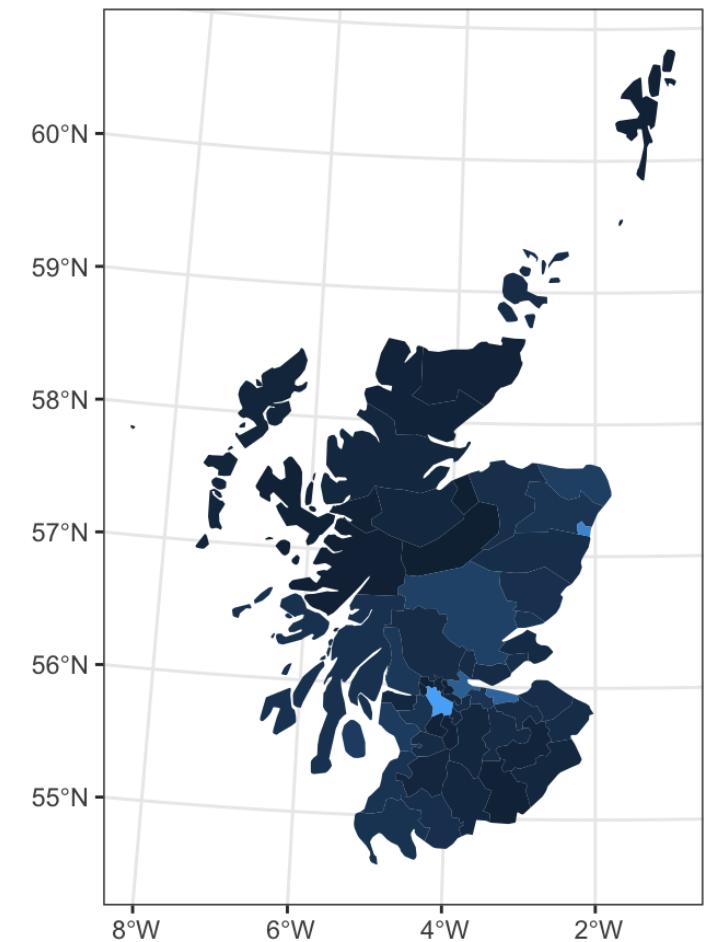
Number of Fisher Scoring iterations: 5

GLM Fit

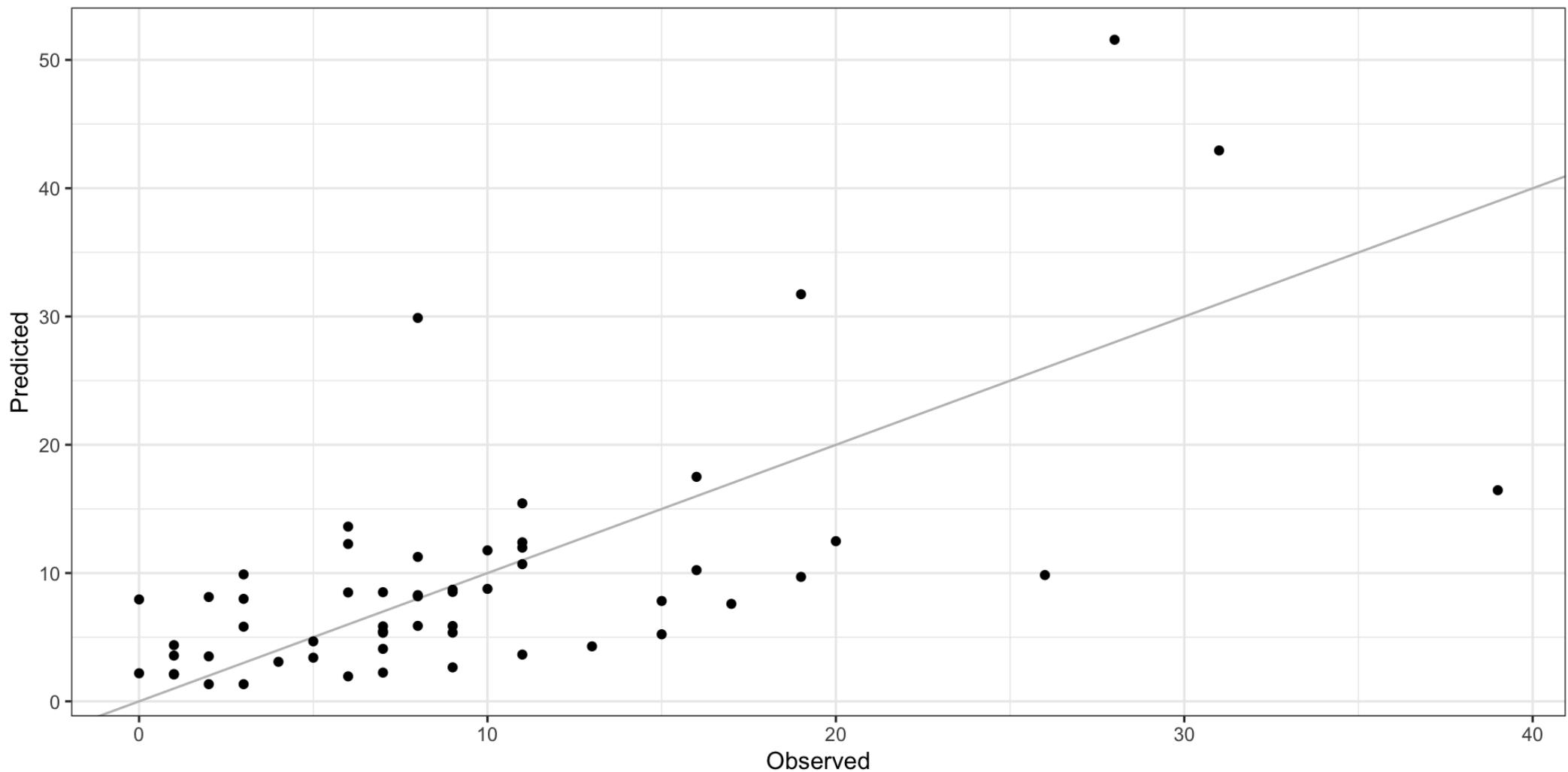
Observed Cases



GLM Predicted Cases

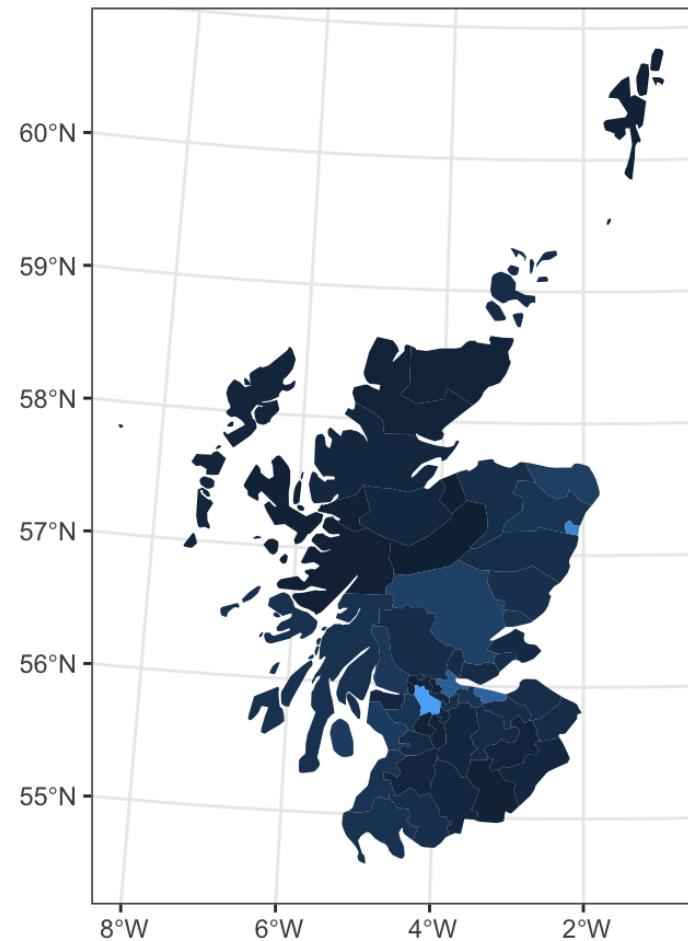


GLM Fit

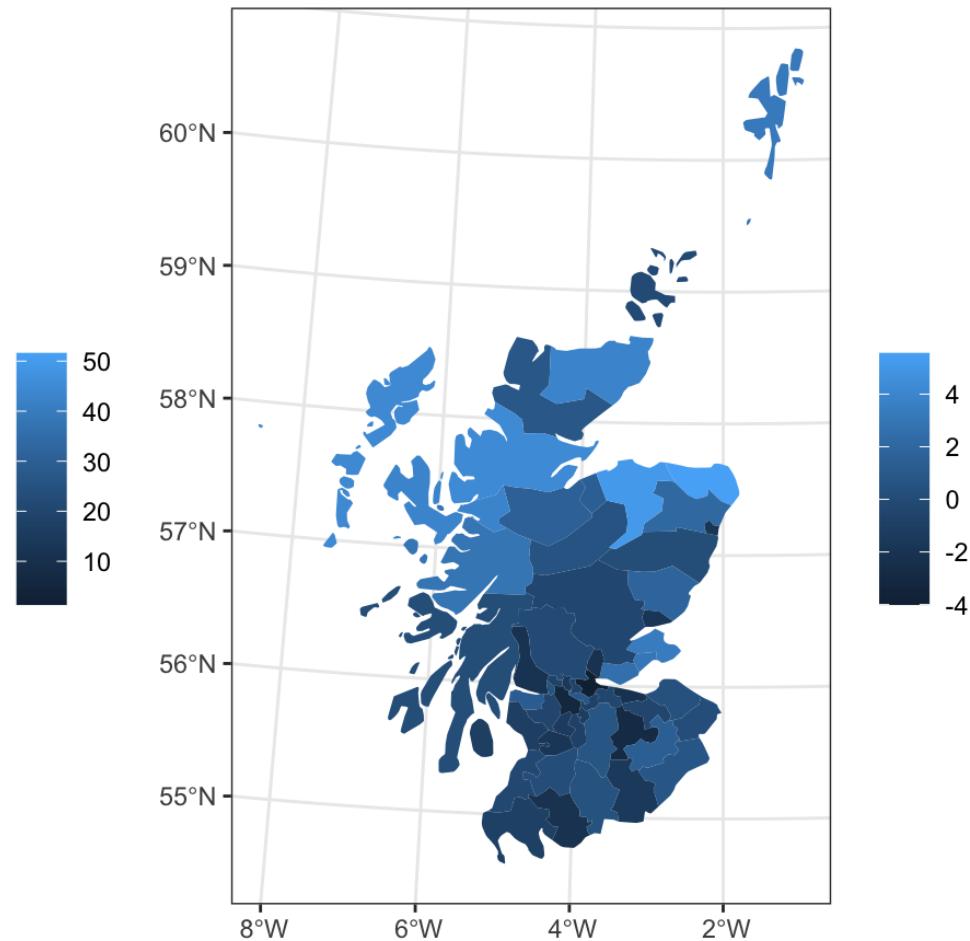


GLM Residuals

GLM Predicted Cases



GLM Pearson 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 hierarchical 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 .

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

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

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

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

CAR model

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

```
1 b_car
```

Family: poisson
Links: mu = log
Formula: Observed ~ offset(log(Expected)) + 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	865	836
sdcar	0.83	0.14	0.60	1.14	1.00	3134	3479

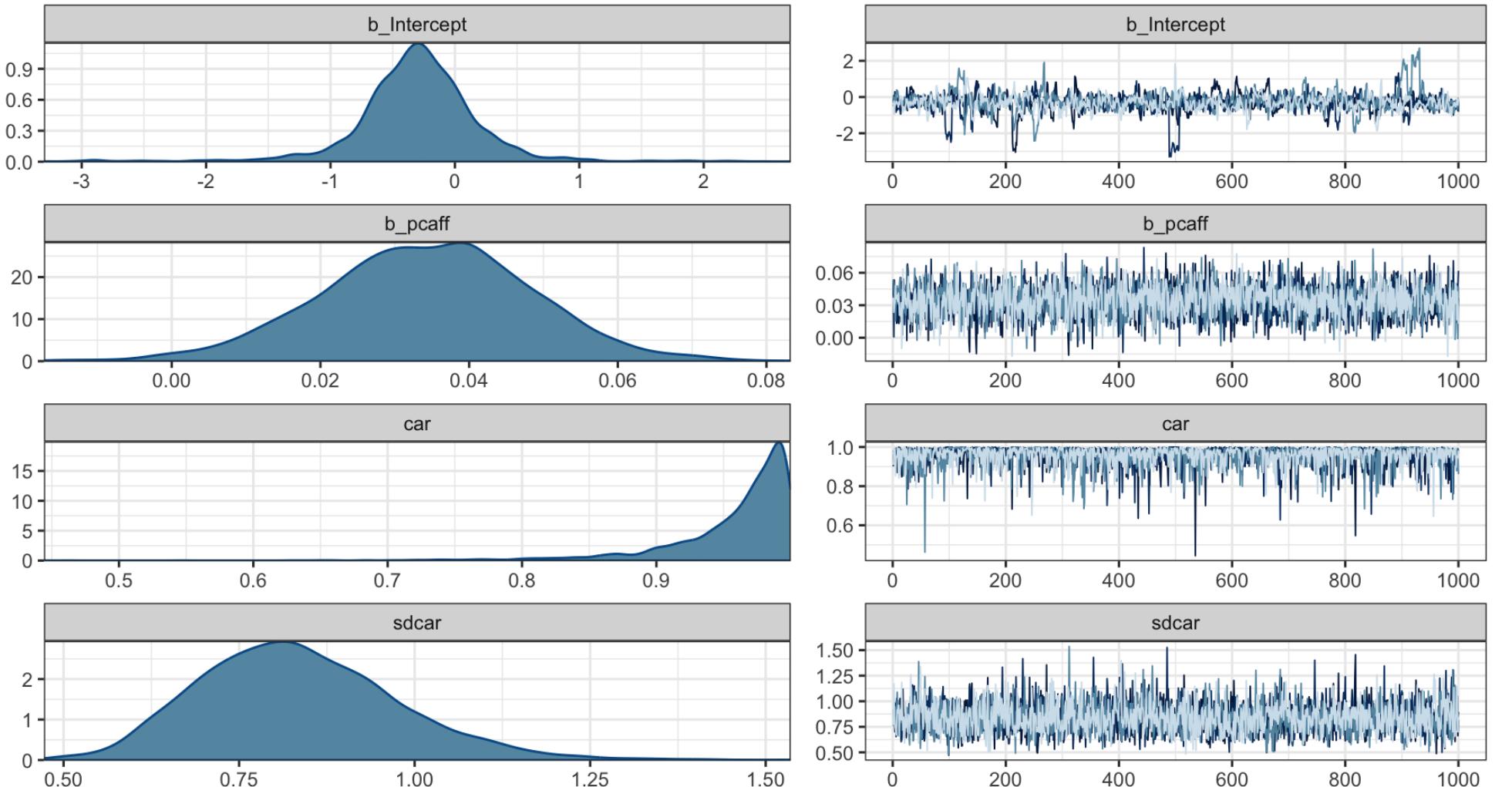
Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-0.30	0.52	-1.34	0.65	1.01	619	449
pcaff	0.03	0.01	0.01	0.06	1.00	3435	3892

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).

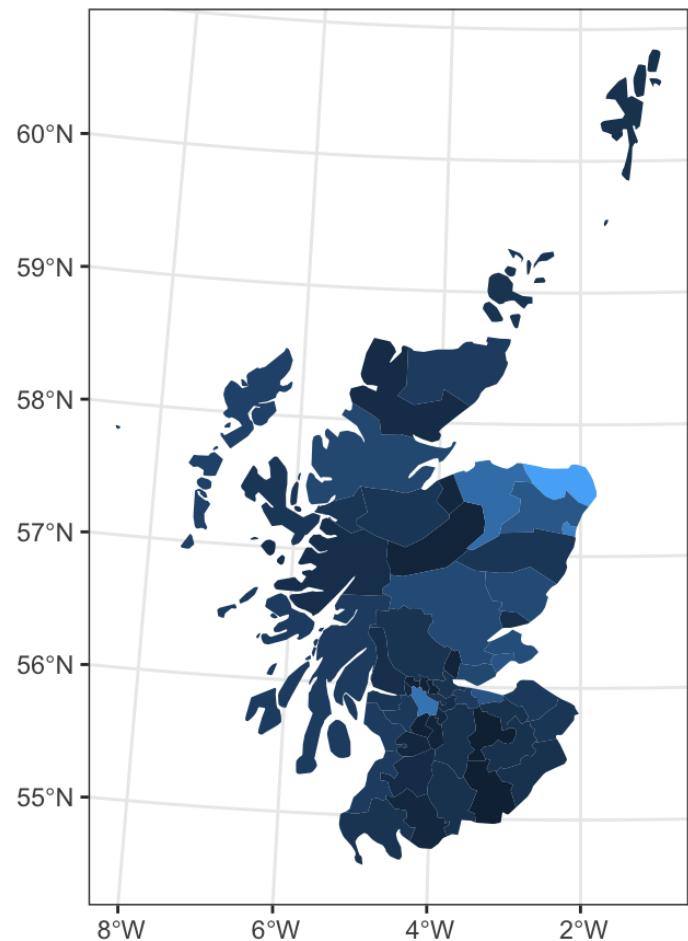
Diagnostics

```
1 plot(b_car)
```

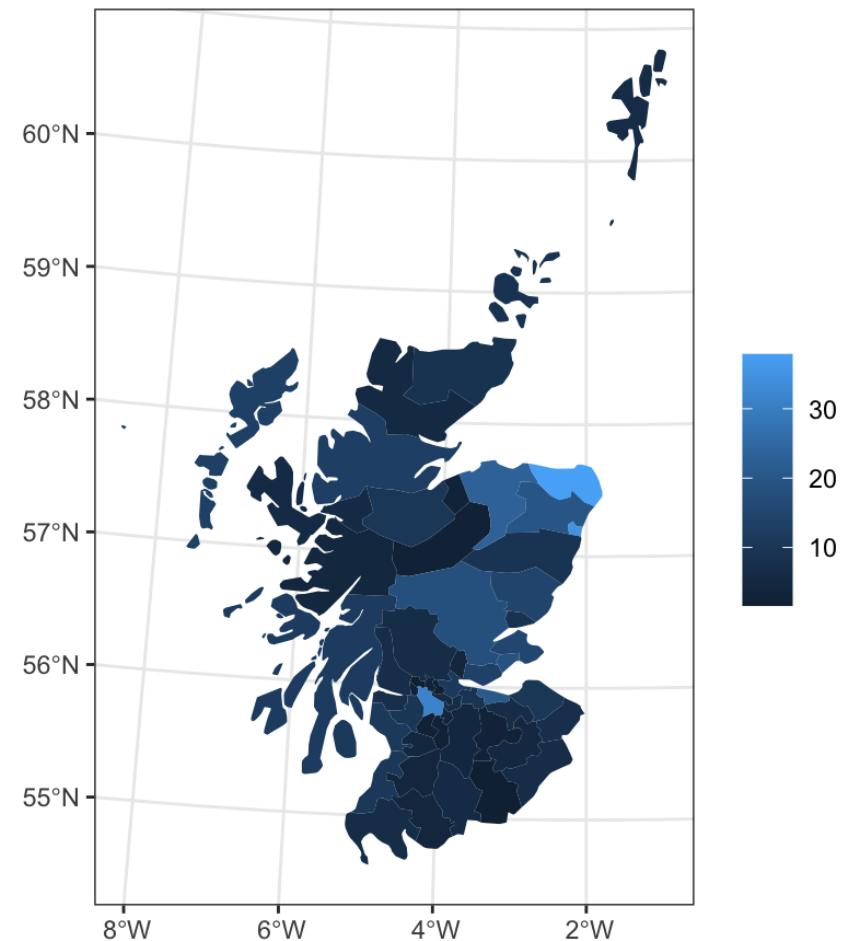


Predictions

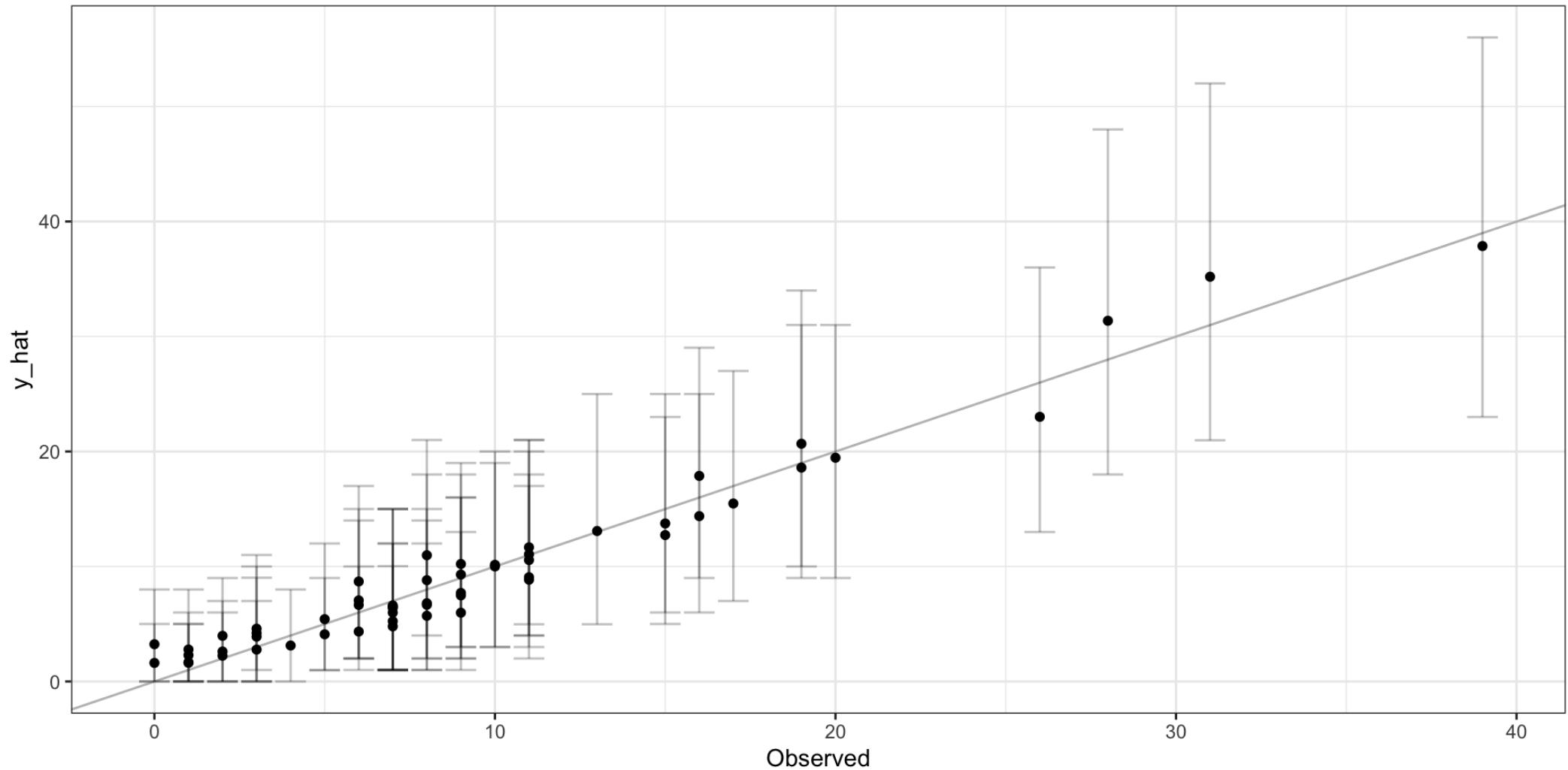
Observed Cases



Predicted Cases

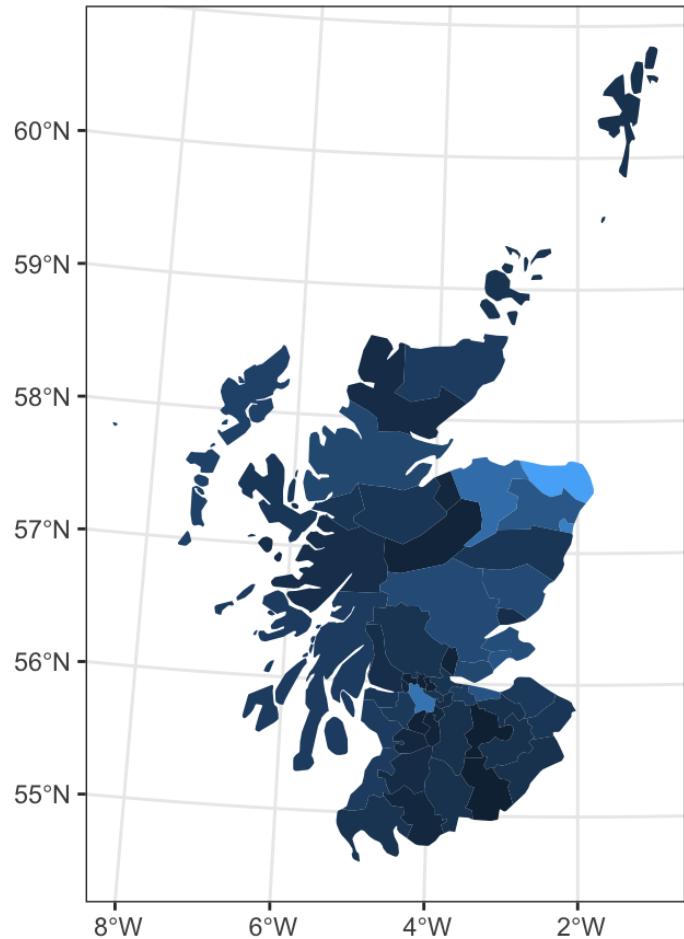


Observed vs predicted

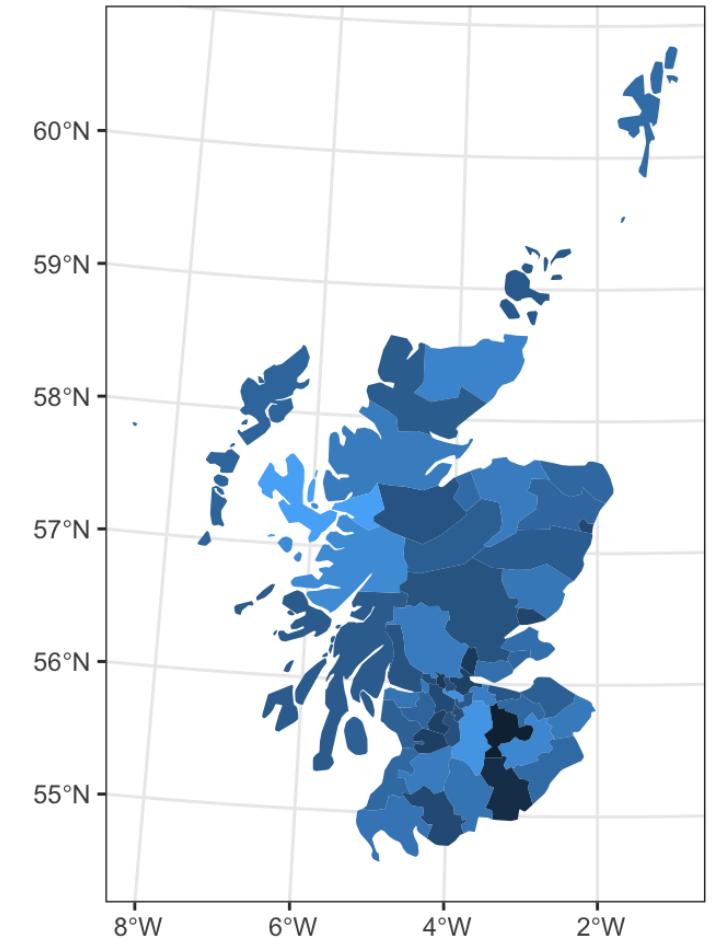


Residuals

Predicted Cases



Residuals



Results

RMSE

```
1 yardstick::rmse_vec(b_car_pred$Observed, b_car_pred$y_pred)
[1] 1.63801
```

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 = 1.5716, p-value = 0.05802
alternative hypothesis: greater
sample estimates:
Moran I statistic      Expectation      Variance
0.112297225     -0.018181818     0.006892491
```

IAR Model

Intrinsic Autoregressive Model (IAR)

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

```
1 b_iar
```

Family: poisson
Links: mu = log
Formula: Observed ~ offset(log(Expected)) + 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:

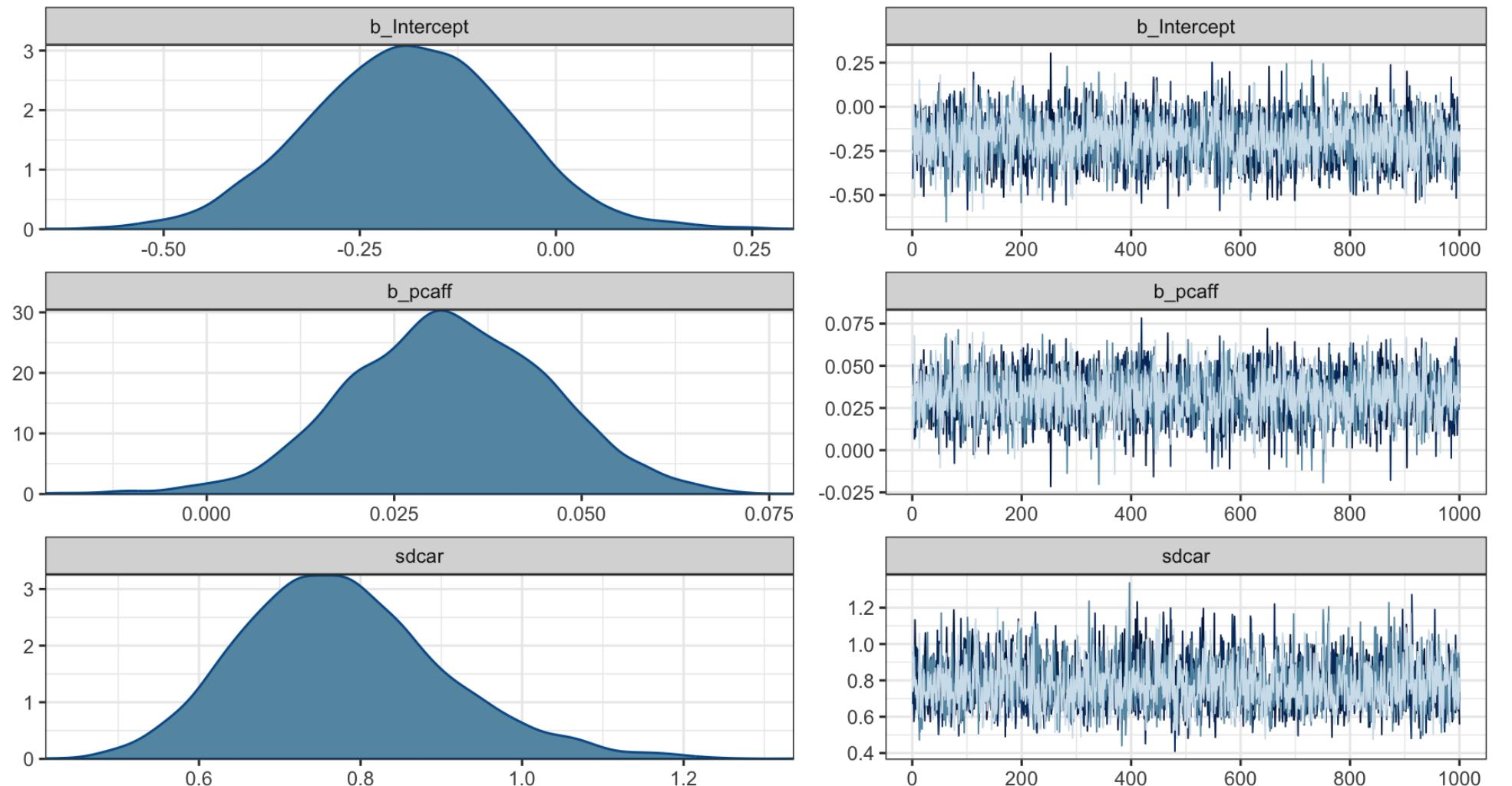
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sdcar	0.78	0.13	0.56	1.06	1.00	3047	3353

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-0.19	0.13	-0.43	0.06	1.00	3793	3998
pcaff	0.03	0.01	0.01	0.06	1.00	3764	3625

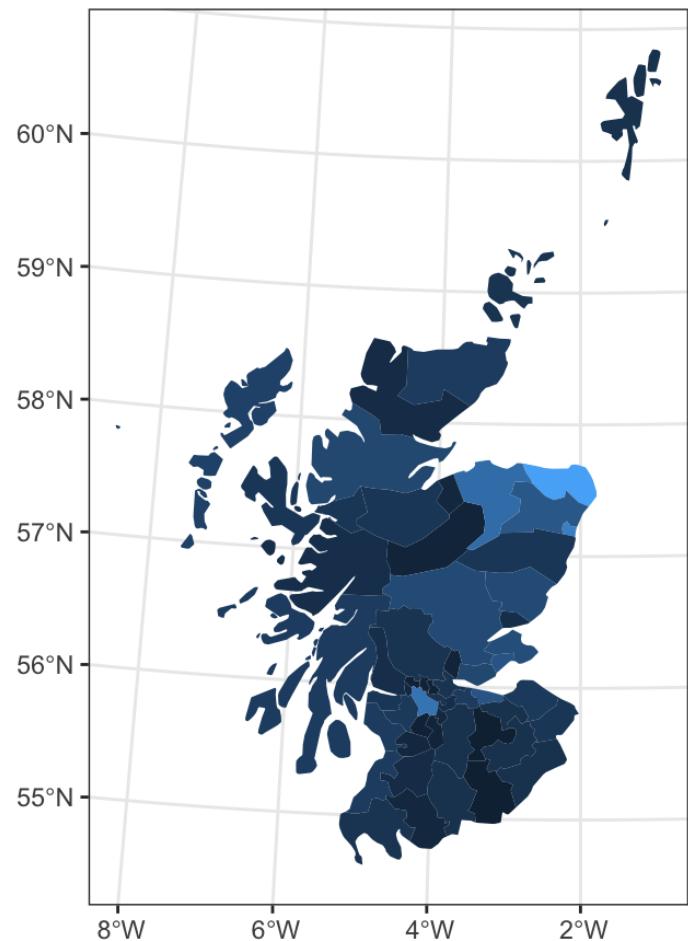
Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS and Tail_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Diagnostics

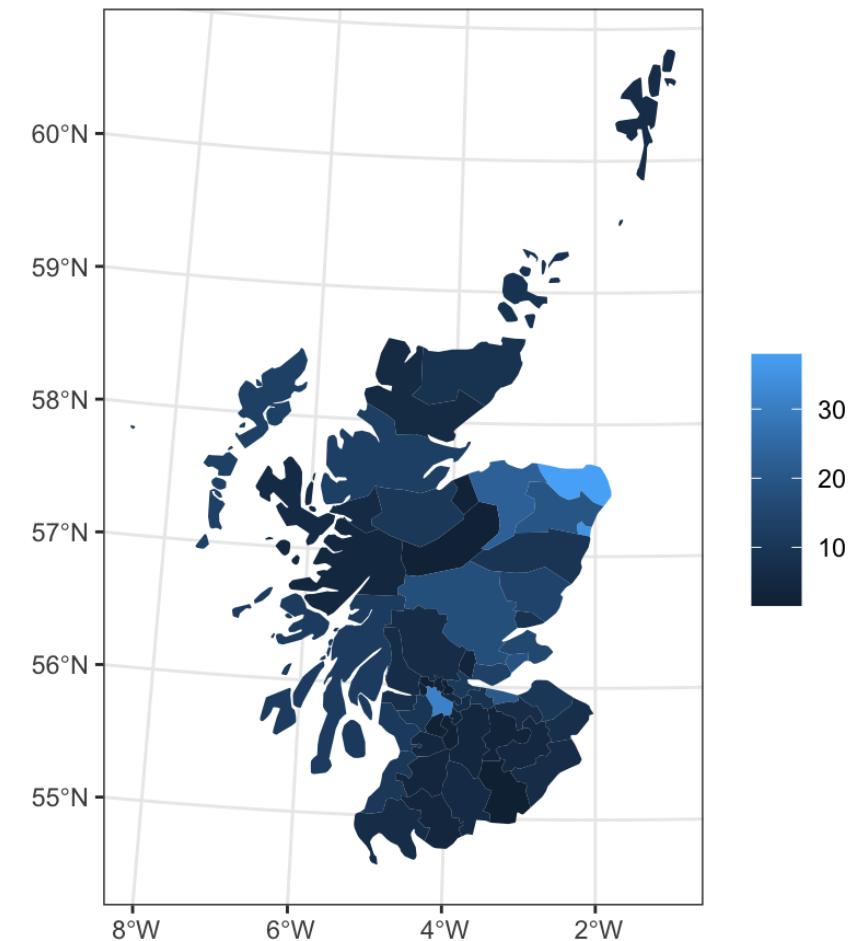


Predictions

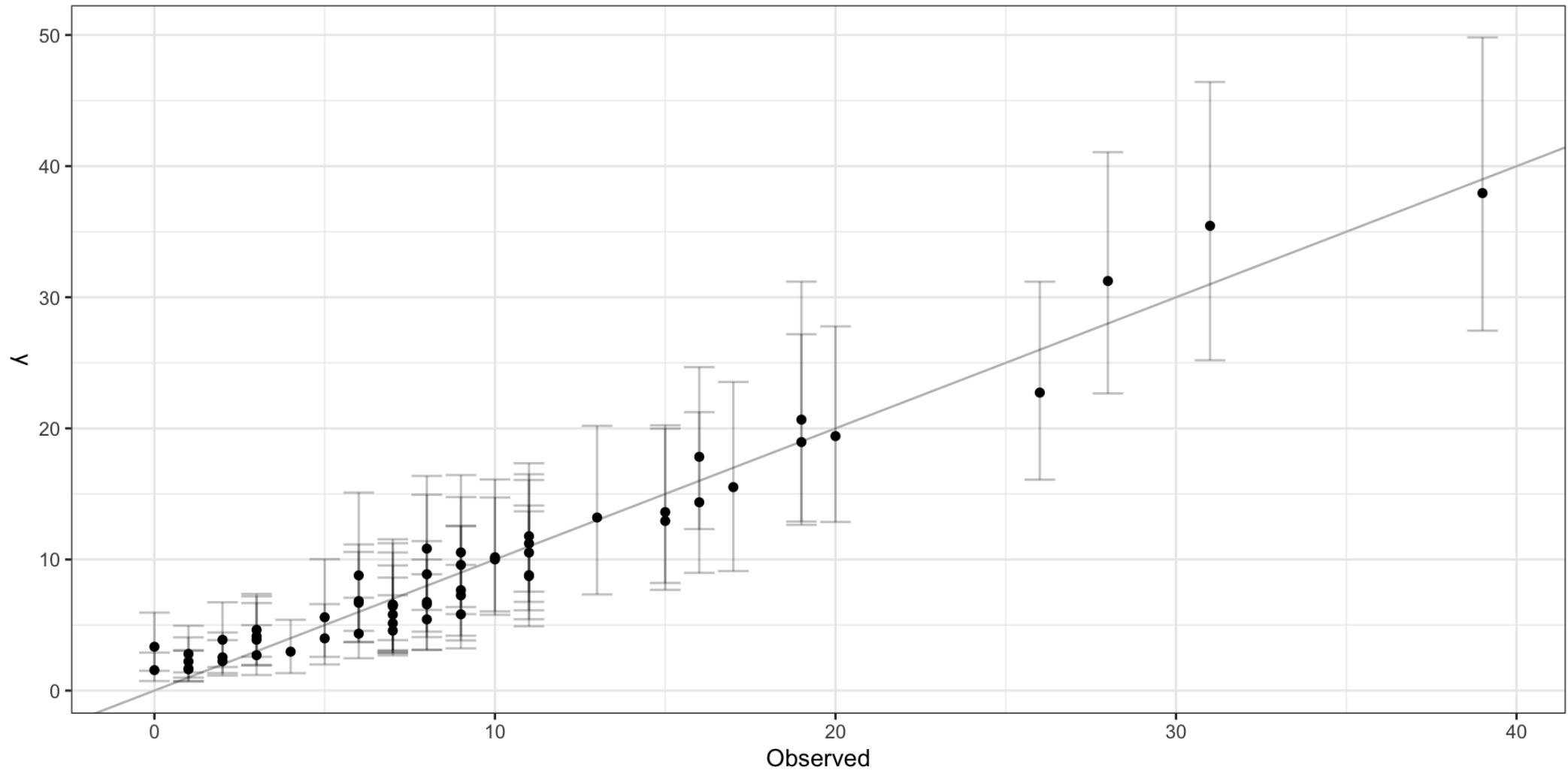
Observed Cases



Predicted Cases

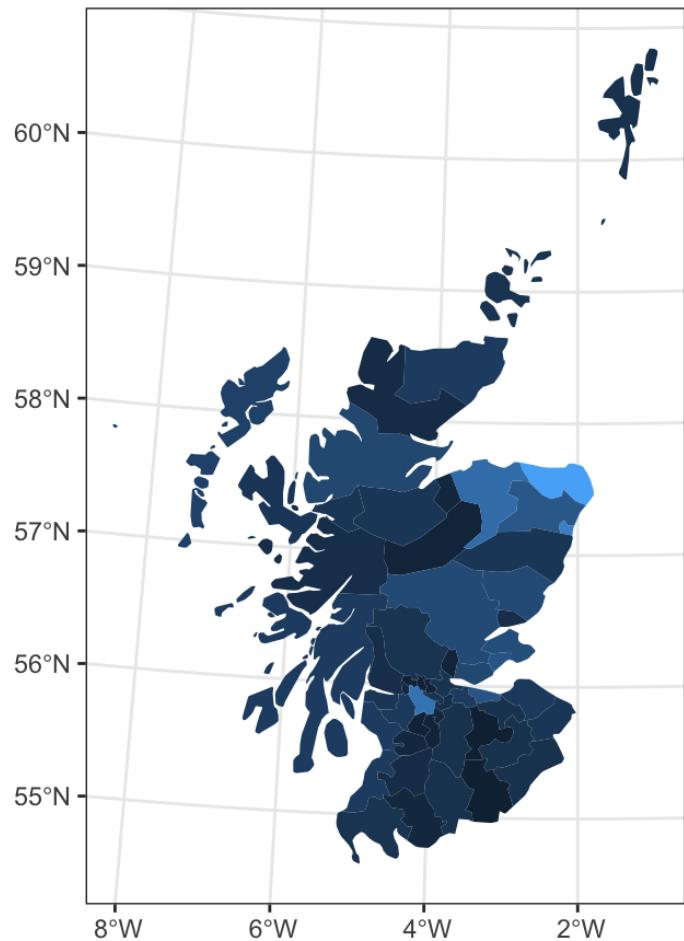


Observed vs predicted

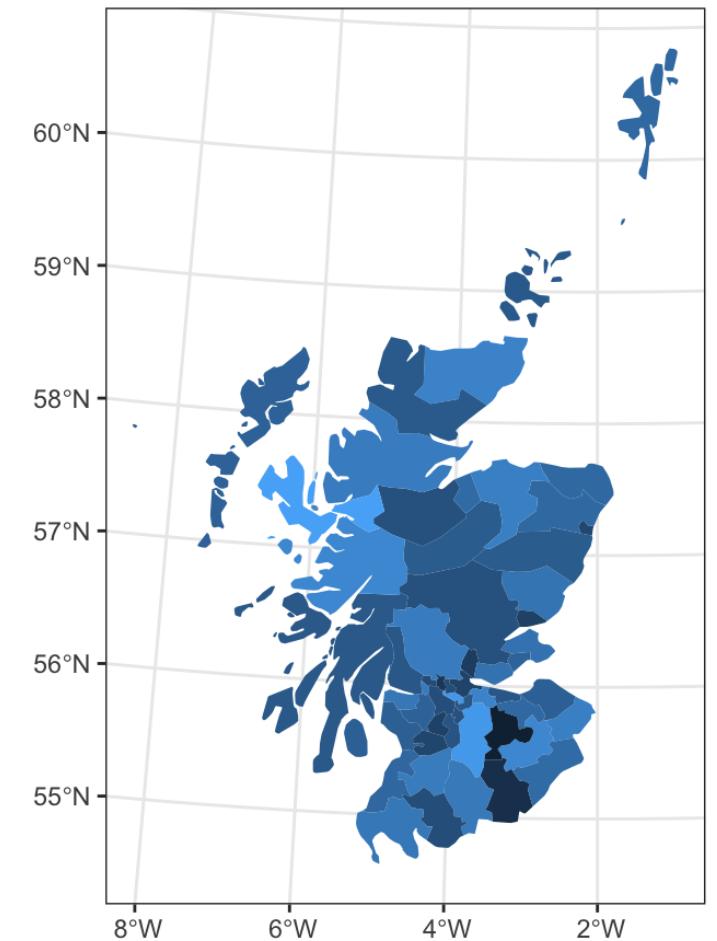


Residuals

Predicted Cases



Residuals



IAR Results

RMSE

```
1 yardstick::rmse_vec(b_iar_pred$Observed, b_iar_pred$y_pred)  
[1] 1.693817
```

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 = 0.90921, p-value = 0.1816
alternative hypothesis: greater
sample estimates:

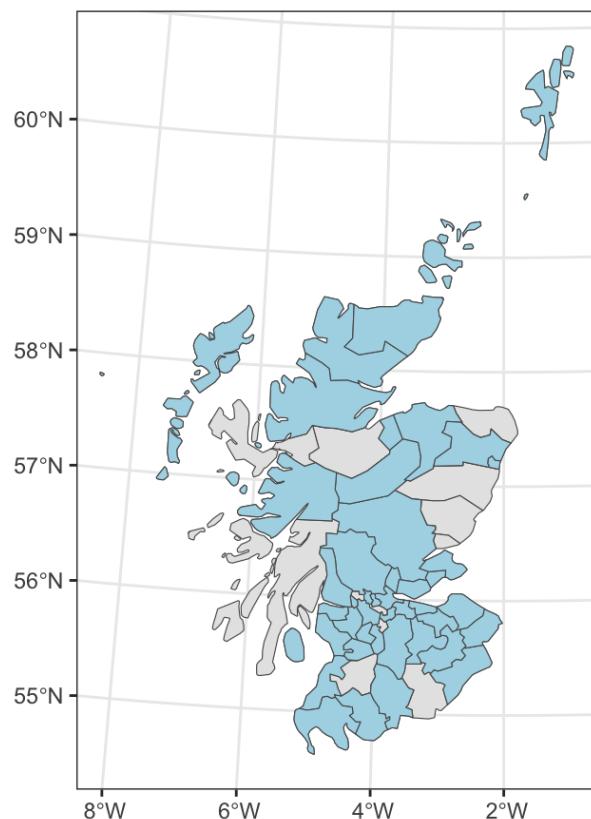
Moran I statistic	Expectation	Variance
0.057638395	-0.018181818	0.006954177

Out of sample predictions

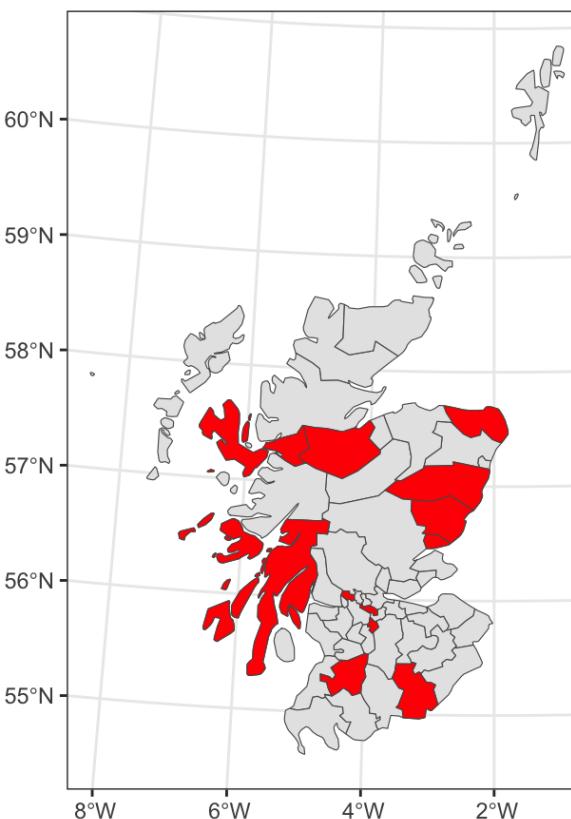
Test / Train split

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

Train



Test



CAR Training Model

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

Prediction

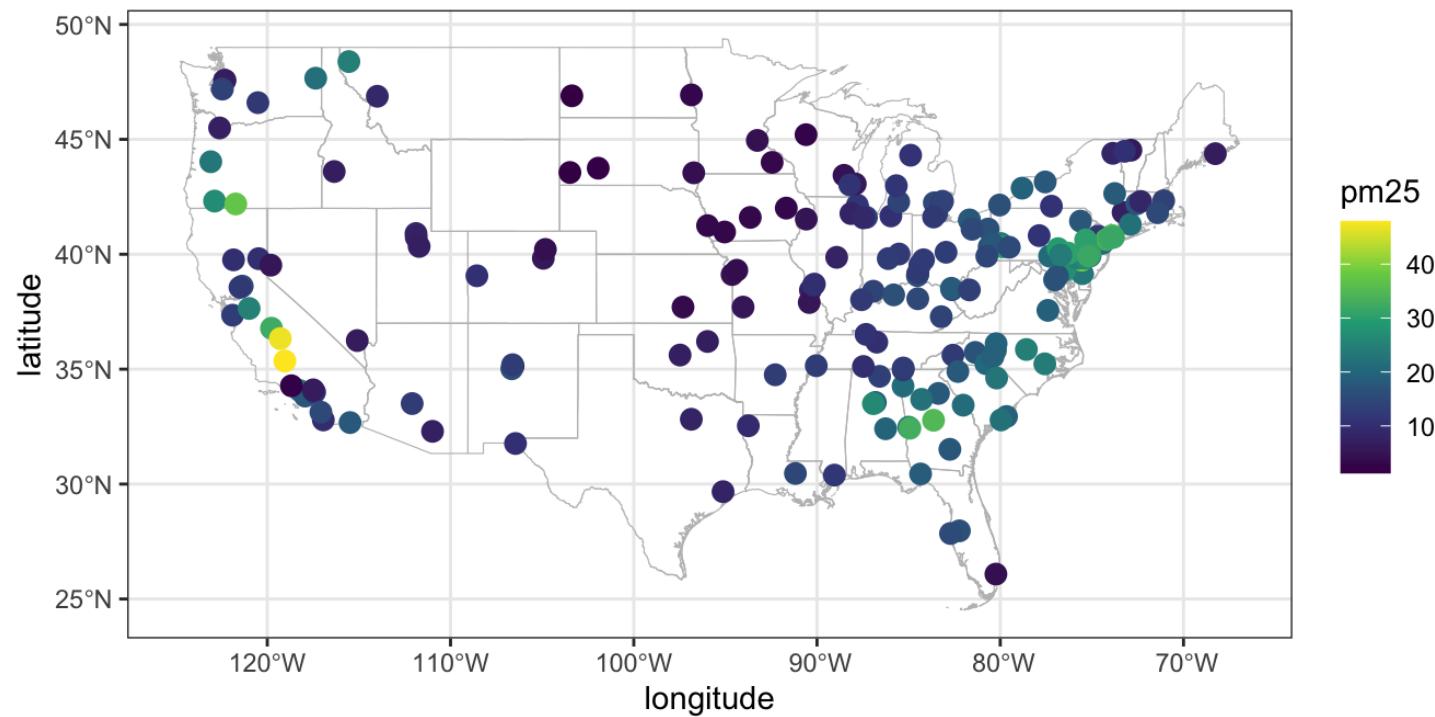
```
1 predict(b_iar_train, newdata=lip_cancer_test)
```

Error: Cannot handle new locations in CAR models.

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.



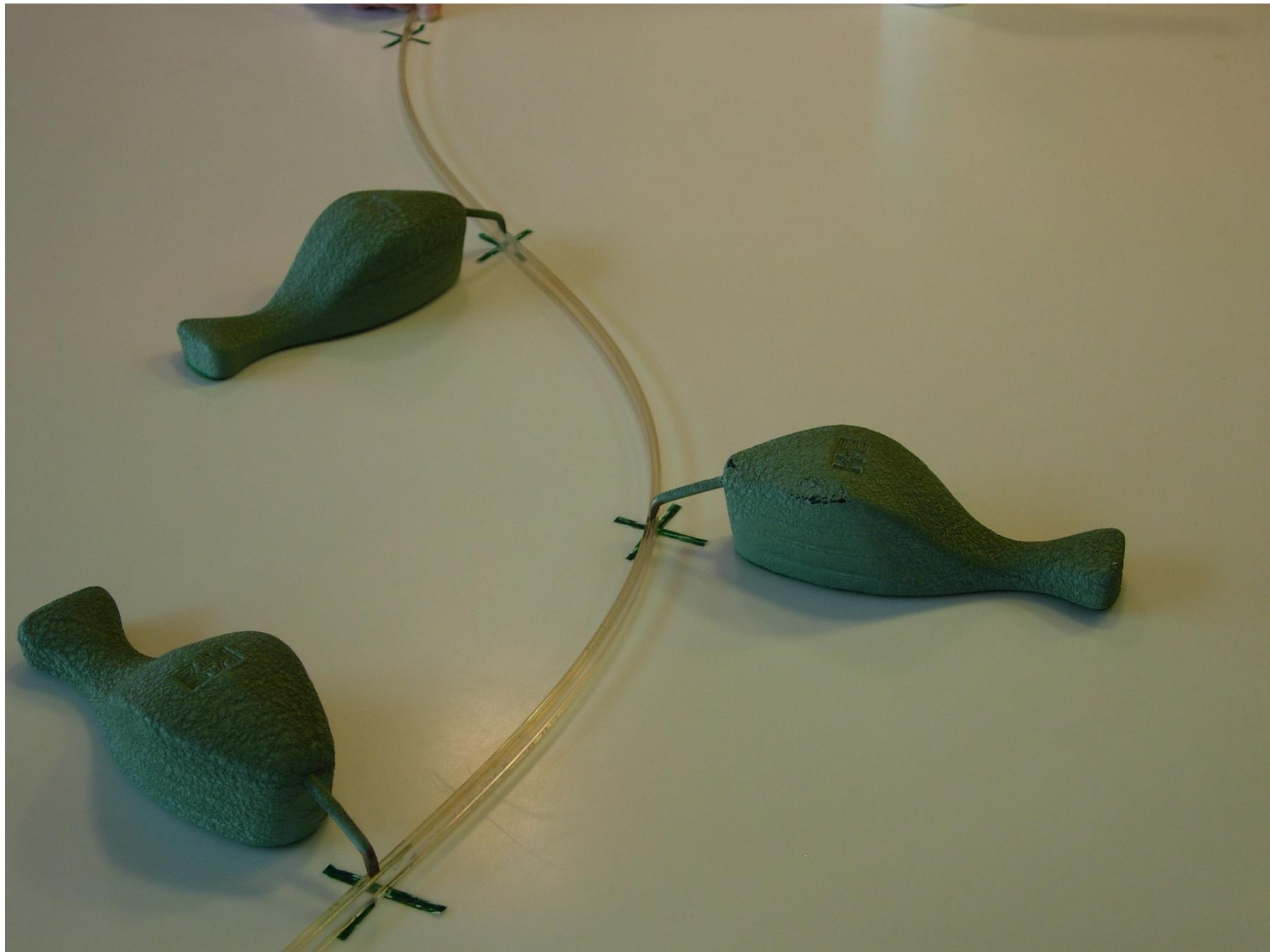
```
1 csn
```

```
# 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





Sta 344/644 - Fall 2023

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 *smooth*.

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

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 (x s).

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,

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

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

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

Prediction locations

```
1 r_usa = stars::st_rasterize(  
2   usa,  
3   stars::st_as_stars(st_bbox(usa),  
4   nx = 100, ny = 50, values=NA_real_)  
5 )
```

```
1 plot(r_usa)
```



Fitting a TPS

```
1 coords = select(csn, long=longitude, lat=latitude) |> as.matrix()
2 (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	NA
GCV.one	1.196496	63.97784	29.91791	4.460553	NA	NA	5
RMSE	NA	NA	NA	NA	NA	NA	NA
pure error	NA	NA	NA	NA	NA	NA	NA

Predictions

```
1 pred = r_usa |>
2   as_tibble() |>
3   filter(!is.na(ID)) |>
4   select(long = x, lat = y)
5
6 tps_pred = pred |>
7   mutate(
8     pred = predict(tps, cbind(long, lat)))
9   ) |>
10 stars::st_as_stars()
```

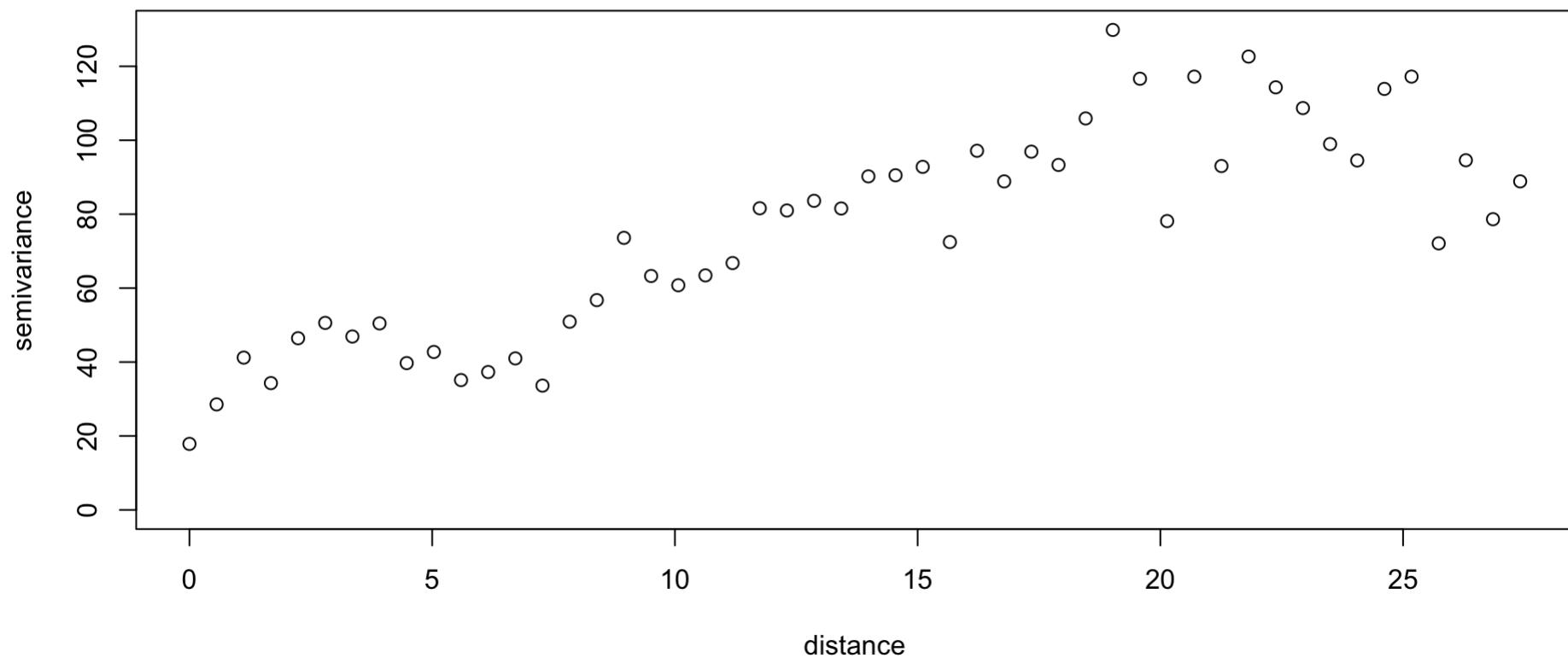
```
1 plot(tps_pred)
```



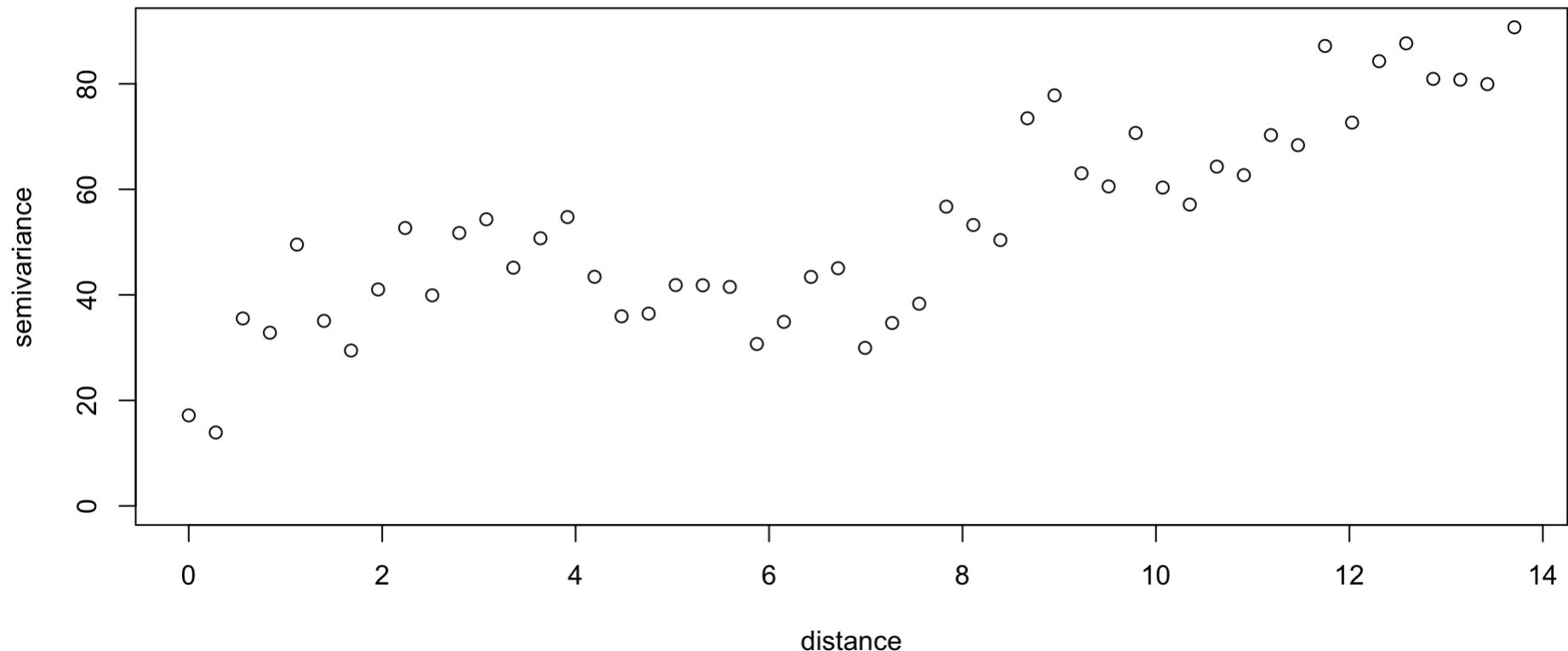
Gaussian Process Models / Kriging

Variogram

```
1 coords = csn |> select(latitude, longitude) |> as.matrix()
2 d = fields:::rdist(coords)
3
4 geoR:::variog(
5   coords = coords, data = csn$pm25, messages = FALSE,
6   uvec = seq(0, max(d)/2, length.out=50)
7 ) |>
8 plot()
```

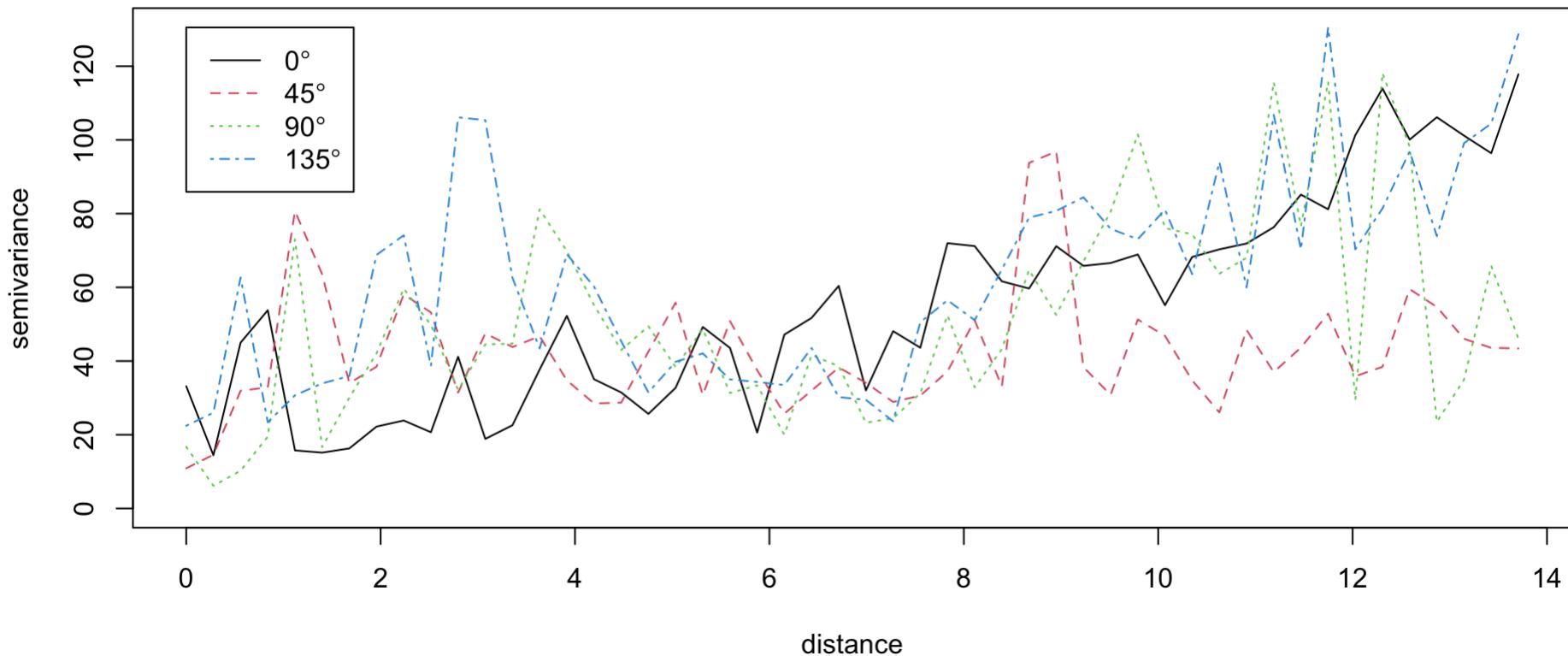


```
1 geoR:::variog(  
2   coords = coords, data = csn$pm25, messages = FALSE,  
3   uvec = seq(0, max(d)/4, length.out=50)  
4 ) |> plot()
```



Isotropy / Anisotropy

```
1 geoR:::variog4(  
2   coords = coords, data = csn$pm25, messages = FALSE,  
3   uvec = seq(0, max(d)/4, length.out = 50)  
4 ) |>  
5 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.

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

$$\{\Sigma\}_{ij} = \sigma^2 \exp(-l \|s_i - s_j\|) + \sigma_n^2 \mathbf{1}_{i=j}$$

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

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

Fitting with gplm()

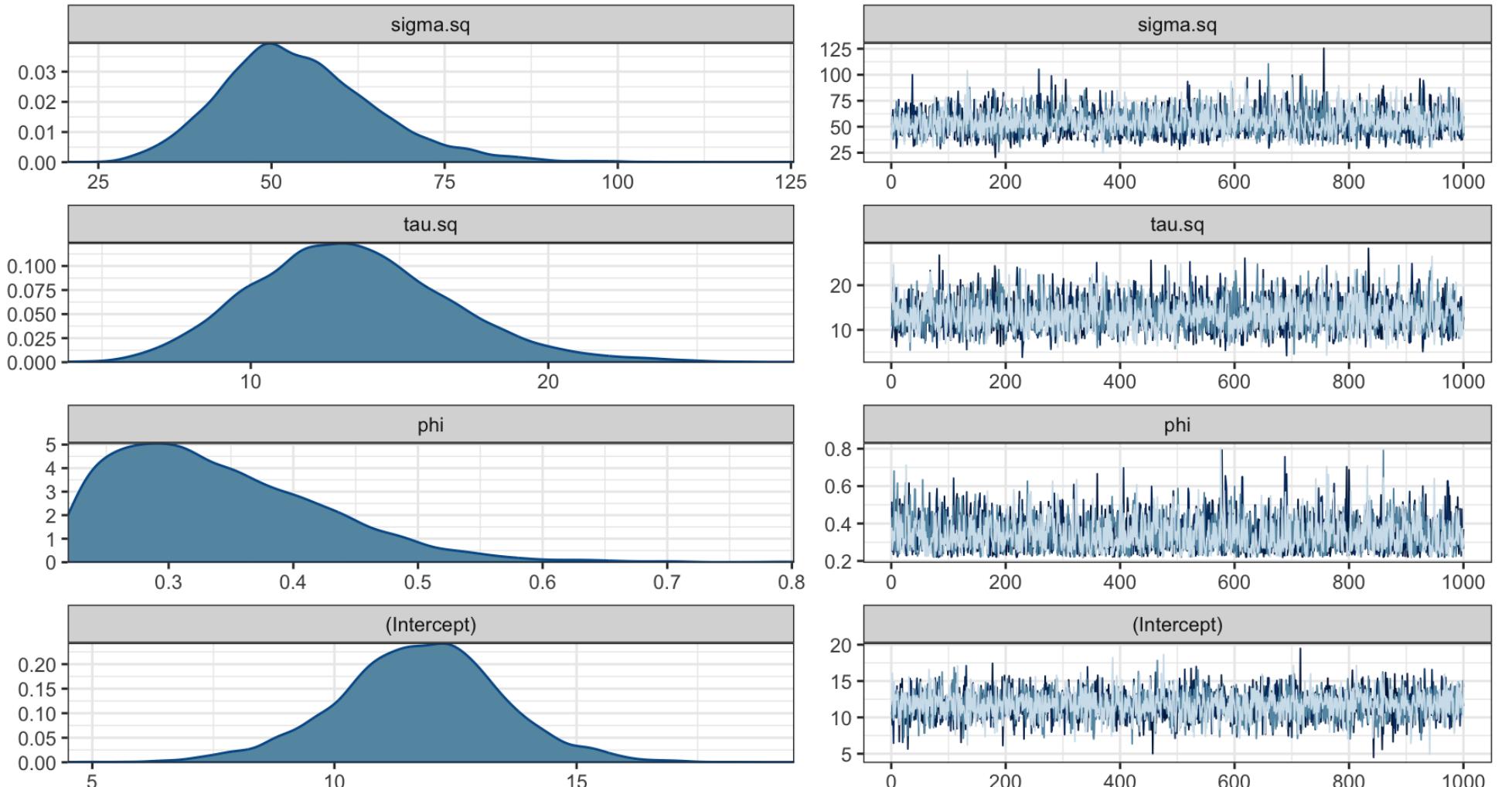
```
1 max_range = max(dist(csn[,c("longitude", "latitude")])) / 4
2
3 m = gplm(
4   pm25~1, data = csn, coords=c("longitude", "latitude"),
5   cov_model = "exponential",
6   starting = list(phi = 3/3, sigma.sq = 33, tau.sq = 17),
7   tuning = list("phi"=0.1, "sigma.sq"=0.1, "tau.sq"=0.1),
8   priors = list(
9     phi.Unif = c(3/max_range, 3/(0.5)),
10    sigma.sq.IG = c(2, 2),
11    tau.sq.IG = c(2, 2)
12  ),
13  thin=10
14 )
```

```
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

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

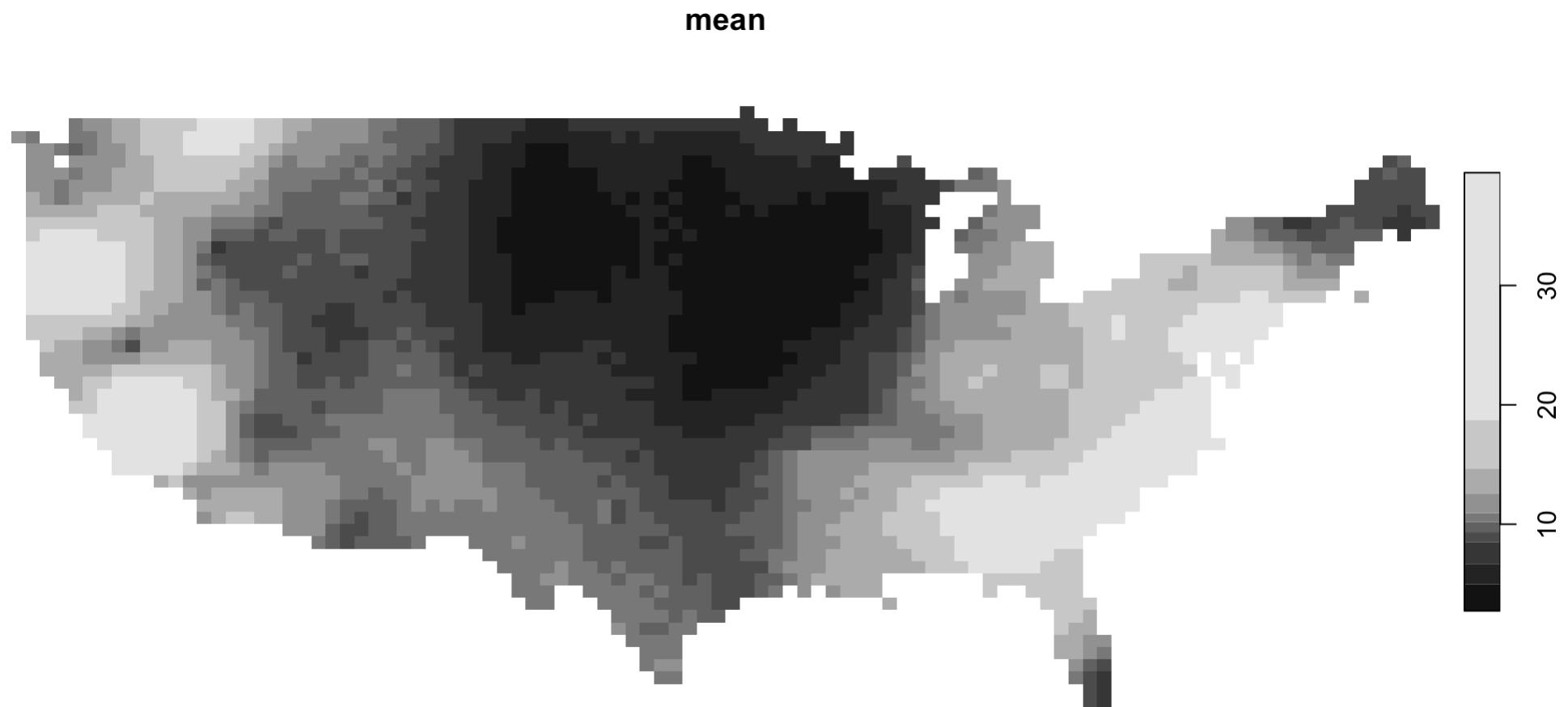
# A draws_matrix: 1000 iterations, 4 chains, and 2828 variables
variable

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

```
1 gp_pred = left_join(  
2   pred |>  
3     mutate(i = row_number()),  
4   tidybayes::gather_draws(p, y[i]) |>  
5     filter(.chain == 1) |>  
6     group_by(.chain, i) |>  
7     summarize(  
8       mean = mean(.value),  
9       med = median(.value),  
10      sd = sd(.value),  
11      .groups = "drop"  
12    ),  
13   by = "i"  
14 )  
15  
16 gp_pred |>  
17   stars::st_as_stars() |>  
18   select(mean) |>  
19   plot()
```

Predictions - raster



```
1 gp_pred |>
2   stars::st_as_stars() |>
3   select(sd) |>
4   plot()
```

sd

