2023 554 Disease Mapping R Notes

Jon Wakefield
Departments of Biostatistics and Statistics
University of Washington

2023-05-05

Scottish lip cancer data

In these notes we will analyze the famous Scottish lip cancer data discussed in lectures. We will fit both non-spatial and spatial random effects smoothing models.

We will also discuss some other topics including how to deal with:

- missing observations, and
- censored observations

We first load some libraries. Note that INLA has a non-standard installation, see INLA DOWNLOAD

```
library(SpatialEpi)
library(RColorBrewer)
library(ggplot2)
library(ggridges)
library(INLA)
library(sf)
library(spdep)
```

We will fit a number of models to the Scottish lip cancer data, but first access the data.

In area i, let Y_i and E_i represent the disease count and expected count.

An initial summary is the Standardized Morbidity Ratio (SMR), which for area i is

$$\text{SMR}_i = \frac{Y_i}{E_i},$$

for $i = 1, \ldots, 56$.

We also have an area-based covariate X_i (proportion in agriculture, fishing and farming) in each of the $i = 1, \ldots, 56$ areas.

```
data(scotland)
names(scotland)
## [1] "geo"             "data"            "spatial.polygon" "polygon"
names(scotland$data)
## [1] "county.names" "cases"        "expected"     "AFF"
head(scotland$data)
```

```
##    county.names cases expected  AFF
## 1 skye-lochalsh     9      1.4 0.16
## 2  banff-buchan    39      8.7 0.16
## 3     caithness    11      3.0 0.10
## 4  berwickshire     9      2.5 0.24
## 5 ross-cromarty    15      4.3 0.10
## 6        orkney     8      2.4 0.24
```

The following is taken from Section 6.2 of Moraga (2020).

We form a data frame `scotdata` containing key variables, and add the SMRs.

```
scotdata <- scotland$data[,c("county.names", "cases", "expected", "AFF")]
scotdata$SMR <- scotdata$cases/scotdata$expected
smap <- (scotland$spatial.polygon)
```

We can use `sapply()` to see that the polygone `ID` slot corresponds to the county names, meaning that polygons and attribute records are in the same order:

```
all.equal(
  sapply(slot(smap, "polygons"), function(x){slot(x, "ID")}),
  as.character(scotdata$county.names)
)
## [1] TRUE
```

We can convert `smap` to an `sfc` object, and add it as geometry column to the data.frame, then convert into an `sf` object:

```
scotdata$geometry <- st_as_sfc(smap)
smap <- st_as_sf(scotdata)
```

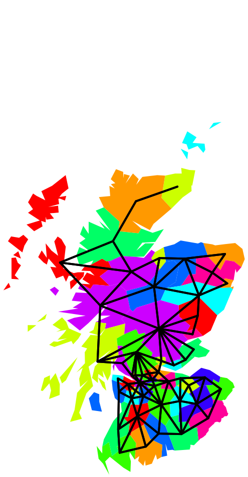
We can look at the first part of the spatial data frame:

```
head(smap)
## Simple feature collection with 6 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 112.576 ymin: 637.8977 xmax: 412.572 ymax: 1043.704
## Projected CRS: +proj=eqc +lat_ts=0 +lat_0=0 +lon_0=0 +x_0=0 +y_0=0 +datum=WGS84 +units=m +no_defs
##    county.names cases expected  AFF      SMR                       geometry
## 1 skye-lochalsh     9      1.4 0.16 6.428571 MULTIPOLYGON (((188.201 800...
## 2  banff-buchan    39      8.7 0.16 4.482759 MULTIPOLYGON (((383.866 865...
## 3     caithness    11      3.0 0.10 3.666667 MULTIPOLYGON (((311.487 968...
## 4  berwickshire     9      2.5 0.24 3.600000 MULTIPOLYGON (((377.18 672....
## 5 ross-cromarty    15      4.3 0.10 3.488372 MULTIPOLYGON (((278.6801 88...
## 6        orkney     8      2.4 0.24 3.333333 MULTIPOLYGON (((319.441 100...
```

We can plot the areas within Scotland as follows:

```
cent <- getSpPPolygonsLabptSlots(scotland$spatial.polygon)
cols <- rainbow(min(10,
                    dim(smap)[1]))
```

```
admin1.mat <- poly2nb(SpatialPolygons(scotland$spatial.polygon@polygons))
admin1.mat <- nb2mat(admin1.mat, zero.policy = TRUE)
colnames(admin1.mat) <- rownames(admin1.mat) <- paste0("admin1_", 1:dim(admin1.mat)[1])

plot(smap$geometry, col = cols, border = F, axes = F,)
  for(i in 1:dim(cent)[1]){
    neighbs <- which(admin1.mat[i,] != 0)
    if(length(neighbs) != 0){
      for(j in 1:length(neighbs)){
        ends <- cent[neighbs,]
        segments(x0 = cent[i, 1],
                 y0 = cent[i, 2],
                 x1 = cent[neighbs[j], 1],
                 y1 = cent[neighbs[j], 2],
                 col = 'black')
      }
    }
  }
```



Now for the map.

```
pal = function(n) brewer.pal(n, "Purples")
plot(smap["SMR"], pal = pal, nbreaks = 8)
```

Observations:

- The SMRs have a large spread, but how much does this reflect sampling variation, rather than true variation?
- There is also increasing trend in the south-north direction.

The variance of the estimate in area i is

$$\text{var}(\text{SMR}_i) = \frac{\text{SMR}_i}{E_i},$$

which will be large if E_i is small.

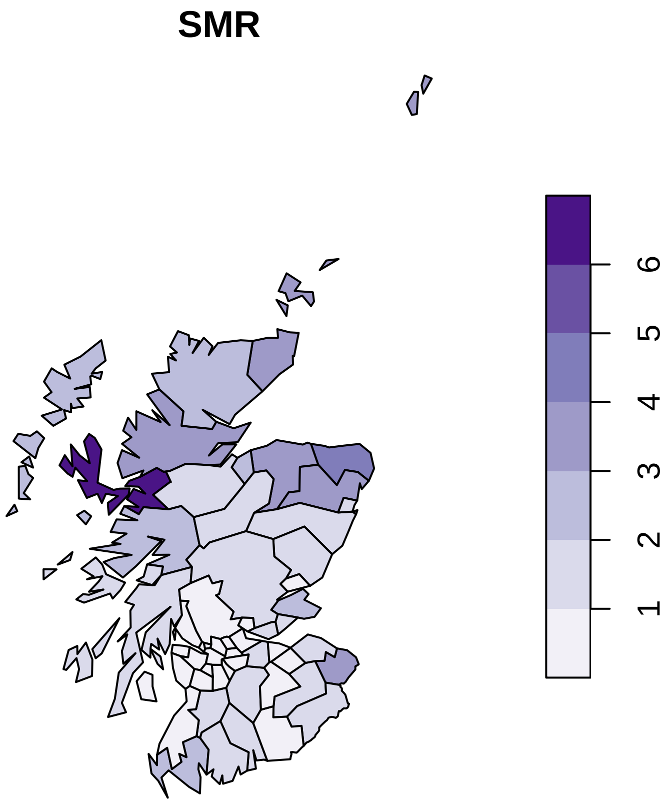


Figure 1: SMRs for Scottish lip cancer data

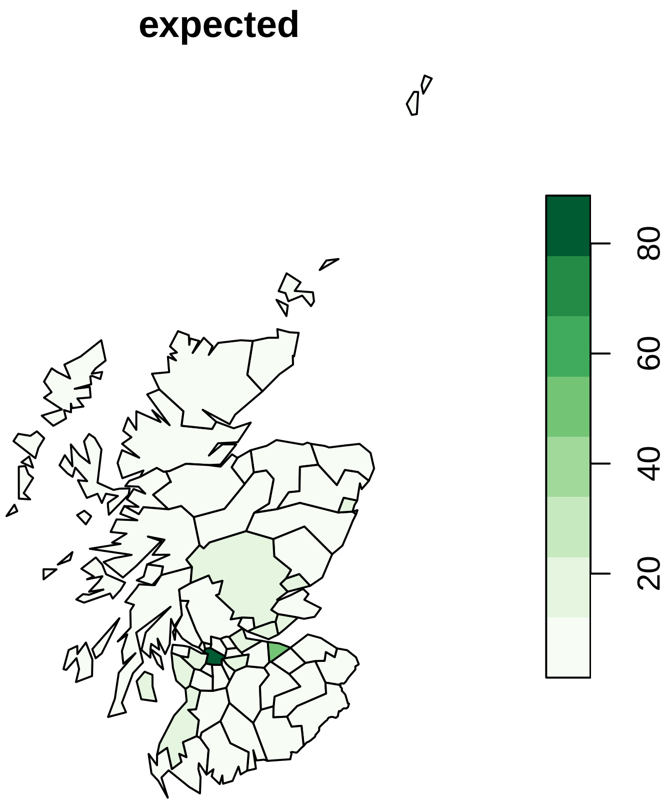
```
summary(smap$expected)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.100   4.050   6.300   9.575  10.125  88.700
```

For the Scottish data the expected numbers are highly variable, with range 1.1–88.7.

This variability suggests that there is a good chance that the extreme SMRs are based on small expected numbers (many of the large, sparsely-populated rural areas in the north have high SMRs).

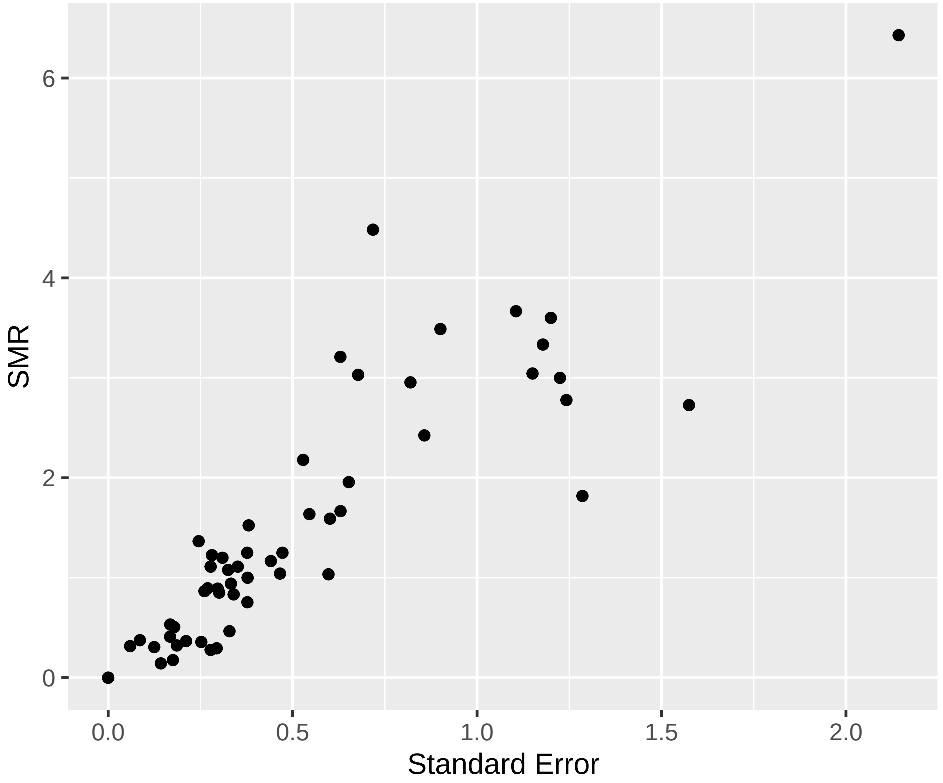
We next map the expected numbers for Scottish lip cancer data

```
pal = function(n) brewer.pal(n, "Greens")
plot(smap["expected"], pal = pal, nbreaks = 8, breaks = "equal")
```



The highest SMRs tend to have the largest standard errors.

```
ggplot(data.frame(se=sqrt(scotdata$SMR/scotdata$expected),SMR=scotdata$SMR),aes(x=se,y=SMR)) + geom_poi
```



SMR interval estimates

Let $\theta_i = \text{SMR}_i$.

We obtain an interval estimate for $\alpha_i = \log \theta_i$ (since the normality of the estimator is likely to be better on this scale) and then transform.

Via the delta method

$$\widehat{\text{var}}(\widehat{\alpha}_i) = \widehat{\text{var}}(\widehat{\theta}_i)|J|^2$$

where $J = \frac{d\alpha_i}{d\theta_i} = \exp(-\alpha_i)$ and $\widehat{\text{var}}(\widehat{\theta}_i) = \widehat{\theta}_i / E_i$.

We obtain:

$$\widehat{\text{var}}(\widehat{\alpha}_i) = [E_i \exp(\widehat{\alpha}_i)]^{-1},$$

to give a 95% confidence interval for θ_i of

$$\exp\left(\widehat{\alpha}_i \pm 1.96 \times \sqrt{\widehat{\text{var}}(\widehat{\alpha}_i)}\right).$$

SMR estimates when $Y_i = 0$

When $Y_i = 0$, we obtain an SMR of 0, and (more worryingly) a standard error of zero.

In this case, we carry out an adjustment and set $Y_i^\star = Y_i + 0.5$ and $E_i^\star = E_i + 0.5$ to give the estimator

$$\theta_i^\star = \text{SMR}_i^\star = Y_i^\star / E_i^\star,$$

with $\hat{\text{var}}(\widehat{\theta}_i^\star) = \widehat{\theta}_i^\star / E_i^\star$.

Also let $\alpha_i^\star = \log \theta_i^\star$.

We obtain:

$$\widehat{\text{var}}(\widehat{\alpha}_i^\star) = (E_i^\star \exp(\widehat{\alpha}_i^\star))^{-1},$$

to give a 95% confidence interval of

$$\exp\left(\widehat{\alpha}_i^\star \pm 1.96 \times \sqrt{\widehat{\text{var}}(\widehat{\alpha}_i^\star)}\right).$$

SMR interval estimates

The addition of 0.5 is somewhat ad hoc but corresponds to a Ga(0.5,0.5) prior on the relative risk. This prior has 0.025, 0.5, 0.975 quantiles of 0.00098, 0.45, 5.0.

The addition of a non-integer also highlights that some adjustment has been made!

This prior is contributing information equivalent to observing an expected number of 0.5 and 'half a case'.

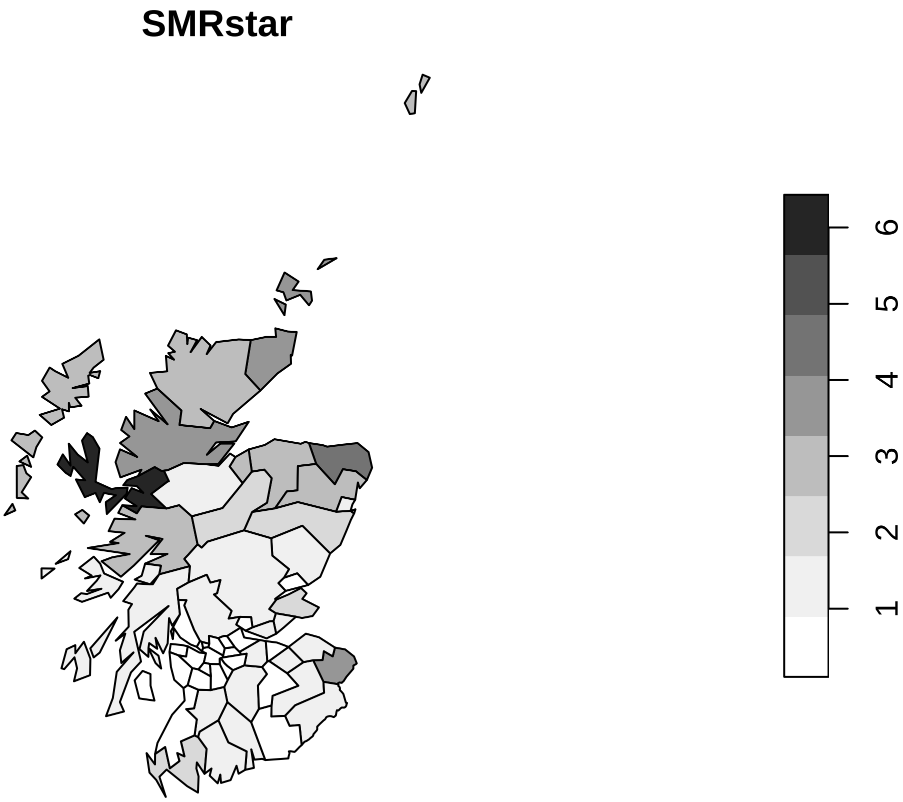
SMR estimates with adjustment. We create estimates adjusted for zeroe: if the number of cases is equal to zero both the number of cases and the expecteds are increased by 0.5.

```
Ystar <- ifelse(scotdata$cases==0,0.5,scotdata$cases)
Estar <- ifelse(scotdata$cases==0,scotdata$expected+0.5,scotdata$expected)
SMRstar <- Ystar/Estar
alphastar <- log(SMRstar)
varalphastar <- 1/(SMRstar*Estar)
SMRlower <- exp(alphastar-1.96*sqrt(varalphastar))
SMRupper <- exp(alphastar+1.96*sqrt(varalphastar))
SMRwidth <- SMRupper - SMRlower
scotdata$SMRstar <- SMRstar
```

```
scotdata$Estar <- Estar
scotdata$SMRlower <- SMRlower
scotdata$SMRupper <- SMRupper
scotdata$SMRwidth <- SMRwidth
geometries = st_geometry(smap)
smap = scotdata
smap$geometries = geometries
smap = st_as_sf(smap)
```

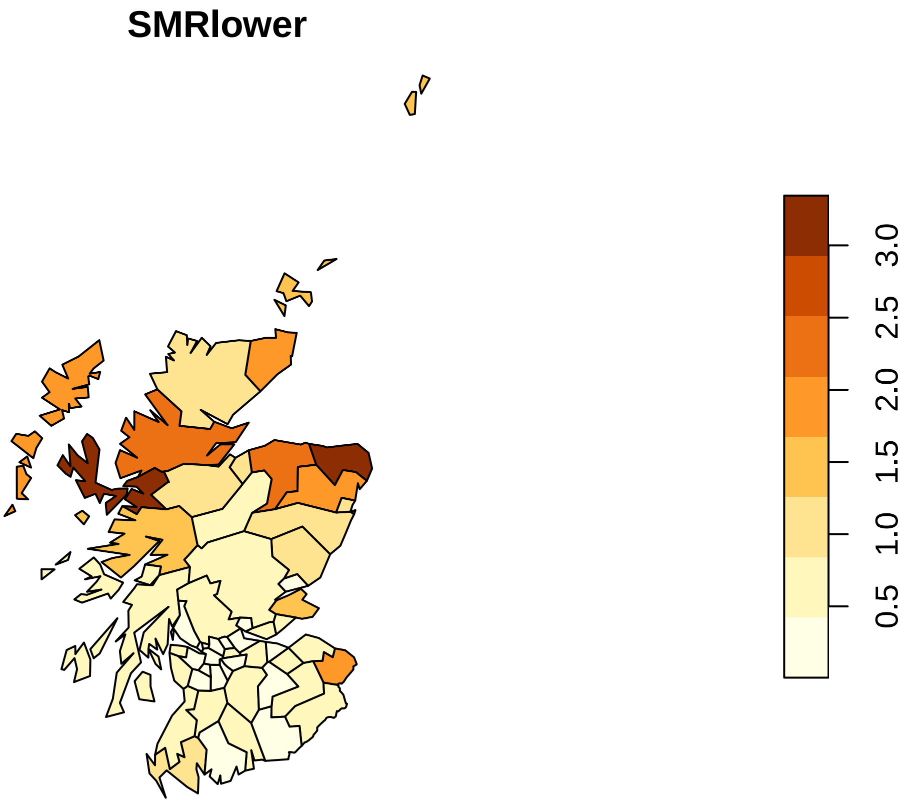
Point estimates with adjustment

```
pal = function(n) brewer.pal(n,"Greys")
plot(smap["SMRstar"],pal = pal, nbreaks = 8, breaks = "equal")
```

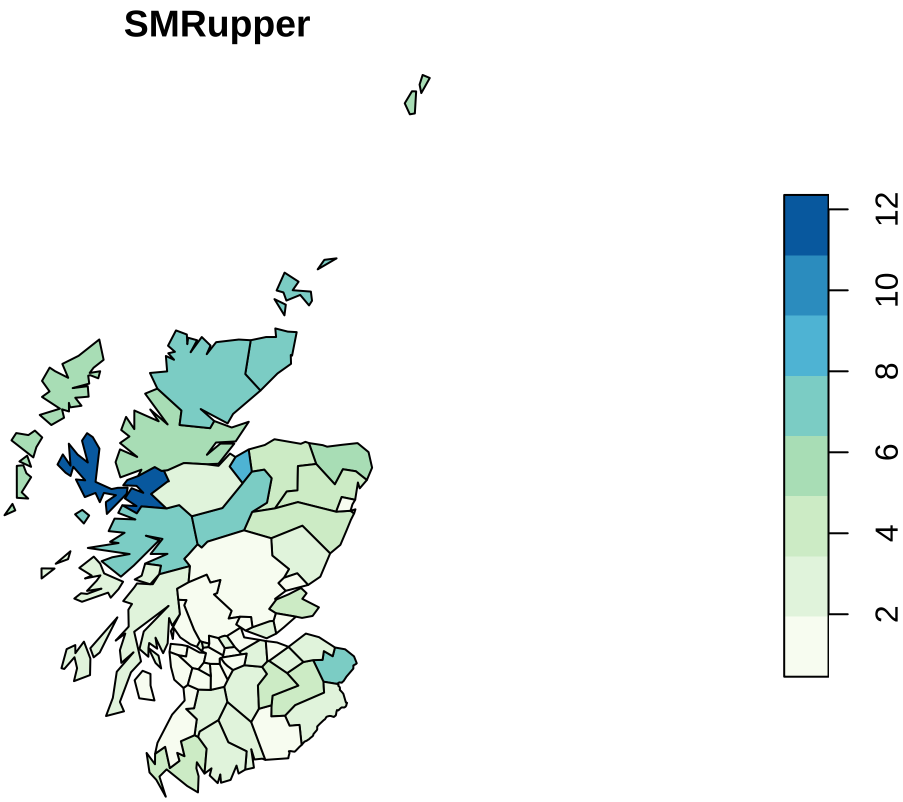


Lower and upper confidence intervals with adjustment.

```
pal = function(n) brewer.pal(n, "YlOrBr")
plot(smap["SMRlower"], pal = pal, nbreaks = 8, breaks = "equal")
```

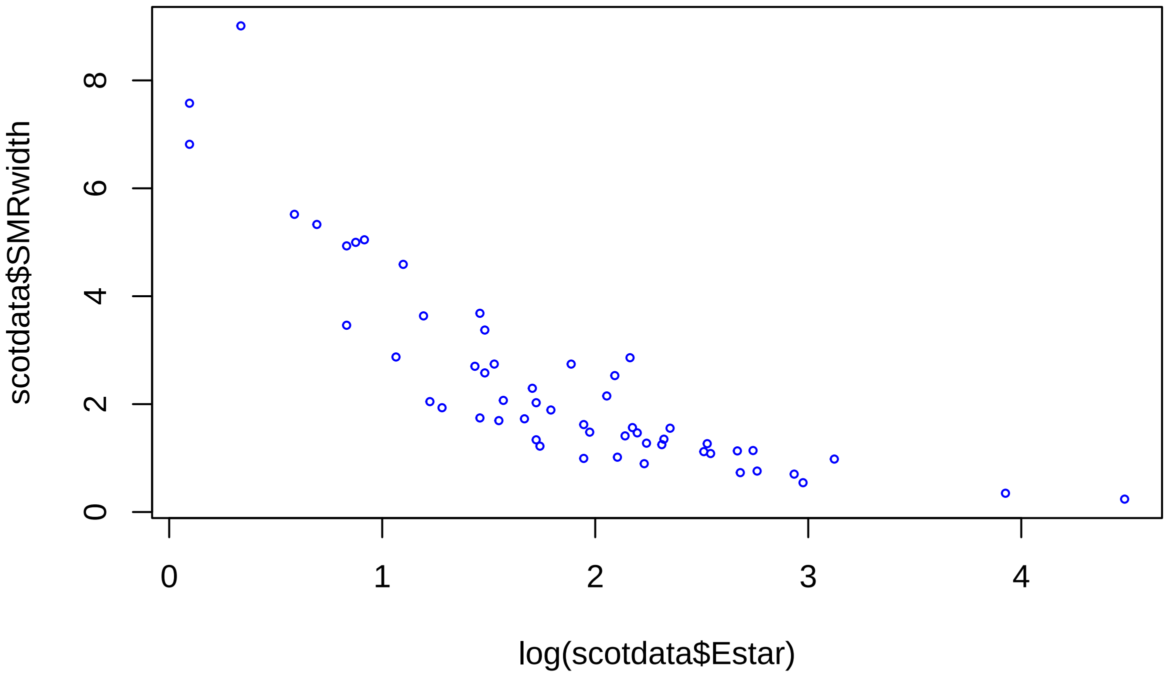

SMRlower

```
pal = function(n) brewer.pal(n, "GnBu")
plot(smap["SMRupper"], pal=pal, nbreaks = 8, breaks = "equal")
```


SMRupper

The width decreases with increasing expected numbers!

```
plot(scotdata$SMRwidth~log(scotdata$Estar),col="blue",cex=.5)
```



Poisson-Lognormal non-spatial smoothing model

We now consider an alternative lognormal model for the relative risks, but still independent.

A Poisson-lognormal non-spatial random effect model is given by:

$$\begin{aligned} Y_i|\beta_0, e_i &\sim_{ind} \text{Poisson}(E_i \mathrm{e}^{\beta_0} \mathrm{e}^{e_i}), \\ e_i|\sigma_e^2 &\sim_{iid} \mathrm{N}(0, \sigma_e^2) \end{aligned}$$

where e_i are area-specific random effects that capture the residual or unexplained (log) relative risk of disease in area i, $i = 1, ..., n = 56$.

Note that in INLA the uncertainty in the distribution of the random effect is reported in terms of the precision (the reciprocal of the variance, τ_e).

This model gives rise to the posterior distribution;

$$p(\beta_0, \tau_e, e_1, \ldots, e_n|y) = \frac{\prod_{i=1}^{n} \Pr(Y_i|\beta_0, e_i) p(e_i|\tau_e) p(\beta_0) p(\tau_e)}{\Pr(y)}.$$

The full posterior is an $(n + 2)$-dimensional distribution and INLA by default produces summaries of the univariate posterior distributions for β_0 and τ_e.

The posteriors on the random effects $p(e_i|y)$ can be extracted, as we will show in subsequent slides.

INLA for the Poisson-Lognormal model

We fit the Poisson-Lognormal model to the Scottish lip cancer data.

We first show a fit with no prior specifications given.

```
# Fit Poisson-lognormal model in INLA:
scotland.fit0 <- inla(Counts ~ 1 + f(Region, model="iid"),
  data=Scotland, family="poisson", E=E)
scotland.fit0$summary.fixed[,1:5]
##                   mean        sd 0.025quant   0.5quant 0.975quant
## (Intercept) 0.08193016 0.1155899 -0.1496767 0.08336085  0.3054301
scotland.fit0$summary.hyper[,1:5]
##                          mean        sd 0.025quant 0.5quant 0.975quant
## Precision for Region 1.839585 0.4649889   1.077331 1.788511   2.866096
```

A sanity check:

```
beta0est <- scotland.fit0$summary.fixed[,4]
sigma2est <- 1/scotland.fit0$summary.hyper[,4]
exp(beta0est+0.5*sigma2est)
## [1] 1.437525
```

Now we place a prior on the precision and ask for fitted values to be computed.

Note:

- The specification of the penalized complexity prior (Simpson et al 2017) for the precision $\tau_e = \sigma_e^{-2}$. Here we specify that there is a 5% chance that the standard deviation σ_e is greater than 1. The end of these notes contains a brief description of penalized complexity (PC) priors.

```
# Fit Poisson-lognormal model in INLA with prior specified
pcprec <- list(theta=list(prior='pc.prec',param=c(1,.05)))
scotland.fit1 <- inla(Counts ~ 1 + f(Region, model="iid", hyper=pcprec),
  data=Scotland, family="poisson", E=E,
  # Next two lines give us calculated fitted values
  control.predictor = list(compute = TRUE),
  control.compute = list(return.marginals.predictor = TRUE))

scotland.fit1$summary.fixed[,1:5]
##                  mean        sd 0.025quant   0.5quant 0.975quant
## (Intercept) 0.0807738 0.1165394 -0.1526246 0.08218384  0.3061921
scotland.fit1$summary.hyper[,1:5]
##                          mean        sd 0.025quant 0.5quant 0.975quant
## Precision for Region 1.799251 0.4490446   1.064517 1.748894    2.80371
```

Very little sensitivity to the prior on the precision.

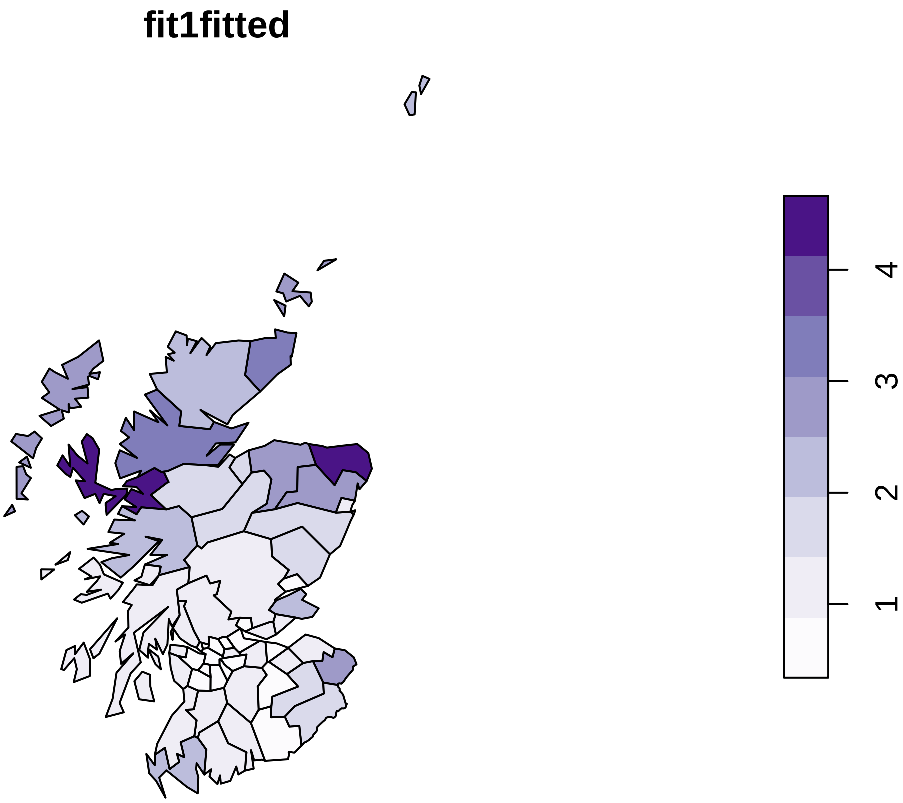
Let's look at the potential output:

```
names(scotland.fit1)
##  [1] "names.fixed"                 "summary.fixed"
##  [3] "marginals.fixed"             "summary.lincomb"
##  [5] "marginals.lincomb"           "size.lincomb"
##  [7] "summary.lincomb.derived"     "marginals.lincomb.derived"
##  [9] "size.lincomb.derived"        "mlik"
## [11] "cpo"                         "gcpo"
## [13] "po"                          "waic"
## [15] "model.random"                "summary.random"
## [17] "marginals.random"             "size.random"
## [19] "summary.linear.predictor"     "marginals.linear.predictor"
## [21] "summary.fitted.values"        "marginals.fitted.values"
## [23] "size.linear.predictor"        "summary.hyperpar"
## [25] "marginals.hyperpar"           "internal.summary.hyperpar"
## [27] "internal.marginals.hyperpar" "offset.linear.predictor"
## [29] "model.spde2.blc"              "summary.spde2.blc"
## [31] "marginals.spde2.blc"          "size.spde2.blc"
## [33] "model.spde3.blc"              "summary.spde3.blc"
## [35] "marginals.spde3.blc"          "size.spde3.blc"
## [37] "logfile"                      "misc"
## [39] "dic"                          "mode"
## [41] "joint.hyper"                  "nhyper"
## [43] "version"                      "Q"
## [45] "graph"                        "ok"
## [47] "cpu.used"                     "all.hyper"
## [49] ".args"                        "call"
## [51] "model.matrix"
```

We now extract the posterior medians of the log relative risks.

We now map the posterior medians of the relative risks.

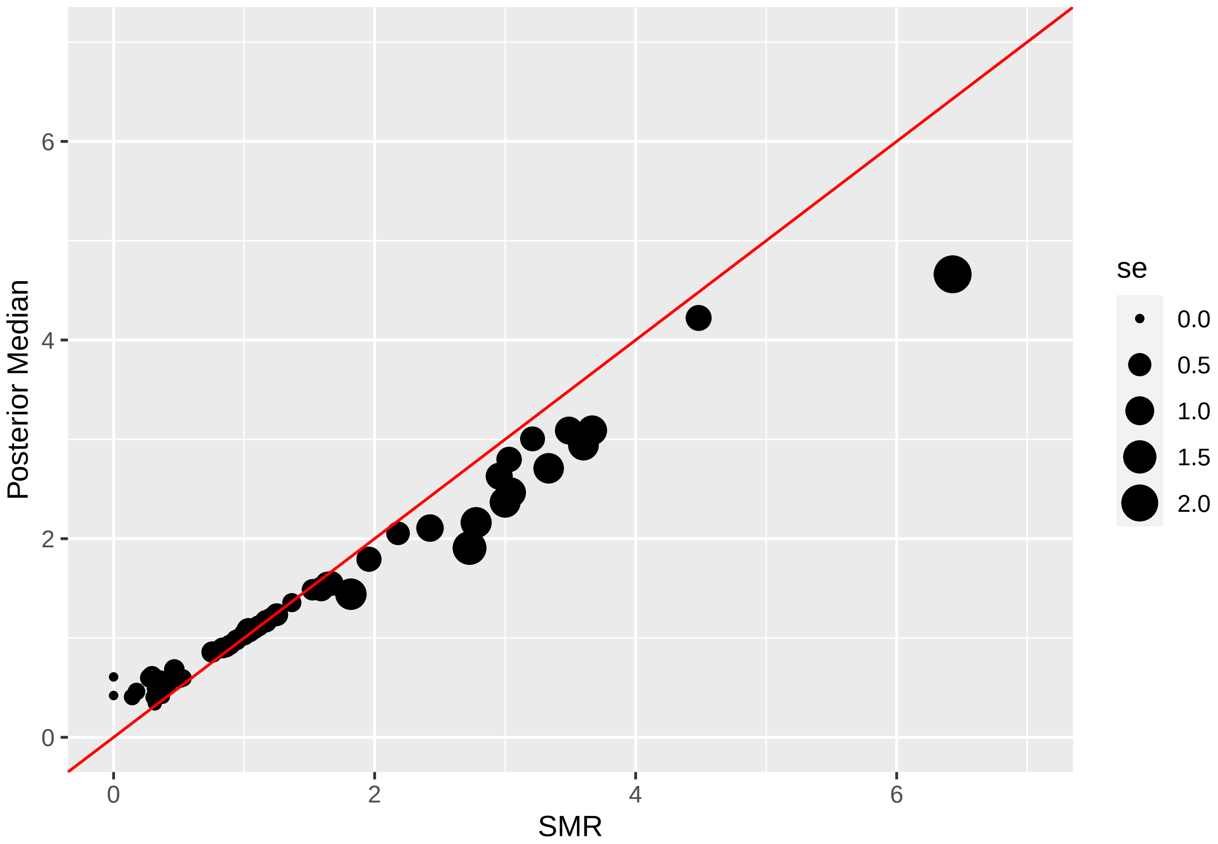
```
smap$fit1fitted <- scotland.fit1$summary.fitted.values$`0.5quant`
pal = function(n) brewer.pal(n,"Purples")
plot(smap["fit1fitted"], pal = pal, nbreaks = 8, breaks = "equal")
```



Now compare the medians with the SMRs - we see some shinrkage, particularly for the low and high SMRs that have relatively large standard errors.

The standard erroors of zero are artfifacts of the SMRs being estimated as zero when $Y_i = 0$.

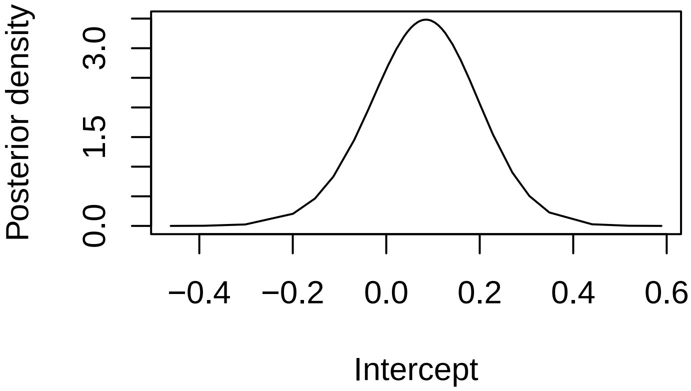
```
se <- sqrt(scotdata$SMR/scotdata$expected)
ggplot(data.frame(pmedian=scotland.fit1$summary.fitted.values$`0.5quant`,SMR=scotdata$SMR),
       aes(y=pmedian,x=SMR,size = se)) + geom_point() + labs(y="Posterior Median",x="SMR") + geom_ablin
```



```
summary(scotland.fit1)
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts,
##    ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##    scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##    ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##    verbose, ", " lincomb = lincomb, selection = selection, control.compute
##    = control.compute, ", " control.predictor = control.predictor,
##    control.family = control.family, ", " control.inla = control.inla,
##    control.fixed = control.fixed, ", " control.mode = control.mode,
##    control.expert = control.expert, ", " control.hazard = control.hazard,
##    control.lincomb = control.lincomb, ", " control.update =
##    control.update, control.lp.scale = control.lp.scale, ", "
##    control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##    ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##    num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
##    working.directory = working.directory, ", " silent = silent, inla.mode
##    = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
##    .parent.frame)")
## Time used:
##     Pre = 2.82, Running = 0.435, Post = 0.0252, Total = 3.28
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant mode kld
## (Intercept) 0.081 0.117     -0.153    0.082      0.306   NA   0
##
## Random effects:
##   Name     Model
##     Region IID model
##
## Model hyperparameters:
##                     mean    sd 0.025quant 0.5quant 0.975quant mode
## Precision for Region 1.80 0.449       1.06     1.75       2.80   NA
##
## Marginal log-Likelihood:  -185.49
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
expbeta0med <- scotland.fit1$summary.fixed[4] # intercept
sdmed <- 1/sqrt(scotland.fit1$summary.hyperpar[4]) # sd
```

We examine the posterior marginal distribution for the intercept β_0.

```
plot(scotland.fit1$marginals.fixed$`(Intercept)`[,2]~
scotland.fit1$marginals.fixed$`(Intercept)`[,1],type="l",
xlab="Intercept",ylab="Posterior density")
```



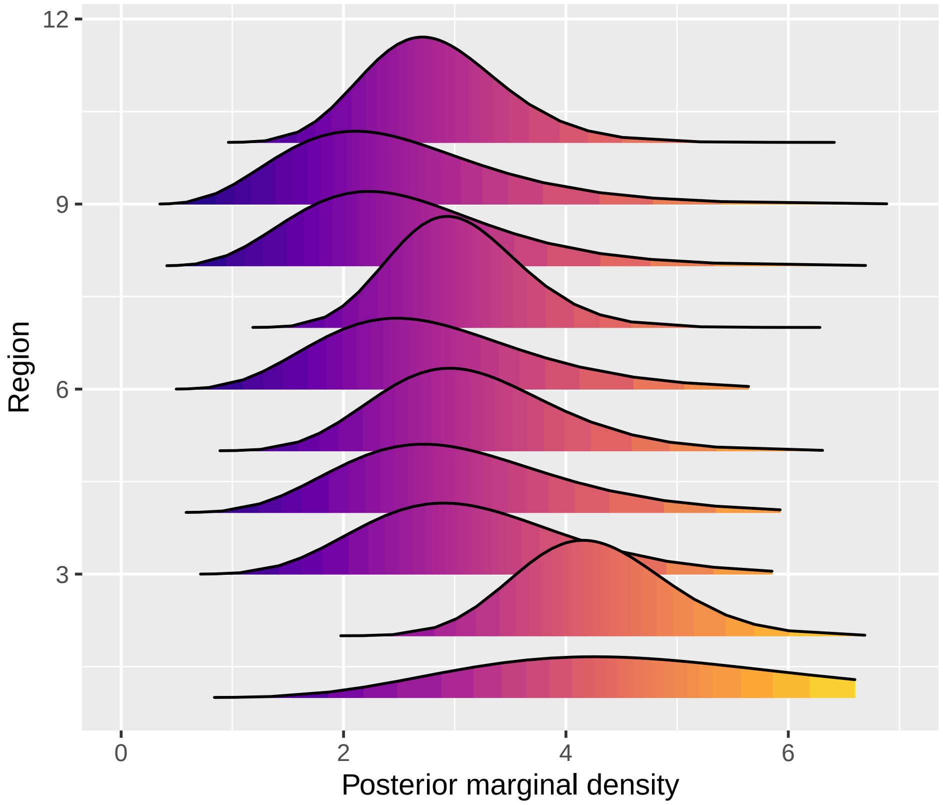
Ridgeplots: posterior marginals for regions

A function to extract a specified marginal for all regions from an INLA model

```
# function to extract the marginal densities and make a data frame to plot
extract_marginals_to_plot <- function(marg) {
  posterior_densities <- data.frame()
  for (i in 1:length(marg)) {
    tmp <- data.frame(marg[[i]])
    tmp$Region <- i
    posterior_densities <- rbind(posterior_densities,tmp)
  }
  return(posterior_densities)
}
```

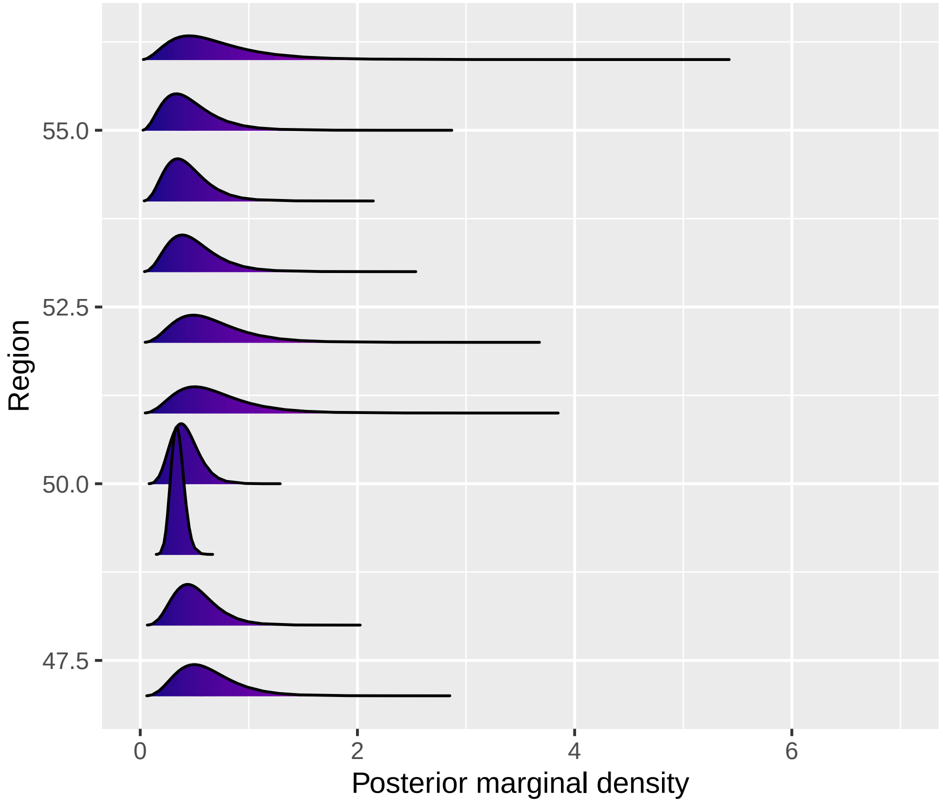
We display ridgeplots for marginal posterior RRs in regions 1–10:

```
marginal_of_interest <- scotland.fit1$marginals.fitted.values
post_dens <- extract_marginals_to_plot(marginal_of_interest)
# we use the ggridges package to plot the marginals for first 28 Regions
ggplot(data = post_dens[post_dens$Region <= 10,],
       aes(x = x, y = Region, height = y, group = Region, fill = ..x..)) +
  geom_density_ridges_gradient(stat = "identity", alpha = 0.5) +
  scale_fill_viridis_c(option = "C") + xlab("Posterior marginal density") +
  xlim(0,7) +
  theme(legend.position = 'none')
```



Next, ridgeplots for marginal posterior RRs in regions 47–56

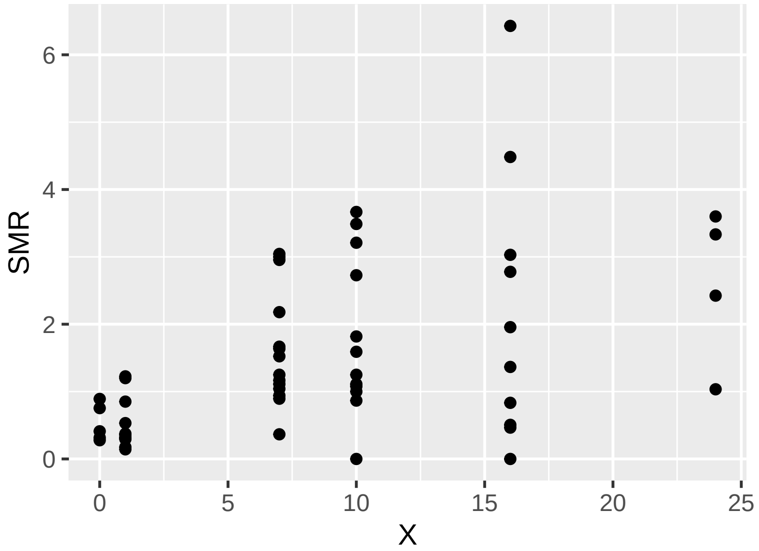
```
# we use the ggridges package to plot the marginals for last 10 Regions
ggplot(data = post_dens[post_dens$Region > 46,],
       aes(x = x, y = Region, height = y, group = Region, fill = ..x..)) +
  geom_density_ridges_gradient(stat = "identity", alpha = 0.5) +
  scale_fill_viridis_c(option = "C") + xlab("Posterior marginal density") +
  xlim(0,7) +
  theme(legend.position = 'none')
```



An excess-Poisson model with covariate

We now add AFF, as a sanity check we first plot the SMR versus AFF.

```
ggplot(Scotland,aes(x=X,y=Counts/E)) + geom_point() + labs(y="SMR")
```



We fit a quasi-likelihood model with

$$\begin{aligned} E[Y_i] &= E_i \exp(\beta_0 + \beta_1 x_i) \\ \text{var}(Y_i) &= \kappa \times E[Y_i] \end{aligned}$$

This model allows for excess-Poisson variation (overdispersion) via κ, but does not allow for spatial dependence.

```
modQL <- glm(Scotland$Counts~Scotland$X,offset=log(Scotland$E),family="quasipoisson")
coef(modQL)
## (Intercept)  Scotland$X
## -0.54226816  0.07373219
sqrt(diag(vcov(modQL)))
## (Intercept)  Scotland$X
##  0.15418099  0.01320769
summary(modQL)
##
## Call:
## glm(formula = Scotland$Counts ~ Scotland$X, family = "quasipoisson",
##     offset = log(Scotland$E))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7632  -1.2156   0.0967   1.3362   4.7130
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.54227    0.15418  -3.517 0.000893 ***
## Scotland$X   0.07373    0.01321   5.583 7.89e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 4.917964)
##
##     Null deviance: 380.73  on 55  degrees of freedom
## Residual deviance: 238.62  on 54  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

The estimated RR is $\exp(0.074) = 1.08$, so that an area whose AFF is 1 unit higher has an 8% higher relative risk – not an individual-level association (beware the ecological fallacy!).

The overdispersion is estimated as $\widehat{\kappa} = 4.9$, which is considerable. Large excess-Poisson variation implies imprtant missing covariates/confounders, and if these have spatial structure, then this will lead to strong spatial dependence (though we emphasize that the quasi-Poisson model does not account for this).

Poisson-Lognormal non-spatial model with covariates

We now fit the three-stage model:

Stage 1: The Likelihood $Y_i|\theta_i \sim \text{Poisson}(E_i\theta_i)$, $i = 1, \dots, n$ with

$$\log\theta_i = \beta_0 + x_i\beta_1 + e_i$$

where x_i is the AFF in area i.

Stage 2: The random effects (prior distribution) is $e_i|\sigma_e^2 \sim_{iid} N(0, \sigma_e^2)$.

Stage 3: The hyperprior on the hyperparameters $\beta_0, \beta_1, \sigma_e^2$:

$$p(\beta_0, \beta_1, \sigma_e^2) = p(\beta_0)p(\beta_1)p(\sigma_e^2)$$

so that here we have assumed independent priors.

```
# No spatial effects with covariate
scotland.fit1X <- inla(Counts~1+X+f(Region, model="iid",
    hyper=pcprec),data=Scotland, family="poisson",E=E)
scotland.fit1X$summary.fixed[1:5]
##                    mean         sd  0.025quant    0.5quant  0.975quant
## (Intercept) -0.49181346 0.15947210 -0.81075911 -0.49006442 -0.18279245
## X            0.06836907 0.01423825  0.04034272  0.06836394  0.09642797
scotland.fit1X$summary.hyperpar[1:5]
##                          mean        sd 0.025quant 0.5quant 0.975quant
## Precision for Region 2.941347 0.8361568   1.639126 2.827342   4.838659
```

Poisson-Lognormal non-spatial model with covariates: inference

If we are interested in the association with the AFF variable we can examine the posterior summaries, on the original (to give a log RR) or exponentiated (to give a RR) scale.

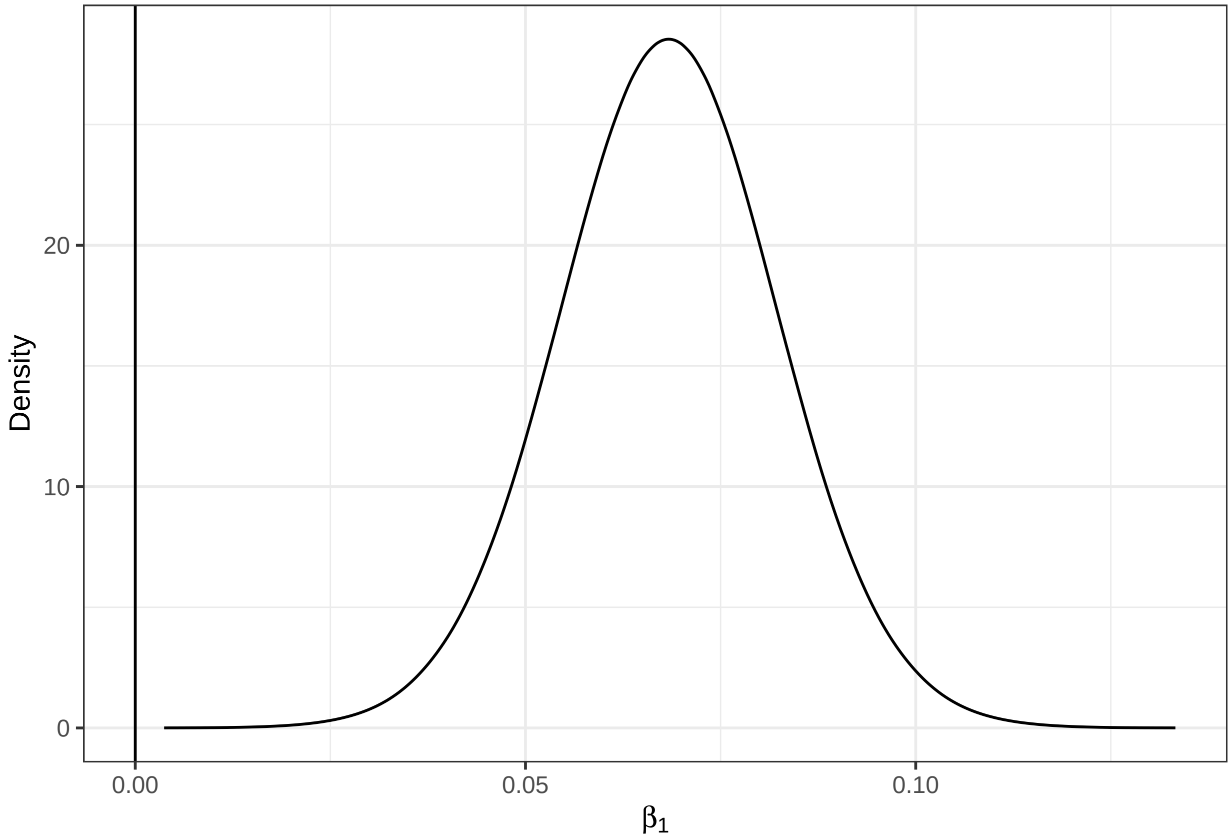
From these summaries we might extract the posterior median as a point estimate, or take the 2.5% and 97.5% points as a 95% credible interval.

```
scotland.fit1X$summary.fixed[2,1:5]
##         mean         sd 0.025quant   0.5quant 0.975quant
## X 0.06836907 0.01423825 0.04034272 0.06836394 0.09642797
exp(scotland.fit1X$summary.fixed[2,3:5])
##   0.025quant 0.5quant 0.975quant
## X   1.041168 1.070755    1.10123
```

Note that we only exponentiate the quantiles of the posterior – the mean and variance cannot be legally exponentiated to give something useful.

Let's look at the posterior marginal for the log relative risk β_1.

```
marginal <- inla.smarginal(scotland.fit1X$marginals.fixed$X)
marginal <- data.frame(marginal)
ggplot(marginal, aes(x = x, y = y)) + geom_line() +
  labs(x = expression(beta[1]), y = "Density") +
  geom_vline(xintercept = 0, col = "black") + theme_bw()
```



Parameter interpretation

```
scotland.fit1X$summary.fixed[1:5]
##                     mean         sd  0.025quant    0.5quant  0.975quant
## (Intercept) -0.49181346 0.15947210 -0.81075911 -0.49006442 -0.18279245
## X            0.06836907 0.01423825  0.04034272  0.06836394  0.09642797
```

The posterior mean for the intercept is $E[\beta_0|y] = -0.49$.

The posterior median for the relative risk associated with a 1 unit increase in X is $\text{median}(\exp(\beta_1)|y) = \exp(0.068) = 1.07$. This latter calculation exploits the fact that we can transform quantiles[1]

Similarly a 95% credible interval for the relative risk $\exp(\beta_1)$ is

$$[\ \exp(0.040), \exp(0.096)\ ] = [\ 1.04, 1.10\ ].$$

Examination of such intervals is a common way of determining whether the association is "significant" – here we have strong evidence that the relative risk associated with AFF is significant.

```
scotland.fit1X$summary.fixed[1:5]
##                     mean         sd  0.025quant    0.5quant  0.975quant
## (Intercept) -0.49181346 0.15947210 -0.81075911 -0.49006442 -0.18279245
## X            0.06836907 0.01423825  0.04034272  0.06836394  0.09642797
scotland.fit1X$summary.hyperpar[1:5]
##                            mean        sd 0.025quant 0.5quant 0.975quant
## Precision for Region 2.941347 0.8361568   1.639126 2.827342   4.838659
```

[1] unlike means since, for example, $E[\exp(\beta_1)|y] \neq \exp(E[\beta_1|y])$.

The posterior median of σ_e is `0.594717677016607` and a 95% interval is

`[0.454608447240487, 0.781076847941202]`

A more interpretable quantity is an interval on the residual relative risk (RRR). The latter follow a lognormal distribution $\text{LogNormal}(0, \sigma_e^2)$ so a 95% interval is $\exp(\pm 1.96 \times \sigma_e)$.

A posterior median of a 95% RRR interval is

$$\begin{aligned} & [\exp(-1.96 \times \text{median}(\sigma_e)), \exp(1.96 \times \text{median}(\sigma_e)] \\ = & [\exp(-1.96 \times 0.595), \exp(1.96 \times 0.595)] = [0.31, 3.2] \end{aligned}$$

which is quite wide.

A more in depth analysis would examine the prior sensitivity to the prior on τ_e.

Variances are in general more difficult to estimate than regression coefficients so there is often sensitivity (unless the number of areas is very large).

Poisson-Lognormal spatial model with a covariate

We now add spatial (ICAR) random effects to the model. We parameterize in terms of total variance and proportion that is spatial.

The model is *Stage 1:* The Likelihood $Y_i|\theta_i \sim \text{Poisson}(E_i\theta_i)$, $i = 1, \dots, n$ with

$$\log \theta_i = \beta_0 + x_i\beta_1 + b_i$$

where x_i is the AFF in area i.

Stage 2: The random effects (prior distribution) is $e_i|\sigma_e^2 \sim_{iid} N(0, \sigma_e^2)$ and the S_i are ICAR. The parameterization is in terms of the total variance σ_b^2 and the proprtion spatial ϕ.

Stage 3: The hyperprior on the hyperparameters $\beta_0, \beta_1, \sigma_b^2, \phi$:

$$p(\beta_0, \beta_1, \sigma_b^2, \phi) = p(\beta_0)p(\beta_1)p(\sigma_b^2)p(\phi)$$

so that here we have assumed independent priors.

We need a graph file containing the neighbors.

```
# Spatial effects with covariate
download.file("http://faculty.washington.edu/jonno/SISMIDmaterial/scotland.graph",destfile = "scotland.g
```

Default specification:

```
formula <- Counts ~ 1 + X +
f(Region, model="bym2",graph="scotland.graph")
scotland.fit2default <- inla(formula, data=Scotland,family="poisson",E=E)
scotland.fit2default$summary.fixed[,1:5]
##                    mean         sd   0.025quant    0.5quant 0.975quant
## (Intercept) -0.11661244 0.11062937 -0.335611320 -0.11602485 0.09900655
## X            0.02656675 0.01147613  0.003690257  0.02668352 0.04878620
scotland.fit2default$summary.hyper[,1:5]
##                           mean         sd 0.025quant  0.5quant 0.975quant
## Precision for Region 4.8292574 1.47092281  2.5616536 4.6225845  8.2990909
## Phi for Region       0.9369205 0.07414969  0.7273244 0.9637913  0.9988012
```

We now place a penalized complexity prior on these two parameters and dot a few more i's.

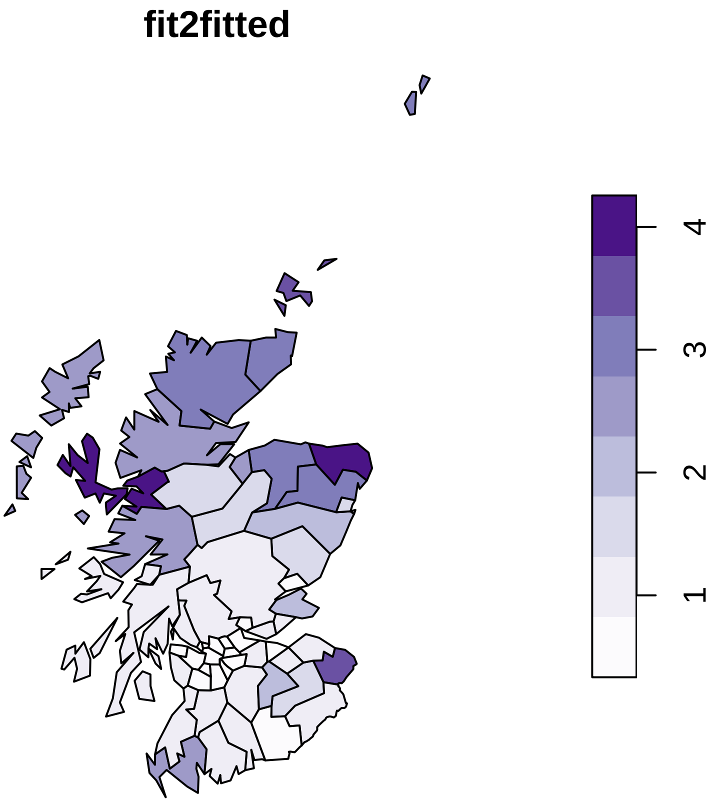
```r
formula <- Counts ~ 1 + X +
f(Region, model="bym2",graph="scotland.graph",
  scale.model=T,
  constr=T,
  rankdef = 1,
  hyper=list(
    phi=list(
      prior="pc",
      param=c(0.5,0.5),
      initial=1),
  prec=list(
    prior="pc.prec",
    param=c(0.5/0.31,0.01),
    initial=5)))
scotland.fit2 <- inla(formula, data=Scotland,
family="poisson",E=E,
control.predictor=list(compute=TRUE),
control.compute=list(return.marginals.predictor=TRUE, config = TRUE))

scotland.fit2$summary.fixed[,1:5]
##                     mean         sd  0.025quant    0.5quant 0.975quant
## (Intercept) -0.11643503 0.11150643 -0.33720324 -0.11583497  0.1008841
## X            0.02635149 0.01158384  0.00325484  0.02647057  0.0487773
scotland.fit2$summary.hyper[,1:5]
##                            mean         sd 0.025quant  0.5quant 0.975quant
## Precision for Region 4.6547885 1.43297449  2.4466567 4.4534284  8.0361878
## Phi for Region       0.9351945 0.07571577  0.7212275 0.9623124  0.9986055
```

Slight differences on the hyperparameters (total precision and proportion spatial) but nothing substantive.

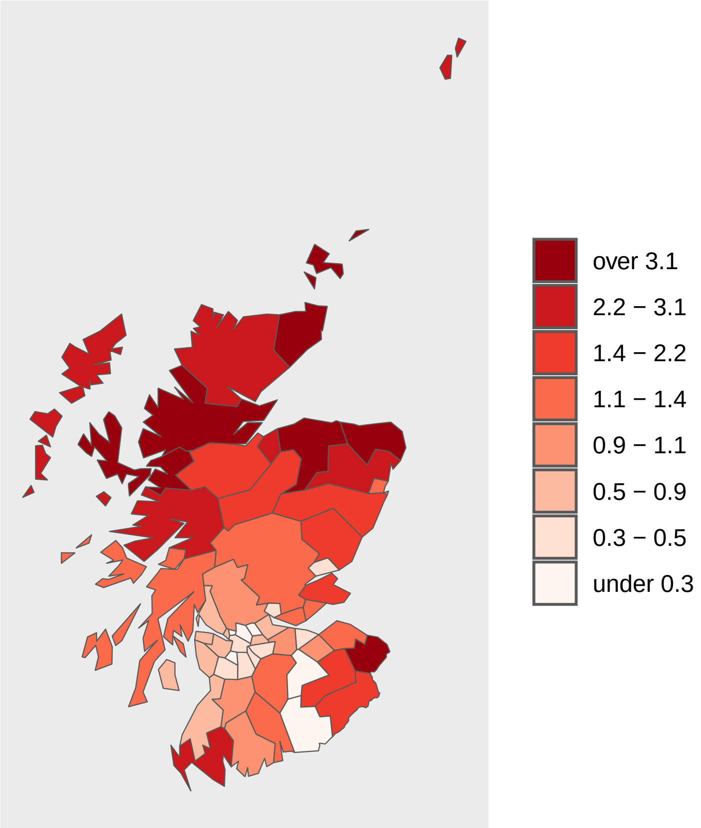
For the user-specified priors: The posterior median of the total standard deviation (on the log relative risk scale) is the posterior median of $1/\sqrt{\tau_b}$ (where τ_b is the precision), which is `0.473863`

The posterior median for the proportion of the residual variation that is spatial ϕ is `0.9623124`.

```r
smap$fit2fitted <- scotland.fit2$summary.fitted.values$`0.5quant`
plot(smap["fit2fitted"], pal = pal, nbreaks=8, breaks = "equal")
```



```
leglabs <- function(vec, under = "under", over = "over", between = "-") {
  x <- vec
  lx <- length(x)
  if (lx < 3) {
    stop("vector too short")
  }
  res <- character(lx - 1)
  res[1] <- paste(under, x[2])
  for (i in 2:(lx - 2)) res[i] <- paste(x[i], between, x[i + 1])
  res[lx - 1] <- paste(over, x[lx - 1])
  res
}

plotvar <- smap$SMR # variable we want to map: SMR
nclr <- 8
plotclr <- brewer.pal(nclr, "Reds")
brks <- round(quantile(plotvar, probs = seq(0, 1, 1/(nclr))), digits = 1)
colornum <- findInterval(plotvar, brks, all.inside = T)
colcode <- plotclr[colornum]

ggplot(smap) +
  geom_sf(aes(fill = colcode)) +
  theme(
    legend.title = element_blank(),
    panel.grid = element_blank(),
    axis.ticks = element_blank(),
    axis.text = element_blank()
  ) +
  theme(legend.position = "right") +
  scale_fill_manual(values = rev(plotclr),
                    labels = rev(leglabs(round(brks, digits = 1)))) +
  coord_sf()
```



Poisson-Lognormal spatial model with covariates

Now we provide maps of the non-spatial and spatial random effects.

Estimates of residual relative risk (posterior medians), of the non-spatial e^{e_i} and the spatial contributions e^{S_i}.

The BYM2 formulation for the random effect is $b_i = S_i + e_i$ where S_i is spatial and e_i is IID. INLA stores b_i (the first 56 rows) and S_i (the next 56 rows) and so we find the non-spatial via $e_i = b_i - S_i$.

Note the differences in the scales: the spatial random effects dominate here.

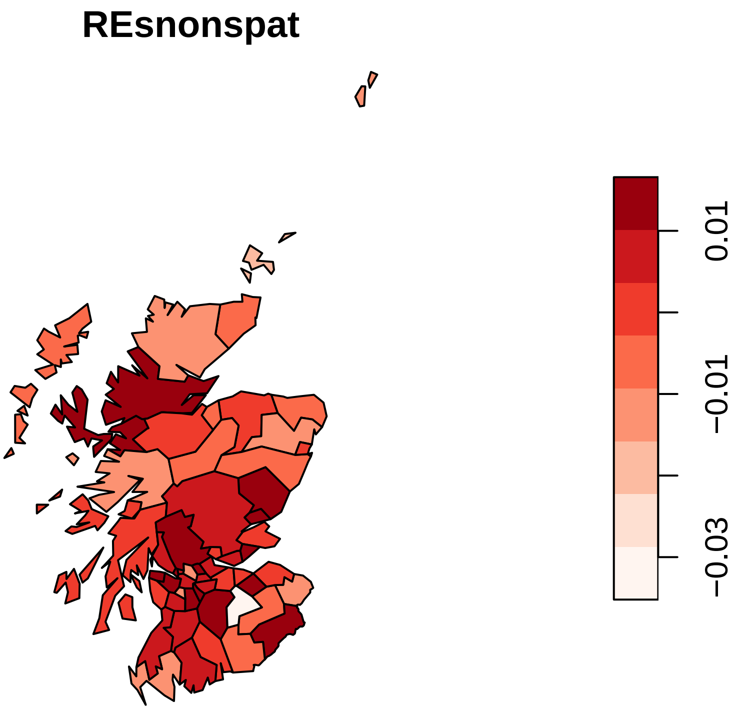
```
samp <- inla.posterior.sample(n = 1000, scotland.fit2)
samp_mat <- matrix(0, nrow = 1000, ncol = 2)
for (i in 1:1000) {
  samp_mat[i,] <- samp[[i]]$hyperpar[1:2]
}
scale_region <- mean(sqrt(samp_mat[,2])/sqrt(samp_mat[,1]))
```

Poisson-Lognormal spatial model with covariates: non-spatial random effects

```
# obtain RE estimates
N <- 56
struct <- scotland.fit2$summary.random[[1]]$mean[(N+1):(N*2)]
combined <- scotland.fit2$summary.random[[1]]$mean[1:N]
struct <- struct * scale_region
iid <- combined - struct
REsnonspat <- iid
REsspat <- struct
scotdata$REsnonspat <- iid
scotdata$REsspat <- struct
```

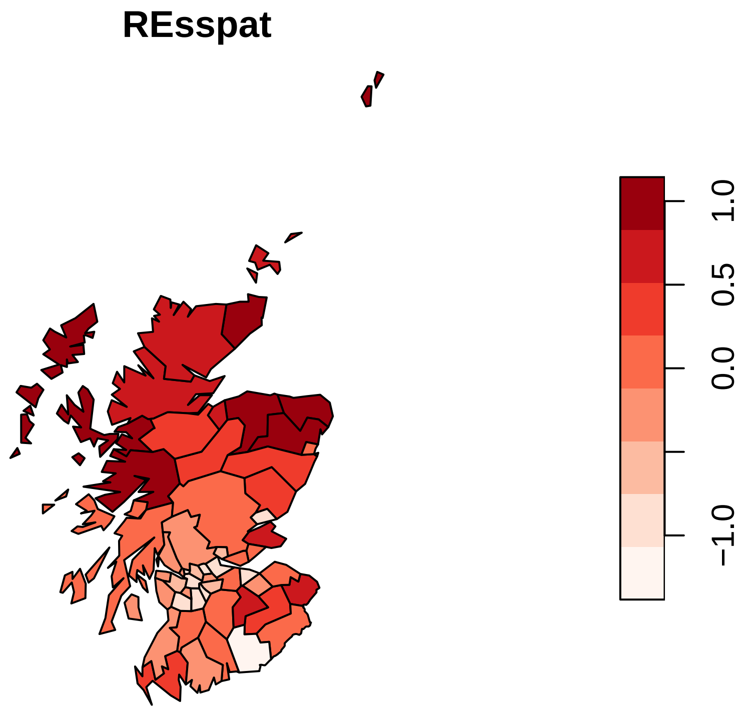
Non-spatial random effects:

```
smap$REsnonspat = scotdata$REsnonspat
pal = function(n) brewer.pal(n,"Reds")
plot(smap["REsnonspat"],pal = pal,nbreaks =8, breaks = "equal")
```



Spatial random effects:

```
smap$REsspat = scotdata$REsspat
plot(smap["REsspat"], pal = pal, nbreaks=8, breaks = "equal")
```



Exceedance probabilities

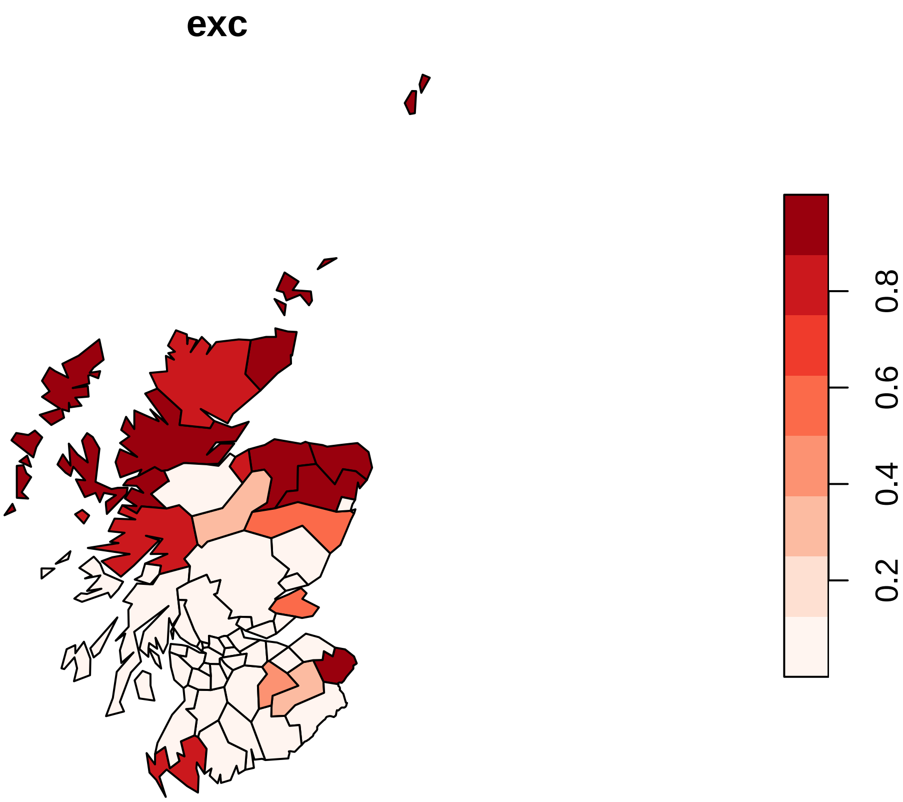
A useful summmary is the posterior probability of excedance of epidemiologically ineresting thresholds.

Below we map the posterior probabilities

$$\Pr(\theta_i > 2|y),$$

for $i = 1, \ldots, 56$.

```
exc <- sapply( scotland.fit2$marginals.fitted.values,
FUN = function(marg){1 - inla.pmarginal(q = 2, marginal = marg)})
smap$exc <- exc
plot(smap["exc"],pal = pal, nbreaks = 8, breaks = "equal")
```



Spatial model: confounding by location

The command `plot(scotland.fit2)` provides plots of: marginal posterior distributions of β_0, β_1, σ_e^{-2}, σ_S^{-2} and summaries of the random effects e_i, S_i and the linear predictors and fitted values, all by area.

Note that the posterior mean estimate of β_1 associated with AFF goes from 0.068 → 0.026 when moving from the non-spatial to spatial model.

This is known as confounding by location.

The model attributes spatial variability in risk to either the covariate or to the spatial random effects.

The posterior median estimate of σ_e decreases from $1/\sqrt{2.9475} = 0.58$ to $1/\sqrt{94.986} = 0.10$ when the spatial random effect is added.

The posterior median estimate of σ_s is $1/\sqrt{1.125} = 0.94$ but, as already noted, this value is not directly comparable to the estimate of σ_e.

However, the scales on the figures shows that the spatial component dominates for these data.

A rough estimate of the standard deviation of the spatial component can be determined by empirically calculating the standard deviation of the random effect estimates $\widehat{S}_i$.

A more complete analysis would address the sensitivity to the prior specifications on σ_e and σ_s.

INLA Graph File Creation

The code below creates a neighborhood file for INLA that looks like:

39

1 4 11 13 22 38 2 2 12 38 3 5 11 13 20 36 39 4 6 9 17 19 24 29 31

...

38 7 1 2 7 11 12 22 32

39 8 3 13 17 19 20 21 27 30

Creating an INLA graph file from a shapefile

```
library(spdep) # for poly2nb and nb2inla
download.file("http://faculty.washington.edu/jonno/SISMIDmaterial/wacounty.shp",destfile = "wacounty.shp
download.file("http://faculty.washington.edu/jonno/SISMIDmaterial/wacounty.shx",destfile = "wacounty.sh
download.file("http://faculty.washington.edu/jonno/SISMIDmaterial/wacounty.dbf",destfile = "wacounty.db
countymap=st_read(dsn=".",layer = "wacounty")
## Reading layer `wacounty' from data source
##   `/Users/andreaboskovic/Desktop/uw classes/year2/winter 23/554/STAB/2023-554'
##   using driver `ESRI Shapefile'
## Simple feature collection with 39 features and 6 fields (with 1 geometry empty)
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -124.7312 ymin: 45.5434 xmax: -116.915 ymax: 49.0026
## CRS:           NA
countymap = countymap[!st_is_empty(countymap),]
nb.map <- poly2nb(countymap)
nb2INLA("wacounty.graph",nb.map)
```

Log relative risks as the outcome variable

Often the data arise in the form of observed rates or observed relative risks. For illustration, we imagine we had received the latter for Scotland, rather than the full data.

We model the log relative risk directly assuming they have a Gaussian distribution. We define $Z_i = \log \widehat{\theta}_i$ to emphasize that the observed data are now taken to be the log relative risks.

Recall that if any of the counts $Y_i = 0$ (which would result in a relative risk of zero and a standard error of zero), we can use the approximations $Y_i^\star = Y_i + 0.5$ and $E_i^\star = E_i + 0.5$ to calculate $\widehat{\theta}_i^\star = \frac{Y_i^\star}{E_i^\star}$. In these cases, $Z_i = \log \widehat{\theta}_i^\star$. For simplicity, we assume this has been done.

In INLA, we can fit the model (with *'s if necessary)

$$Z_i = \log\left(\frac{Y_i}{E_i}\right) \sim \mathrm{N}(\mu_i, \sigma^2)$$

where $\mu_i = E[Z_i]$.

INLA estimates the precision for the Gaussian observations, $1/\sigma^2$. We evaluate the variance of the observed “data’’ by using the Poisson variance assumption —the mean equals the variance (in general the standard error of the rate can be estimated in a variety of ways, including the jackknife).

Therefore, the Z_i have “known’’ variances that we can approximate using the Delta method (as we did previously) as

$$\widehat{\mathrm{var}}(Z_i) = [E_i \exp(Z_i)]^{-1} = \frac{1}{E_i \widehat{\theta}_i}$$

with *'s if needed.

Modeling the log relative risk as normal

We calculate the log relative risks as $Z_i = \log \widehat{\theta^\star}_i$ and their variances for the Scotland lip cancer data.

```
Scotland$Ystar <- ifelse(Scotland$Counts == 0,
                         Scotland$Counts + 0.5,
                         Scotland$Counts)
Scotland$Estar <- ifelse(Scotland$Counts == 0,
                         Scotland$E + 0.5,
                         Scotland$E)
Scotland$thetastar <- Scotland$Ystar/Scotland$Estar
Scotland$Z <- log(Scotland$thetastar)
Scotland$varZ <- 1/(Scotland$Estar * Scotland$thetastar)
Scotland$precZ <- 1/Scotland$varZ
```

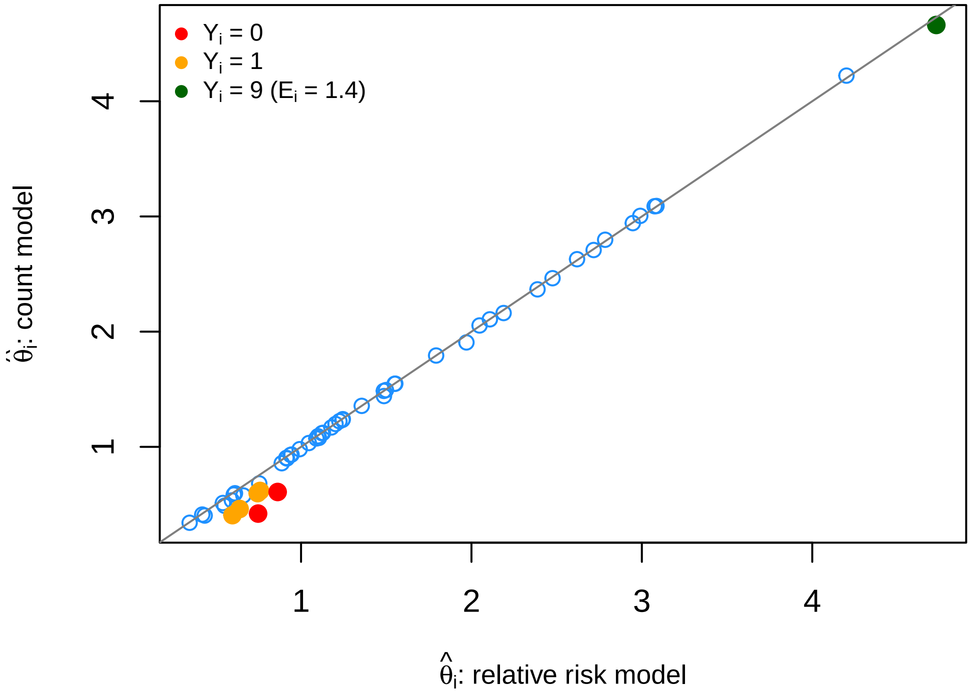
We can fit a normal model for the log relative risks (the likelihood) in INLA with fixed normal precisions (equivalent to known variance) using the following code.

```
pcprec <- list(theta=list(prior='pc.prec',param=c(1,.05)))
scotland.fit3a <- inla(Z ~ 1 + f(Region, model="iid", hyper=pcprec),
  data = Scotland,family="gaussian",control.predictor=list(compute=TRUE),
  control.family = list(hyper = list(prec = list(initial = log(1), fixed=TRUE))),
  scale=precZ)
```

Note the `scale=precZ` which along with the previous line, fixes the measurement error variances.

We now compare the fits of the Poisson-Lognormal count outcome model and relative risk outcome models

```
par(mfrow = c(1,1), mar=c(5,4,1,1))
plot(scotland.fit1$summary.fitted.values[,4] ~ exp(scotland.fit3a$summary.fitted.values[,4]),
     col = "dodgerblue", ylab = expression(paste(hat(theta)[i],": count model")),
     xlab = expression(paste(hat(theta)[i],": relative risk model")), cex.lab = 0.85)
points(exp(scotland.fit3a$summary.fitted.values[,4][which(Scotland$Counts == 1)]),
       scotland.fit1$summary.fitted.values[,4][which(Scotland$Counts == 1)],
       col = "orange", pch = 19, cex = 1.15)
points(exp(scotland.fit3a$summary.fitted.values[,4][which(Scotland$Counts == 0)]),
       scotland.fit1$summary.fitted.values[,4][which(Scotland$Counts == 0)],
       col = "red", pch = 19, cex = 1.15)
points(exp(scotland.fit3a$summary.fitted.values[,4][which(Scotland$thetastar > 6)]),
       scotland.fit1$summary.fitted.values[,4][which(Scotland$thetastar > 6)],
       col = "darkgreen", pch = 19, cex = 1.15)
abline(0, 1, col = "gray50")
legend("topleft",legend = c(expression(paste(Y[i], " = ", 0)),
                            expression(paste(Y[i], " = ", 1)),
                            expression(paste(Y[i], " = ", 9, " (", E[i], " = ", 1.4, ")"))),
       pch = c(19, 19, 19), col = c("red", "orange", "darkgreen"), cex = 0.75, bty="n")
```



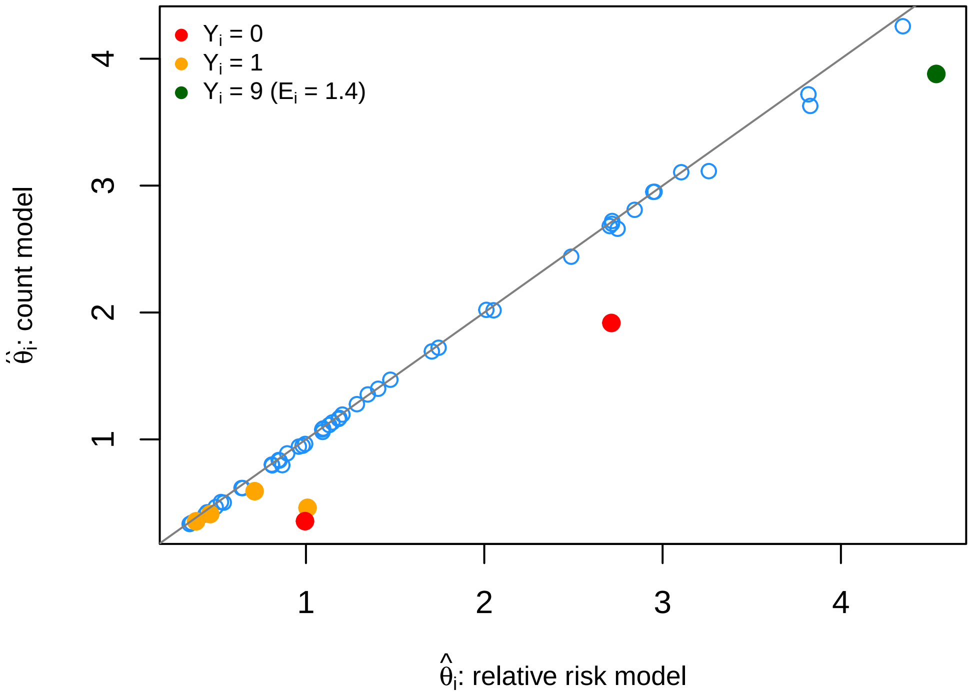
We fit a spatial normal log relative risk model with IID Normal random effects with a covariate

Here, we add in a covariate and the spatial random effects

```
formula <- Z ~ 1 + I(X) +
  f(Region, model="bym2",graph="scotland.graph",
    scale.model=T,
    constr=T,
    hyper=list(
      phi=list(
        prior="pc",
        param=c(0.5,0.5),
        initial=1),
      prec=list(
        prior="pc.prec",
        param=c(0.5/0.31,0.01),
        initial=5)))
scotland.fit3 <- inla(formula, data=Scotland,
family="gaussian",control.predictor=list(compute=TRUE),
control.family = list(hyper = list(prec = list(initial = log(1), fixed=TRUE))),
scale=precZ)
```

Comparison of spatial Poisson-Lognormal count outcome and relative risk outcome fits: differences in low and high extremes

```
par(mfrow = c(1,1), mar=c(5,4,1,1))
plot(scotland.fit2$summary.fitted.values[,4] ~ exp(scotland.fit3$summary.fitted.values[,4]),
     col = "dodgerblue", ylab = expression(paste(hat(theta)[i],": count model")),
     xlab = expression(paste(hat(theta)[i],": relative risk model")), cex.lab = 0.85)
points(exp(scotland.fit3$summary.fitted.values[,4][which(Scotland$Counts == 1)]),
       scotland.fit2$summary.fitted.values[,4][which(Scotland$Counts == 1)],
       col = "orange", pch = 19, cex = 1.15)
points(exp(scotland.fit3$summary.fitted.values[,4][which(Scotland$Counts == 0)]),
       scotland.fit2$summary.fitted.values[,4][which(Scotland$Counts == 0)],
       col = "red", pch = 19, cex = 1.15)
points(exp(scotland.fit3$summary.fitted.values[,4][which(Scotland$thetastar > 6)]),
       scotland.fit2$summary.fitted.values[,4][which(Scotland$thetastar > 6)],
       col = "darkgreen", pch = 19, cex = 1.15)
abline(0, 1, col = "gray50")
legend("topleft",legend = c(expression(paste(Y[i], " = ", 0)),
                            expression(paste(Y[i], " = ", 1)),
                            expression(paste(Y[i], " = ", 9, " (", E[i], " = ", 1.4, ")"))),
       pch = c(19, 19, 19), col = c("red", "orange", "darkgreen"), cex = 0.75, bty="n")
```



Regression coefficient comparison: very similar estimates (and posterior uncertainty estimates) of the regression coefficients:

Count model:

```
scotland.fit2$summary.fixed
##                    mean         sd  0.025quant    0.5quant 0.975quant mode
## (Intercept) -0.11643503 0.11150643 -0.33720324 -0.11583497  0.1008841   NA
## X            0.02635149 0.01158384  0.00325484  0.02647057  0.0487773   NA
##                      kld
## (Intercept) 7.717508e-07
## X           7.211222e-07
```

Relative risk model:

```
##                    mean         sd    0.025quant    0.5quant 0.975quant mode
## (Intercept) -0.01480696 0.11073569 -0.232583955 -0.01470891 0.20236488   NA
## I(X)         0.02501478 0.01138381  0.002442484  0.02509030 0.04716296   NA
##                       kld
## (Intercept) 1.288365e-09
## I(X)        5.110380e-09
```

Missing area data in Scotland

As an illustration we suppose that for the last area Y_{56} is unobserved – it is coded as `NA` (its value is zero in the data).

The missing value can be imputed with the spatial ICAR model helping in this respect.

If the count was missing because low (e.g., not released because less than 5) then this is informative and the following analysis is not approprtiate.

```
Scotland$CountsNA <- Scotland$Counts
Scotland$CountsNA[56] <- NA
scotland.fitNA <- inla(CountsNA ~ 1 + I(X) +
f(Region, model="bym2",graph="scotland.graph",
  scale.model=T,
  constr=T,
  rankdef = 1,
  hyper=list(
    phi=list(
      prior="pc",
      param=c(0.5,0.5),
      initial=1),
  prec=list(
    prior="pc.prec",
    param=c(0.3/0.31,0.01),
    initial=5))),data=Scotland,
family="poisson",E=E,control.predictor=list(compute=TRUE,link=1))
```

```
summary(scotland.fitNA)
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts,
##    ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##    scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##    ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##    verbose, ", " lincomb = lincomb, selection = selection, control.compute
##    = control.compute, ", " control.predictor = control.predictor,
##    control.family = control.family, ", " control.inla = control.inla,
##    control.fixed = control.fixed, ", " control.mode = control.mode,
##    control.expert = control.expert, ", " control.hazard = control.hazard,
##    control.lincomb = control.lincomb, ", " control.update =
##    control.update, control.lp.scale = control.lp.scale, ", "
##    control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##    ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##    num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
##    working.directory = working.directory, ", " silent = silent, inla.mode
##    = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
##    .parent.frame)")
## Time used:
```

```
##     Pre = 17.9, Running = 0.414, Post = 0.0209, Total = 18.3
## Fixed effects:
##                mean    sd 0.025quant 0.5quant 0.975quant mode kld
## (Intercept) -0.082 0.110     -0.300   -0.082      0.133   NA   0
## I(X)         0.024 0.011      0.002    0.024      0.046   NA   0
##
## Random effects:
##   Name     Model
##     Region BYM2 model
##
## Model hyperparameters:
##                         mean    sd 0.025quant 0.5quant 0.975quant mode
## Precision for Region 4.982 1.478      2.683    4.780      8.454   NA
## Phi for Region       0.948 0.064      0.768    0.972      0.999   NA
##
## Marginal log-Likelihood:  -126.75
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

From the graph file we see that area 56 has areas 2,3,4,5 as neighbors – we look at these values and see the SMRs are high, which explains why the predictive mean is high.

We include the prediction of the rate from the model in which the data are observed and see it is much lower.

We look at the last line of the graph file (for area 56):

```
56 4 2 3 4 5
```

We examine the SMRs from the 4 neighboring areas, and they are high.

The covariate value for area 56 (which is also used in the prediction) is mid-range.

We obtain the expected count of the missing area response and compare with the true response (which of course we know in this exercise)

```
scotland.fitNA$summary.fitted.values[56,]
##                         mean       sd 0.025quant 0.5quant 0.975quant mode
## fitted.Predictor.56 3.273038 1.260357   1.490069 3.052708   6.343922   NA
Scotland$E[56]
## [1] 1.8
Scotland$X[56]
## [1] 10
set56 <- c(2,3,4,5)
Scotland$Counts[set56]
## [1] 39 11  9 15
Scotland$E[set56]
## [1] 8.7 3.0 2.5 4.3
Scotland$Counts[set56]/Scotland$E[set56]
## [1] 4.482759 3.666667 3.600000 3.488372
# Compare with fit from model in which Y[56]=0 is used as observed
scotland.fit2$summary.fitted.values[56,]
##                         mean        sd 0.025quant 0.5quant 0.975quant mode
## fitted.Predictor.56 1.988202 0.6240187   0.975791 1.917543   3.409087   NA
```

Much lower predictor because the zero pulls down.

In the model

$$Y_i|\theta_i \sim Poisson(E_i\theta_i),$$

the fitted values/predictions are for θ_i, so no expected number and no Poisson sampling (so we're predicting the the relative risk).

We confirm that the quantiles of the fitted values are the exponentiated predictions.

```
scotland.fitNA$summary.fitted.values[56,c(3:5)]
##                       0.025quant 0.5quant 0.975quant
## fitted.Predictor.56     1.490069 3.052708   6.343922
exp(scotland.fitNA$summary.linear.predictor[56,c(3:5)])
##                0.025quant 0.5quant 0.975quant
## Predictor.56     1.490069 3.052708   6.343922
```

Censored Poisson Data in Scotland

Sometimes, data come in a form where the smallest counts are censored and we only know that a count is below a certain threshold. For example, due to privacy concerns data with small counts in.

To illustrate, we will censor the counts in the Scotland data that are below 2.

```
Scotland$CountsCen <- ifelse(Scotland$Counts < 2, 0, Scotland$Counts)
```

We now show how to implement the censored Poisson model in INLA.

We will fit a Poisson-Lognormal model with a covariate and spatial REs, using PC priors.

```
# fit a Lognormal model with a censored Poisson family
#   with the censoring interval set to anything less than 2
scotland.fit.cen <- inla(CountsCen ~ 1 + I(X) +
                           f(Region, model="bym2",graph="scotland.graph",
                             scale.model=T,
                             constr=T,
                             rankdef = 1,
                             hyper=list(
                               phi=list(
                                 prior="pc",
                                 param=c(0.5,0.5),
                                 initial=1),
                               prec=list(
                                 prior="pc.prec",
                                 param=c(0.3/0.31,0.01),
                                 initial=5))),
                         family = "cenpoisson",
                         control.family = list(cenpoisson.I = c(0,1)),
                         E=E,
                         data = Scotland,
                         control.predictor=list(compute=TRUE,link=1))
```

We examine the regression coefficients for the censored and uncensored data.

Censored data:

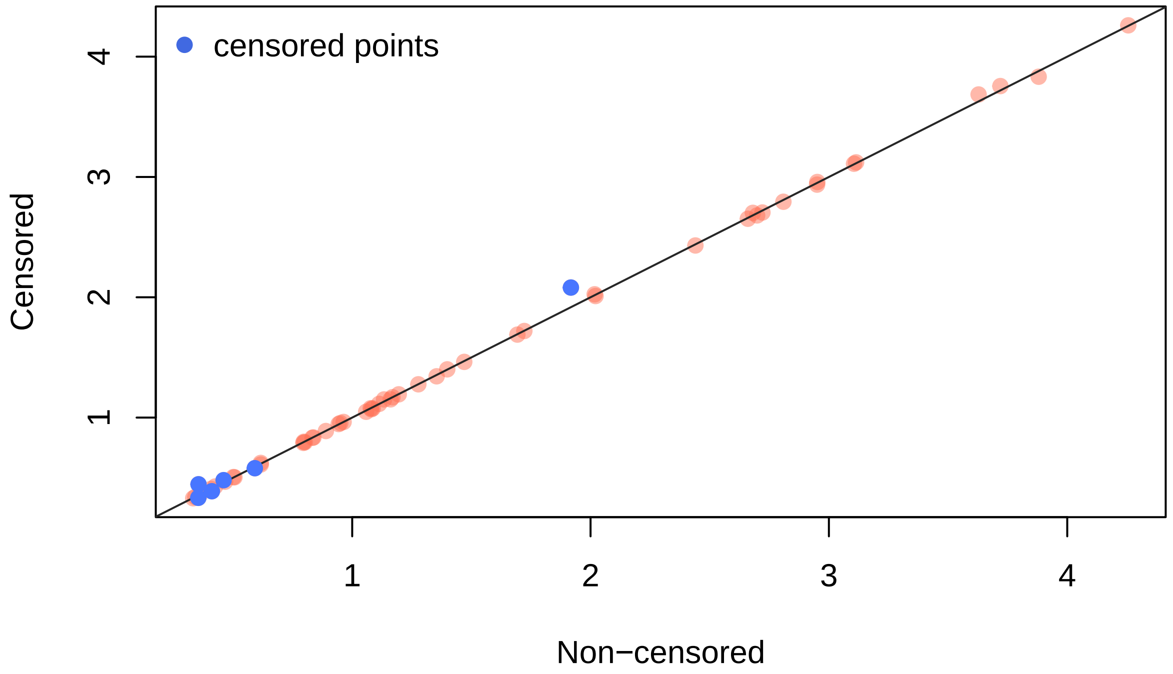
```
scotland.fit.cen$summary.fixed
##                    mean         sd   0.025quant    0.5quant 0.975quant mode
## (Intercept) -0.12131705 0.11040086 -0.340069357 -0.12066426 0.09368047   NA
## I(X)         0.02726738 0.01140925  0.004518384  0.02738523 0.04935410   NA
##                      kld
## (Intercept) 7.259966e-07
## I(X)        7.226271e-07
```

Uncensored data:

```
scotland.fit2$summary.fixed
##                    mean         sd  0.025quant    0.5quant 0.975quant mode
## (Intercept) -0.11643503 0.11150643 -0.33720324 -0.11583497  0.1008841   NA
## X            0.02635149 0.01158384  0.00325484  0.02647057  0.0487773   NA
##                      kld
## (Intercept) 7.717508e-07
## X           7.211222e-07
```

We compare fitted values with the model fitted to the uncensored data

```
cen_est <- scotland.fit.cen$summary.fitted.values$`0.5quant`
non_cen_est <- scotland.fit2$summary.fitted.values$`0.5quant`
par(mfrow=c(1,1))
plot(cen_est ~ non_cen_est,
     col = alpha("coral1",0.5),pch=19,
       xlab="Non-censored",ylab="Censored")
abline(0,1,col="gray15")
points(non_cen_est[Scotland$CountsCen==0],
       cen_est[Scotland$CountsCen==0],
       col = "royalblue1",pch=19)
legend("topleft",c("censored points"),pch=c(19), col=c("royalblue"),bty="n")
```



PC prior details

For a precision in the model $x|\tau \sim N(0, 1/\tau)$, the PC prior is obtained via the following rationale:

- The prior on the sd is exponential with rate λ, which we need to specify
- The exponential leads to a type-2 Gumbel on the precision (change of variables)
- Hence we have the model:

$$\begin{aligned} x|\tau &\sim N(0, 1/\tau) \\ \tau &\sim \text{Gumbel}(\lambda) \end{aligned}$$

- If we integrate out τ, we can find the marginal sd of x
- For more details see Simpson et al (2017, p. 9, top of right column) and Bakka et al (2018).

The PC prior for ϕ in the BYM2 random effect is more complex, and does not have a nice distribution. However, it retains the interpretation of a PC prior, which is why it useful. See details in Simpson et al. (2017).