

SISMID Spatial Statistics in Epidemiology and Public Health

2016 R Notes: Disease Mapping

Jon Wakefield
Departments of Statistics and Biostatistics, University of
Washington

2016-07-03

Scottish lip cancer data

We will first fit a number of models to the famous Scottish lip cancer data.

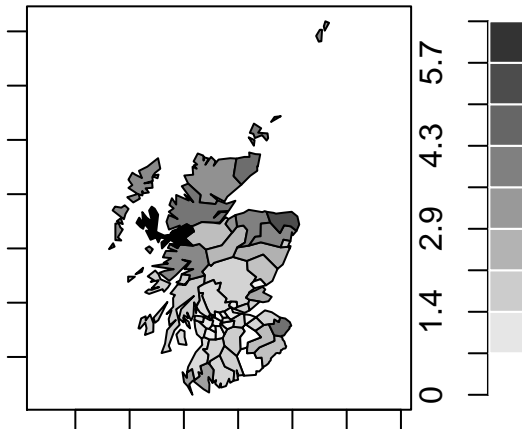
We have counts of disease, expected numbers and an area-based covariate (proportion in agriculture, fishing and farming) in each of 56 areas.

```
library(SpatialEpi)
data(scotland)
Y <- scotland$data$cases
X <- scotland$data$AFF
E <- scotland$data$expected
# Relative risk estimates
smr <- Y/E
summary(E)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.100   4.050   6.300   9.575  10.120  88.700
summary(smr)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.496   1.111   1.522   2.241   6.429
```

Scottish lip cancer data

The SMRs have a large spread with an increasing trend in the south-north direction.

```
scotland.map <- scotland$spatial.polygon  
par(mar = c(1, 1, 1, 1))  
mapvariable(smr, scotland.map) # Function in SpatialEpi
```



Scottish lip cancer data

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.

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).

Expected numbers for Scottish lip cancer data

```
par(mar = c(1, 1, 1, 1))  
mapvariable(E, scotland.map)
```

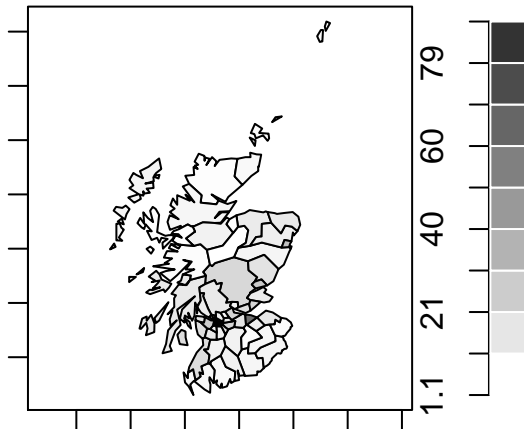
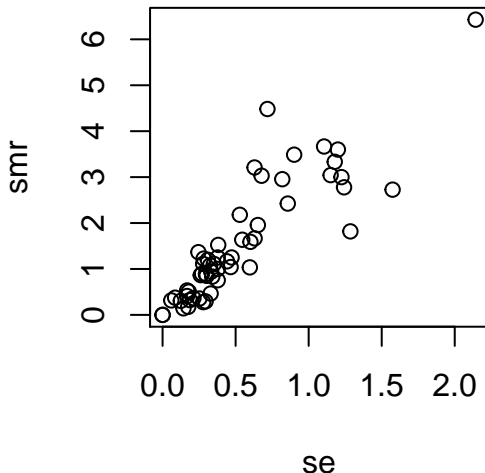


Figure 2: Expected numbers for Scottish lip cancer data

SMR for Scottish lip cancer data

The highest SMRs tend to have the largest standard errors.

```
se <- sqrt(smr/E)  
plot(smr ~ se)
```



Gamma smoothing model

In the gamma model it is assumed that

$$Y_i | \theta_i \sim_{iid} \text{Poisson}(E_i \exp(\alpha) \theta_i)$$

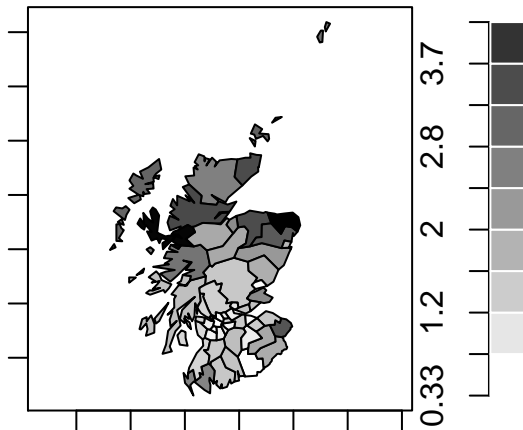
with the relative risks

$$\theta_i \sim_{iid} \text{Ga}(a, a).$$

```
# Gamma random effects model
ebresults <- eBayes(Y, E)
ebresults$alpha # the estimate of a in our model
## [1] 1.87949
ebresults$beta # the estimate of alpha in our model
## (Intercept)
## 0.3521065
# the RRs are greater in the study region than in
# the reference region
summary(ebresults$RR)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3319  0.6079  1.1530  1.4220  2.0290  4.0790
# The smoothed estimates have a narrower range
```

Relative risk estimates

```
par(mar = c(1, 1, 1, 1))  
mapvariable(ebresults$RR, scotland.map)
```



Assessment of gamma assumption

Below we assess the gamma assumption.

The ordered observed residual relative risks $\frac{Y_i/E_i}{e^\alpha}$ are plotted against those expected from the gamma distribution. Let f_i be the ordered expected relative risk if n samples were taken from a gamma distribution.

If the gamma distribution is appropriate the points should lie approximately on a straight line.

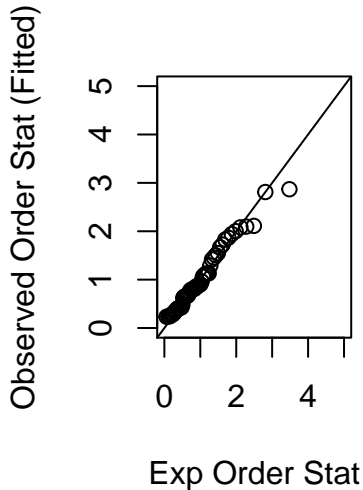
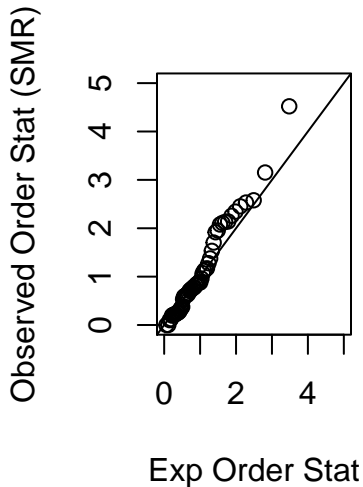
We can also plot the estimated relative risks \widehat{RR}_i under the model against f_i .

Note that the estimates have been constructed under the assumption of gamma random effects, which can make the estimates look more like a gamma sample than they really are.

Assessment of gamma assumption

```
# Now let's examine the gamma assumption
egamma <- qgamma(seq(0.5, length(Y), 1)/length(Y),
  ebresults$alpha, ebresults$alpha)
par(mfrow = c(1, 2))
# First plot is the estimates from the gamma model
plot(egamma, exp(-ebresults$beta) * sort(smr), xlim = c(0,
  5), ylim = c(0, 5), xlab = "Exp Order Stat", ylab = "Obs Ord
abline(0, 1)
# Second plot is the SMR estimates
plot(egamma, exp(-ebresults$beta) * sort(ebresults$RR),
  xlim = c(0, 5), ylim = c(0, 5), xlab = "Exp Order Stat",
  ylab = "Obs Order Stat (Gamma)")
abline(0, 1)
```

Assessment of gamma assumption: looks reasonable



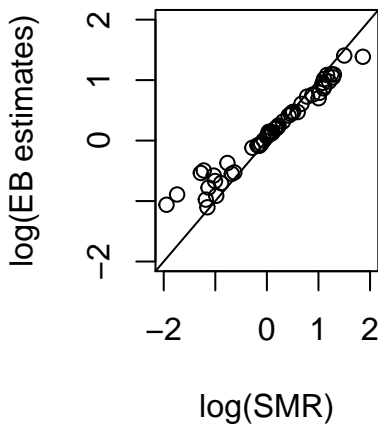
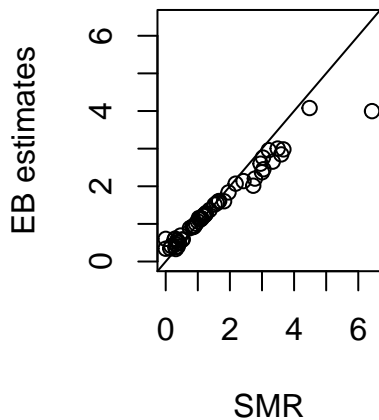
Gamma smoothing model

Below we give code to compare the SMRs and empirical Bayes gamma smoothed estimates.

We plot on the original and on the log scale

```
# Comparison of point estimates
par(mfrow = c(1, 2))
plot(smr, ebresults$RR, xlab = "SMR", ylab = "EB estimates",
     xlim = c(0, max(smr)), ylim = c(0, max(smr)))
abline(0, 1)
plot(log(smr[1:54]), log(ebresults$RR[1:54]), xlim = c(-2,
     2), ylim = c(-2, 2), xlab = "log(SMR)", ylab = "log(EB estim
abline(0, 1)
```

Gamma smoothed estimates are shrunk



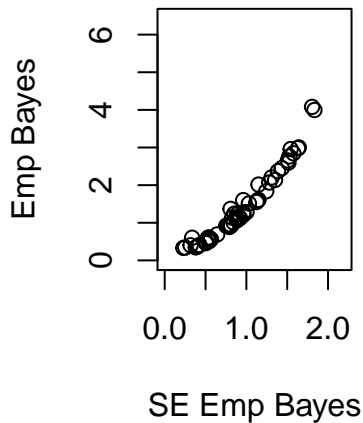
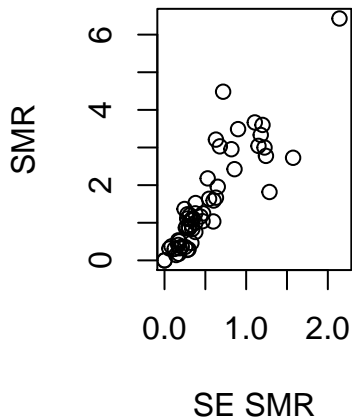
Gamma smoothing model

We repeat the earlier plot of SMRs versus standard errors of SMRs, and also empirical smoothed Bayes estimates versus their standard error.

Again the shrinkage of the Bayes estimates is apparent, and the standard errors are smoothed also (note no zeroes on the right).

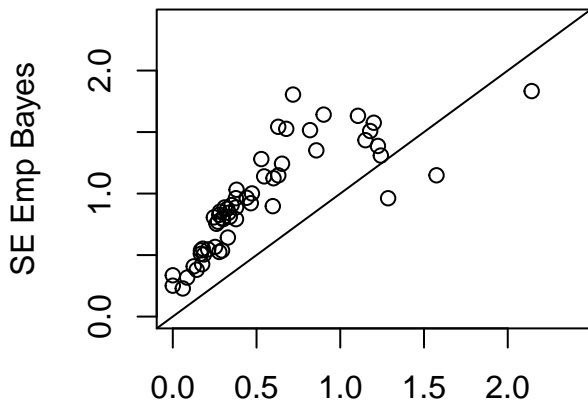
```
seEBests <- sqrt((ebresults$beta + Y) * exp(2 * ebresults$beta)/
  E * ebresults$beta)^2)
par(mfrow = c(1, 2))
plot(se, smr, ylim = c(0, max(smr)), xlim = c(0, 2.2),
  xlab = "SE SMR", ylab = "SMR")
plot(seEBests, ebresults$RR, ylim = c(0, max(smr)),
  xlim = c(0, 2.2), xlab = "SE Emp Bayes", ylab = "Emp Bayes")
```

Gamma smoothing model



Gamma smoothing model: comparison of standard errors

```
plot(seEBests ~ se, ylab = "SE Emp Bayes",  
     xlab = "SE SMR", xlim = c(0, 2.4), ylim = c(0,  
         2.4))  
abline(0, 1)
```



Gamma smoothing model

Given that the posterior distributions of the relative risks have a known distribution (which is gamma, with known α and a), we can examine the complete distribution.

Previously we looked at the posterior means, below we look at the distributions for areas 1 and 56.

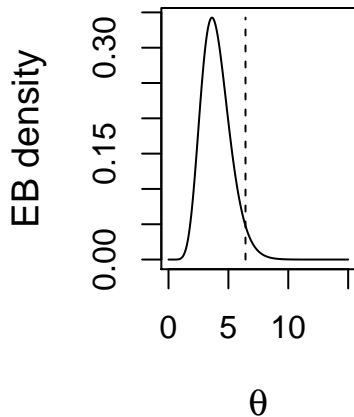
The function `EBpostdens` is in the `SpatialEpi` package.

```
# Densities for areas 1 and 56
par(mfrow = c(1, 2))
EBpostdens(Y[1], E[1], eresults$alpha, eresults$beta,
  lower = 0, upper = 15, main = "Area 1")
EBpostdens(Y[56], E[56], eresults$alpha, eresults$beta,
  lower = 0, upper = 15, main = "Area 56")
```

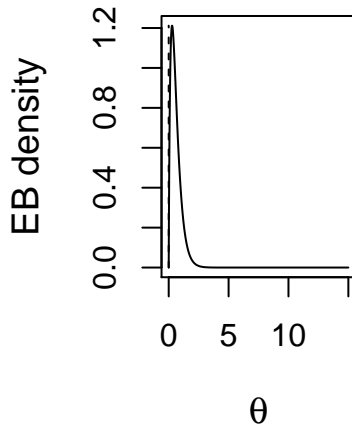
The SMRs are indicated as dashed line - note the zero on the right.

Posterior distributions of relative risks

Area 1



Area 56



Posterior probabilities of exceeding a threshold

We calculate the posterior probability that the relative risk exceeds a value of 3 (this value was chosen for illustration).

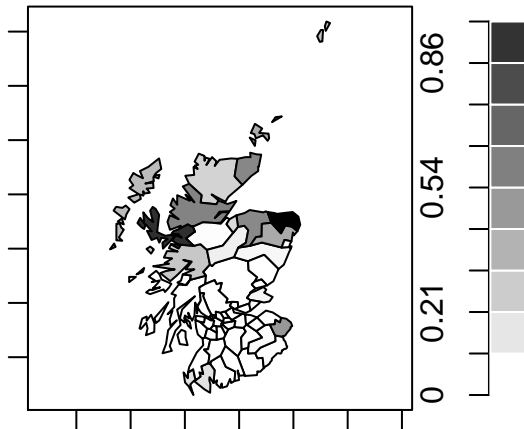
We then map these probabilities.

The function `EBpostthresh` is in the `SpatialEpi` package.

```
par(mfrow = c(1, 1), mar = c(1, 1, 1, 1))
thresh3 <- EBpostthresh(Y, E, alpha = ebresults$alpha,
  beta = ebresults$beta, rrthresh = 3)
mapvariable(thresh3, scotland.map)
```

Posterior probabilities of exceeding a threshold

Two areas in particular have high values.



Gamma smoothing model with covariate

We fit the model with the covariate AFF:

$$Y_i | \theta_i \sim_{iid} \text{Poisson}[E_i \theta_i]$$

with

$$\theta_i = \exp(\alpha + \beta X_i) \delta_i,$$

and $\delta_i \sim_{iid} \text{Ga}(a, a)$.

```
# Now with AFF
ebresultsX <- eBayes(Y, E, X)
ebresultsX$alpha
## [1] 2.98428
ebresults$alpha
## [1] 1.87949
# note the reduction in excess-Poisson variation
# compared to the no covariate model
ebresultsX$beta
## (Intercept)          Xmat
## -0.3527686    7.1481551
```

Gamma smoothing model with covariate

We fit the model with the covariate AFF below and plot the fitted line versus the data.

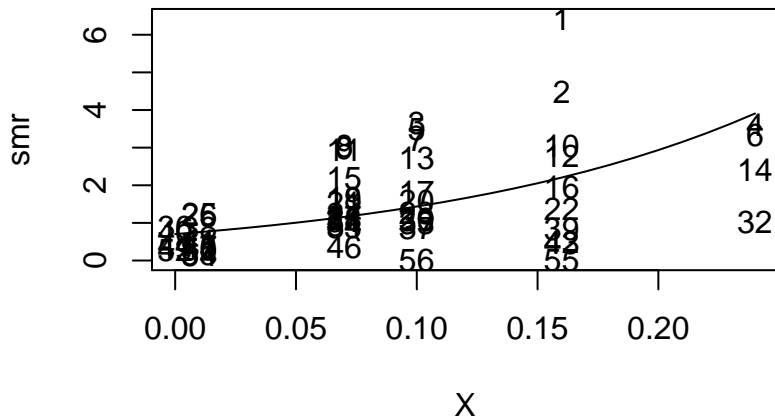
```
# Now with AFF
ebresultsX <- eBayes(Y, E, X)
par(mfrow = c(1, 1))
plot(X, smr, type = "n")
text(X, smr)
xval <- seq(0, max(X), 0.01)
lines(xval, exp(ebresultsX$beta[1] + ebresultsX$beta[2] *
  xval))
```

Gamma smoothing model

Recall the variance of the $\text{Ga}(a, a)$ is $1/a$ (the a parameter is labeled alpha here).

In the above, the variance of the random effects is $1/1.88 = 0.53$ without AFF and $1/2.98 = 0.34$ with AFF, showing the variability “explained” by the covariate.

Gamma smoothing model with covariate



Comparison of models

We compare the SMRs with the gamma estimates with and without covariates.

```
x0 <- rep(0, length(Y))
x1 <- rep(1, length(Y))
x2 <- rep(2, length(Y))
plot(x0, smr, xlim = c(0, 2), type = "n", ylab = "RELATIVE RISK",
     xlab = "", axes = F)
axis(2)
axis(1)
text(x0, smr)
text(x1, ebresults$RR)
text(x2, ebresultsX$RR)
for (i in 1:length(Y)) {
  lines(c(0, 1, 2), c(smr[i], ebresults$RR[i], ebresultsX$RR[i]),
        lty = 2)
}
abline(1, 0)
```

Comparison of models

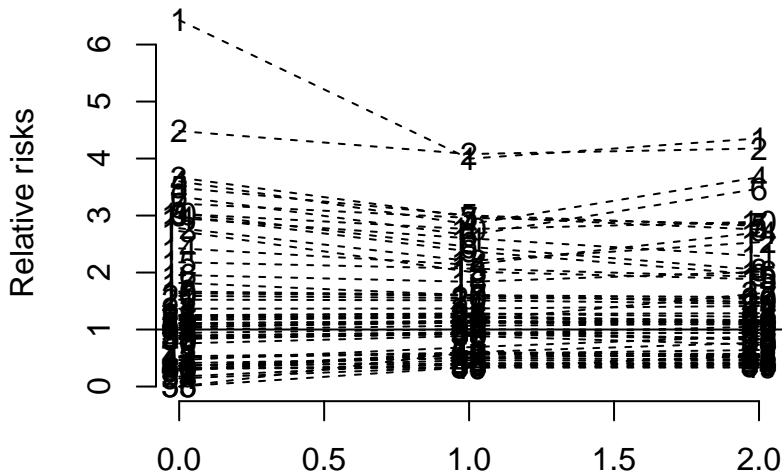


Figure 3: SMRs at 0, EB estimates without covariate at 1 and with EB estimates with covariate AFF at 2.

Comparison of models

We note that the random effects estimates (from the two gamma models) are relatively similar, except in areas 4 and 6 whose estimates are increased, based on the loglinear model.

The fit of this model was shown earlier and we might question the adequacy of this model for large AFF.

Lognormal 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:

$$\begin{aligned} Y_i | \alpha, V_i &\sim_{ind} \text{Poisson}(E_i e^{\alpha} e^{V_i}), \\ V_i | \sigma_v^2 &\sim_{iid} N(0, \sigma_v^2) \end{aligned}$$

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

Note that in INLA the random effect variances are reported in terms of the precisions (the reciprocal of the variance).

Lognormal model

This model gives rise to the posterior distribution;

$$p(\alpha, \tau_v, V_1, \dots, V_n | y) = \frac{\prod_{i=1}^n \Pr(Y_i | \alpha, V_i) p(V_i | \tau_v) p(\alpha) p(\tau_v)}{\Pr(y)}.$$

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

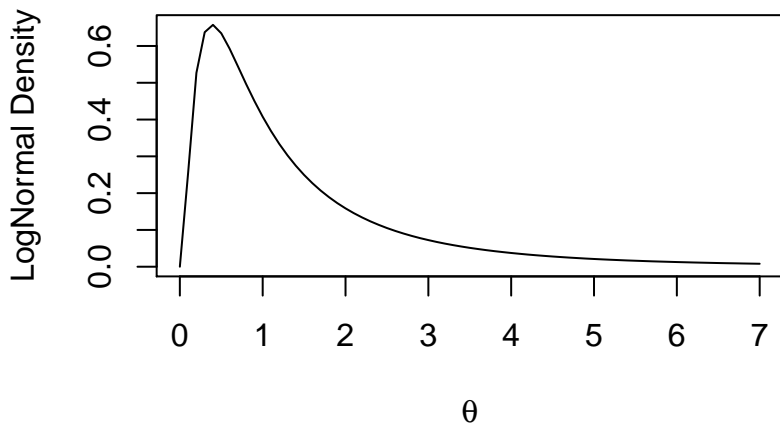
The posteriors on the random effects $p(V_i | y)$ can be extracted, as we will show in subsequent slides.

Prior choice for lognormal model

We specify the 5% and 95% points of the relative risk associated with β as 1 and 5.

```
lnprior <- LogNormalPriorCh(1, 5, 0.5, 0.95)
lnprior
## $mu
## [1] 0
##
## $sigma
## [1] 0.9784688
plot(seq(0, 7, 0.1), dlnorm(seq(0, 7, 0.1),
  meanlog = lnprior$mu, sdlog = lnprior$sigma),
  type = "l", xlab = expression(theta),
  ylab = "LogNormal Density")
```

Prior choice for Lognormal model



Prior choice for lognormal model

The priors $\sigma_v^{-2} \sim \text{Ga}(1, 0.0260)$ or $\sigma_v^{-2} \sim \text{Ga}(0.5, 0.0005)$ will often be suitable in a mapping context.

The 2.5%, 50% (median) and 97.5% quantiles for σ_v are calculated below.

Recall that

$$\text{s.d.}(\text{Relative Risk}) \approx \sigma_v.$$

```
1/sqrt(qgamma(c(0.975, 0.5, 0.025), 0.5, 5e-04))  
## [1] 0.01410848 0.04688400 1.00908784  
1/sqrt(qgamma(c(0.975, 0.5, 0.025), 1, 0.026))  
## [1] 0.08395362 0.19367517 1.01338302
```


INLA for lognormal model

We load `inla` and fit the Poisson-Lognormal model for Scotland.

```
# install.packages('INLA',  
# repos='http://www.math.ntnu.no/inla/R/stable')  
library(INLA)  
data(Scotland)  
# Fit Poisson-lognormal model in INLA:  
scotland.fit1 <- inla(Counts ~ 1 + f(Region,  
  model = "iid", param = c(0.5, 5e-04)),  
  data = Scotland, family = "poisson",  
  E = E)
```

Note the specification of the prior for $\sigma_v^{-2} \sim \text{Ga}(0.5, 0.0005)$.

The default prior for α (the intercept) is close to flat.

INLA for lognormal model

```
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"                   "po"
## [13] "waic"                   "model.random"
## [15] "summary.random"        "marginals.random"
## [17] "size.random"           "summary.linear.predictor"
## [19] "marginals.linear.predictor" "summary.fitted.values"
## [21] "marginals.fitted.values" "size.linear.predictor"
## [23] "summary.hyperpar"      "marginals.hyperpar"
## [25] "internal.summary.hyperpar" "internal.marginals.hyperpar"
## [27] "offset.linear.predictor" "model.spde2.blc"
## [29] "summary.spde2.blc"     "marginals.spde2.blc"
## [31] "size.spde2.blc"       "model.spde3.blc"
## [33] "summary.spde3.blc"    "marginals.spde3.blc"
## [35] "size.spde3.blc"       "logfile"
## [37] "misc"                  "dic"
## [39] "mode"                  "neffp"
## [41] "joint.hyper"           "nhyper"
## [43] "version"               "Q"
## [45] "graph"                 "ok"
## [47] "cpu.used"              "all.hyper"
## [49] ".args"                 "call"
## [51] "model.matrix"
```

INLA for lognormal model

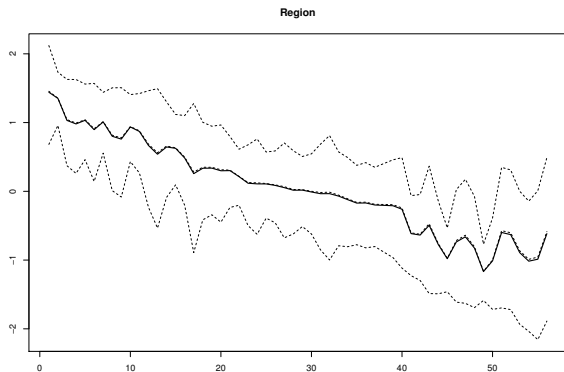
```
summary(scotland.fit1)
##
## Call:
## c("inla(formula = Counts ~ 1 + f(Region, model = \"iid\", param = c(0.5, \"    5e-04)), family = \"p
##
## Time used:
##   Pre-processing      Running inla Post-processing           Total
##         0.7877         0.1237         0.1305         1.0419
##
## Fixed effects:
##              mean      sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) 0.0806 0.1162    -0.1511   0.0816    0.3063 0.0837   0
##
## Random effects:
## Name      Model
## Region    IID model
##
## Model hyperparameters:
##              mean      sd 0.025quant 0.5quant 0.975quant   mode
## Precision for Region 1.782 0.4558     1.045   1.729     2.824 1.63
##
## Expected number of effective parameters(std dev): 43.84(1.621)
## Number of equivalent replicates : 1.277
##
## Marginal log-Likelihood: -187.38
temp <- scotland.fit1$summary.fixed[4]
```

The posterior median for α is 0.082 and for σ_v is $1/\sqrt{1.73} = 0.76$.

Lognormal model: posterior marginals

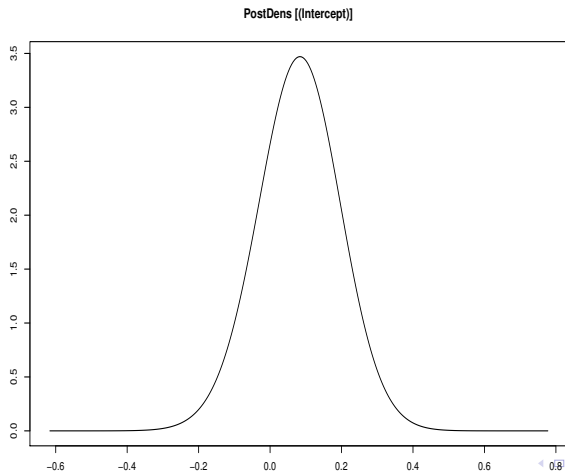
Note that the `inla` plot function and `knitr` don't play nicely so we create a postscript file and then include.

```
plot(scotland.fit1, plot.hyperparameter = FALSE,  
     plot.random.effects = TRUE, plot.fixed.effects = FALSE,  
     prefix = "logmodplot1", postscript = T)
```



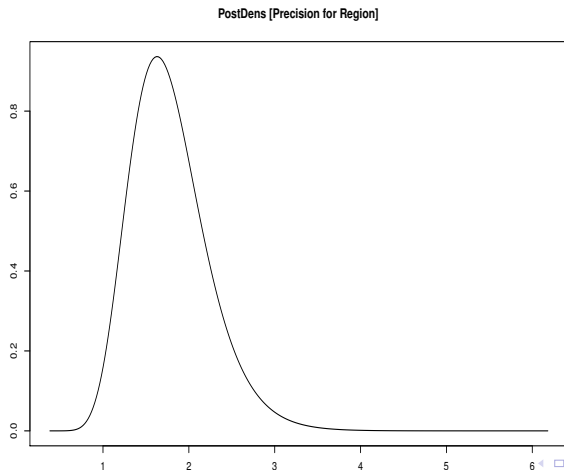
Lognormal model: posterior marginals

```
plot(scotland.fit1, plot.hyperparameter = FALSE, plot.random.effects = FALSE, plot.fixed.effects = TRUE, prefix = "logmodplot2", postscript = T)
```



Lognormal model: posterior marginals

```
plot(scotland.fit1, plot.hyperparameter = TRUE, plot.random.effects = FALSE, plot.fixed.effects = FALSE, prefix = "logmodplot3", postscript = T)
```



Comparison of lognormal and gamma models

```
scotland.fit1$summary.fixed
##              mean              sd 0.025quant    0.5quant 0.975quant
## (Intercept) 0.08059062 0.1161634 -0.1510851 0.08162286 0.3061634
##              mode              kld
## (Intercept) 0.08366219 3.926892e-12
# First illustrate how to extract the intercept
# (alpha)
lnorminter <- scotland.fit1$summary.fixed[4]
# Now extract the medians of the random effects
# (which are centered around alpha)
lnormREs <- exp(scotland.fit1$summary.random$Region[5])
```

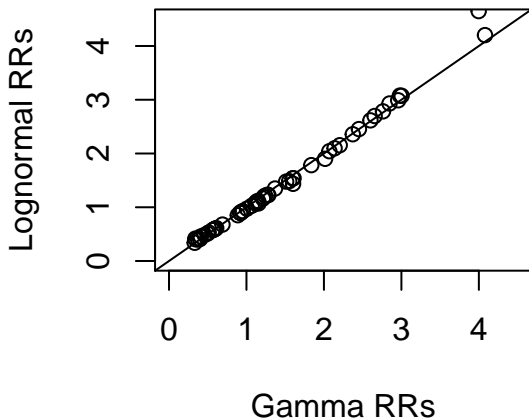
Comparison of lognormal and gamma models

We now compare the gamma and lognormal analyses.

The figure and maps below show the results: the analyses provide very similar estimates, which is reassuring.

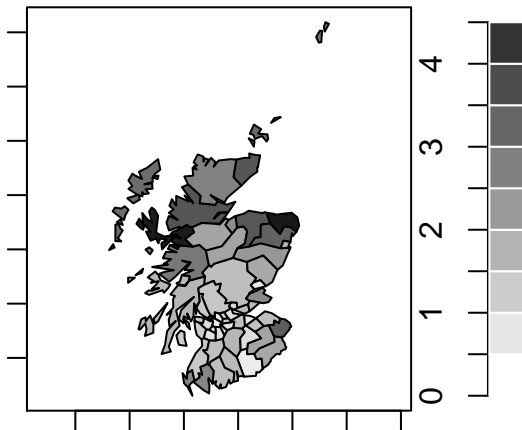
Comparison of lognormal and gamma models

```
lnormRRs <- as.double(exp(lnorminter)) * lnormREs[,  
  1]  
plot(ebresults$RR, lnormRRs, xlim = c(0, 4.5), ylim = c(0,  
  4.5), xlab = "Gamma RRs", ylab = "Lognormal RRs")  
abline(0, 1)
```



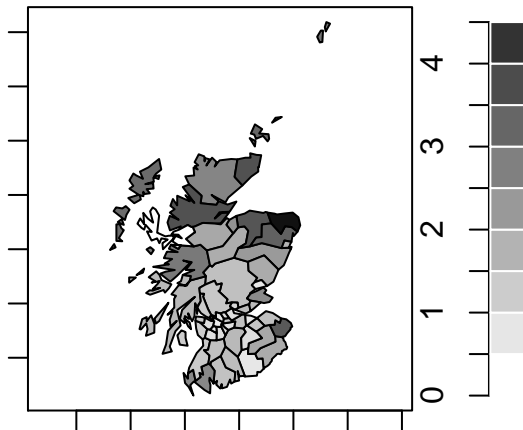
Mapped gamma relative risk estimates

```
par(mar = c(1, 1, 1, 1))  
mapvariable(ebresults$RR, scotland.map, lower = 0,  
            upper = 4.5)
```



Mapped lognormal relative risk estimates

```
par(mar = c(1, 1, 1, 1))  
mapvariable(lnnormRRs, scotland.map, lower = 0, upper = 4.5)
```



Scottish lip cancer

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

$$\log \theta_i = \alpha + x_i\beta + V_i$$

where x_i is the AFF in area i .

Stage 2: The random effects (prior distribution) is

$$V_i|\sigma_v^2 \sim_{iid} N(0, \sigma_v^2).$$

Stage 3: The hyperprior on the hyperparameters $\alpha, \beta, \sigma_v^2$:

$$p(\alpha, \beta, \sigma_v^2) = p(\alpha)p(\beta)p(\sigma_v^2)$$

so that here we have assumed independent priors.

Lognormal non-spatial model with covariates

For ease of implementation reasons Bayesian implementations (including INLA) work with the precision $\tau_v = \sigma_v^{-2}$, rather than the variance σ_v^2 .

We use the default (flattish) priors on α and β and the $\text{Ga}(1, 0.014)$ prior on the precision σ_v^{-2} .

Lognormal non-spatial model with covariates

```
# No spatial effects with covariate
scotland.fit1X <- inla(Counts ~ 1 + I(X) + f(Region,
  model = "iid", param = c(1, 0.014)), data = Scotland,
  family = "poisson", E = E)
summary(scotland.fit1X)
##
## Call:
## c("inla(formula = Counts ~ 1 + I(X) + f(Region, model = \"iid\", param = c(1, \"    0.014))), family
##
## Time used:
##   Pre-processing      Running inla Post-processing          Total
##       0.5476          0.1092          0.0788          0.7356
##
## Fixed effects:
##              mean      sd 0.025quant 0.5quant 0.975quant    mode kld
## (Intercept) -0.4894 0.1560   -0.8003  -0.4881   -0.1858 -0.4855  0
## I(X)         0.0683 0.0139    0.0409  0.0683    0.0958  0.0683  0
##
## Random effects:
## Name      Model
## Region    IID model
##
## Model hyperparameters:
##              mean      sd 0.025quant 0.5quant 0.975quant    mode
## Precision for Region 3.078 0.8927    1.691    2.956    5.168 2.728
##
## Expected number of effective parameters(std dev): 39.22(2.259)
## Number of equivalent replicates : 1.428
##
## Marginal log-Likelihood: -184.65
```

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$summary.fixed[2, ]  
##           mean           sd 0.025quant  0.5quant 0.975quant         mode  
## I(X) 0.06832446 0.01394474 0.04086598 0.0683225  0.0957614 0.06832124  
##           kld  
## I(X) 2.9216e-13  
exp(scotland.fit1X$summary.fixed[2, ])  
##           mean           sd 0.025quant  0.5quant 0.975quant         mode kld  
## I(X) 1.070713 1.014042    1.041712 1.070711    1.100496 1.070709    1
```

Parameter interpretation

```
scotland.fit1X$summary.fixed
##               mean              sd  0.025quant   0.5quant 0.975quant
## (Intercept) -0.48936022 0.15604798 -0.80029637 -0.4880810 -0.1858145
## I(X)         0.06832446 0.01394474  0.04086598  0.0683225  0.0957614
##               mode              kld
## (Intercept) -0.48554319 3.349056e-12
## I(X)         0.06832124 2.921600e-13
```

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

The posterior median for the relative risk associated with a 1 unit increase in X is $\text{median}(\exp(\beta)|y) = \exp(0.068) = 1.07$. This latter calculation exploits the fact that we can transform quantiles¹

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

$$[\exp(0.0409), \exp(0.0958)] = [1.042, 1.101].$$

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.

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

Scottish Lip Cancer: Parameter Interpretation

```
scotland.fit1X$summary.fixed
##              mean              sd 0.025quant  0.5quant 0.975quant
## (Intercept) -0.48936022 0.15604798 -0.80029637 -0.4880810 -0.1858145
## I(X)         0.06832446 0.01394474  0.04086598  0.0683225  0.0957614
##              mode              kld
## (Intercept) -0.48554319 3.349056e-12
## I(X)         0.06832124 2.921600e-13
```

The posterior median of σ_v is $1/\sqrt{2.95} = 0.582$ and a 95% interval is

$$[1/\sqrt{5.13}, 1/\sqrt{1.70}] = [0.44, 0.766].$$

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

Scottish Lip Cancer: Parameter Interpretation

A posterior median of a 95% RRR interval is

$$\begin{aligned} & [\exp(-1.96 \times \text{median}(\sigma_v)), \exp(1.96 \times \text{median}(\sigma_v))] \\ &= [\exp(-1.96 \times 0.582), \exp(1.96 \times 0.582)] = [0.320, 3.13] \end{aligned}$$

which is quite wide.

This also explains the disparity with the prior – large residual relative risks such as this are quite surprising.

Prior influence

To see if our prior was influential we can examine a 95% prior interval for σ_v :

```
1/sqrt(qgamma(c(0.975, 0.025), 1, 0.014))  
## [1] 0.06160512 0.74361984
```

So the posterior interval lies just outside the prior interval which suggests our prior may be having an influence.

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

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

Lognormal spatial model with covariates

We now add spatial (ICAR) random effects to the model.

We need a graph file containing the neighbors.

```
# Spatial effects with covariate
Scotland$Region2 <- Scotland$Region
scotland.fit2 <- inla(Counts ~ 1 + I(X) + f(Region,
  model = "iid", param = c(1, 0.014)) + f(Region2,
  model = "besag", graph = "examples/scotland.graph",
  param = c(1, 0.68)), data = Scotland, family = "poisson",
  E = E, control.predictor = list(compute = TRUE))
```

Lognormal spatial model with covariates

```
summary(scotland.fit2)
##
## Call:
## c("inla(formula = Counts ~ 1 + I(X) + f(Region, model = \"iid\", param = c(1, \"\", \" 0.014)) + f(Reg:
##
## Time used:
##   Pre-processing      Running inla Post-processing          Total
##         0.8672         0.2084         0.0963         1.1718
##
## Fixed effects:
##              mean      sd 0.025quant 0.5quant 0.975quant      mode kld
## (Intercept) -0.1207 0.1151   -0.3483  -0.1202    0.1042 -0.1193  0
## I(X)         0.0263 0.0119    0.0024   0.0264    0.0494  0.0267  0
##
## Random effects:
## Name      Model
## Region    IID model
## Region2    Besags ICAR model
##
## Model hyperparameters:
##              mean      sd 0.025quant 0.5quant 0.975quant
## Precision for Region 114.630 81.1623    22.896   94.610   324.483
## Precision for Region2  2.103  0.6416     1.114    2.014    3.612
##              mode
## Precision for Region  59.169
## Precision for Region2  1.847
##
## Expected number of effective parameters(std dev): 29.49(2.922)
## Number of equivalent replicates : 1.899
##
## Marginal log-Likelihood: -198.37
## Posterior marginals for linear predictor and fitted values computed
```

Lognormal spatial model with covariates

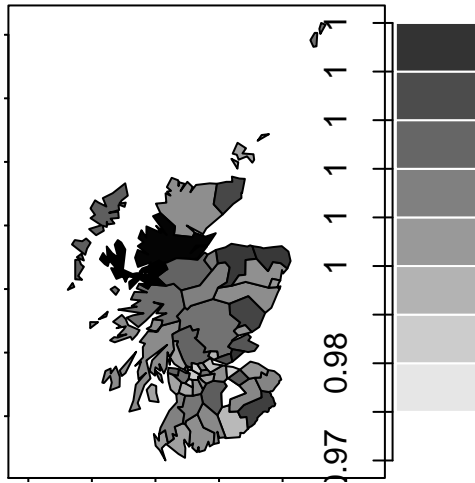
Now we provide maps of the non-spatial and spatial random effects.

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

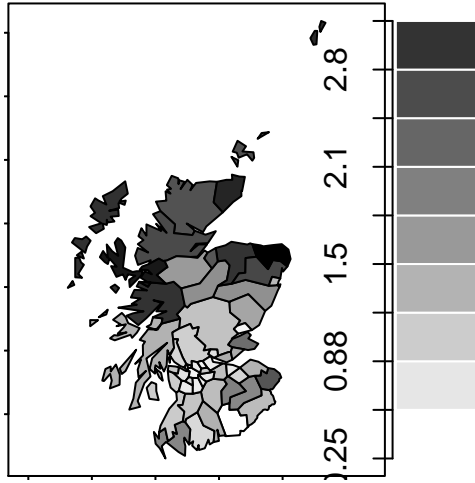
Note the differences in the scales: the spatial random effects dominate here.

```
REsnonspat <- exp(scotland.fit2$summary.random$Region[5])  
REsspat <- exp(scotland.fit2$summary.random$Region2[5])  
par(mfrow = c(1, 1), mar = c(0.1, 0.1, 0.1, 0.1))  
mapvariable(REsnonspat[, 1], scotland.map)
```

Lognormal spatial model with covariates: non-spatial random effects

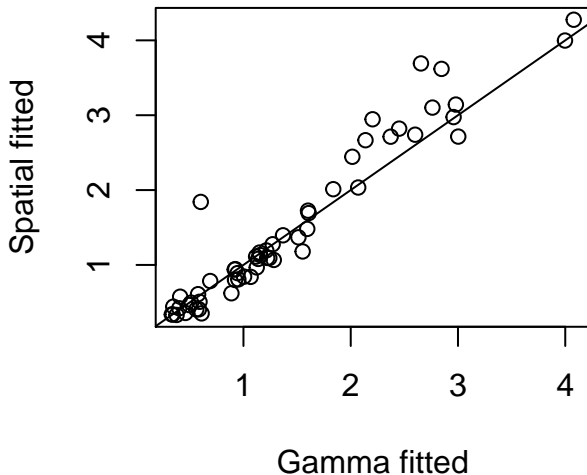


Lognormal spatial model with covariates: spatial random effects



Comparison of spatial lognormal and gamma fits: some differences

```
plot(ebresults$RR, scotland.fit2$summary.fitted.values[,  
  4], xlab = "Gamma fitted", ylab = "Spatial fitted")  
abline(0, 1)
```



INLA representation

Recall the spatial model is $Y_i|\theta_i \sim \text{Poisson}(E_i\theta_i)$ with

$$\log \theta_i = \alpha + \beta x_i + V_i + U_i.$$

The relationship between the statistical model and the `inla` specification is:

$$\begin{aligned} \log \theta_i &= \underbrace{\alpha}_1 + \underbrace{x_i \beta}_{\text{I(X)}} \\ &+ \underbrace{V_i}_{\text{f(Region,model="iid")}} \\ &+ \underbrace{U_i}_{\text{f(Region2,model="besag",graph="scotland.graph")}} \end{aligned}$$

Spatial model: confounding by location

The command `plot(scotland.fit2)` provides plots of: marginal posterior distributions of α , β , σ_v^{-2} , σ_u^{-2} and summaries of the random effects V_i , U_i and the linear predictors and fitted values, all by area.

Note that the posterior mean estimate of β associated with AFF goes from $0.068 \rightarrow 0.026$ 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.

Scotland

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

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 \hat{U}_i .

A more complete analysis would address the sensitivity to the prior specifications on σ_v and σ_u .

Creating an INLA Graph File from a Shapefile

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(rgdal)  # for readOGR
library(spdep)  # for poly2nb and nb2inla
countymap = readOGR(dsn = "examples/wacounty.shp",
                    layer = "wacounty")
## OGR data source with driver: ESRI Shapefile
## Source: "examples/wacounty.shp", layer: "wacounty"
## with 39 features
## It has 6 fields
nb.map <- poly2nb(countymap)
nb2INLA("wacounty.graph", nb.map)
```

RW1 and RW2 models

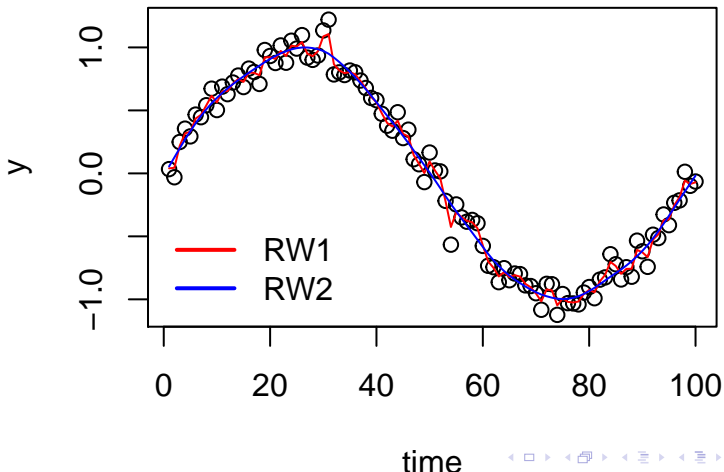
Random walk models are a common Bayesian tool for smoothing in time.

To illustrate their use, we simulate data from a sine curve and then fit using random walk models of order 1 (RW1) and order 2 (RW2).

```
n <- 100
time <- seq(1, n)
y <- sin((time/n) * 2 * pi) + rnorm(n, mean = 0, sd = 0.1)
data <- data.frame(y = y, time = time)
formula1 = y ~ f(time, model = "rw1")
fit1 <- inla(formula1, data = data, family = "gaussian",
             control.predictor = list(compute = TRUE))
formula2 = y ~ f(time, model = "rw2")
fit2 <- inla(formula2, data = data, family = "gaussian",
             control.predictor = list(compute = TRUE))
```

Fitted curves: RW2 fit is smoother than RW1

```
par(mfrow = c(1, 1))  
plot(y ~ time)  
lines(time, fit1$summary.fitted.values[, 1], col = "red")  
lines(time, fit2$summary.fitted.values[, 1], col = "blue")  
legend("bottomleft", bty = "n", legend = c("RW1", "RW2"),  
      lwd = 2, col = c("red", "blue"))
```



Space-time modeling of Ohio data

We examine data on lung cancer deaths in 88 counties of Ohio over the years 1968–1988.

We adjust for gender and race and for age, the latter via 5 age bands: 0–44, 45–54, 55–64, 65–74, 75+.

These data have been analyzed by a number of authors including Waller et al (1997), Xia and Carlin (1998), Knorrheld and Besag (1998).

The figure gives the Ohio map with county numbers. We will focus particular attention on the 6 counties in the North West, numbers 86, 26, 48, 20, 35 and 87.

Space-Time Modeling of Ohio Data

The following code produces the figure below

It uses the `spacetime` package (Pebesma 2012) and various others.

A `spacetime` object needs creating; in particular, from the help file:

```
stConstruct(x,space,time,SpatialObj = NULL,TimeObj =  
NULL,      crs =  
CRS(as.character(NA)),interval,endTime)
```

Space-Time Modeling of Ohio Data

```
library(lattice)
library(spacetime)
library(xts)
library(RColorBrewer)
library(raster)
library(maps)
library(maptools)
library(plm)
```

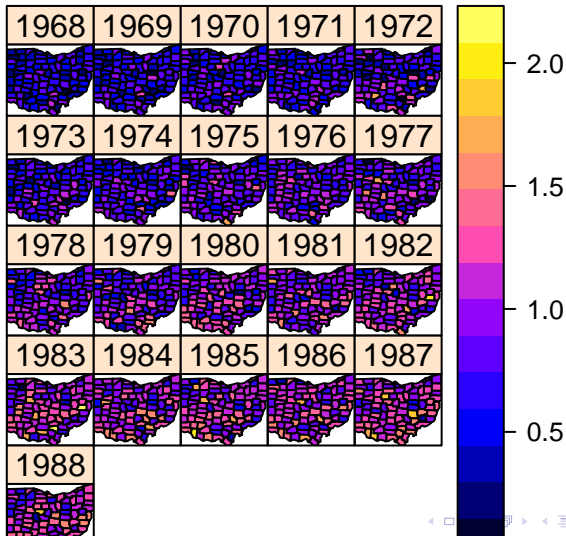
Space-Time Modeling of Ohio Data

```
ohiomap <- map("county", "ohio", fill = TRUE,
              plot = FALSE)
IDs <- sapply(strsplit(ohiomap$names, "[,:]"),
              function(x) x[2])
counties <- map2SpatialPolygons(ohiomap,
                                IDs = IDs)
tmp <- read.table("http://faculty.washington.edu/jonno/SISMIDmat",
                 header = T)
tmp$SMR <- tmp$Y/tmp$E
tmp$counties <- rep(IDs, 21)
yrs <- 1968:1988
tmp$time = rep(as.Date(paste(yrs, "01", "01",
                             sep = "-))), each = 88)
#
x = stConstruct(tmp, "counties", "time",
               counties)
stplot(x[, , "SMR"])
```

Maps over time

```
stplot(x[, , "SMR"])
```

SMR



Space-time modeling of Ohio data

Questions of interest:

- ▶ What do we learn from examining maps at different times?
- ▶ Is there spatial pattern?
- ▶ Is the spatial pattern similar at different times?

From the maps we see that the risk is increasing over years but not clear whether there is a change in the clustering.

We could evaluate a summary measure such as Moran's statistic for each year, or fit separate models to each year's data and examine summary measures of the random effects model (e.g. σ_v and σ_u).

Ohio temporal trends

We fit a Poisson log-linear in time to see the range of intercepts and slopes:

$$Y_{it} | \alpha_i, \beta_i \sim \text{Poisson} [E_{it} \exp(\alpha_i + \beta_i t)]$$

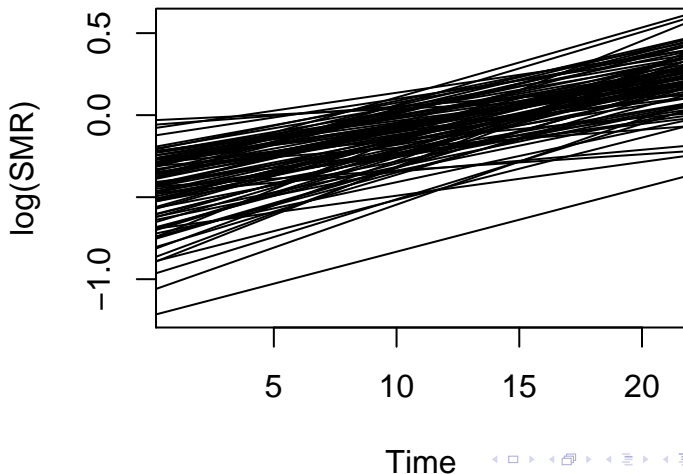
This is a very quick examination and doesn't tell us whether these models fit or not (we could examine residuals to start to answer this).

Linear trends in each Ohio county

```
n.space <- 88
n.time <- 21
# Examine linear associations
inter <- slope <- NULL
for (i in 1:n.space) {
  ind <- which(tmp$county == i, )
  data.temp <- tmp[ind, ]
  mod <- glm(Y ~ offset(log(E)) + seq(1, n.time),
    family = "poisson", data = data.temp)
  inter[i] <- mod$coeff[1]
  slope[i] <- mod$coeff[2]
}
log.SMR <- ifelse(tmp$Y == 0, log((tmp$Y + 0.5)/tmp$E),
  log(tmp$Y/tmp$E))
log.SMR <- matrix(log.SMR, nrow = 88, ncol = 21)
```


Linear trends in each Ohio county

```
plot(x = seq(1, n.time), y = log.SMR[1, ], type = "n",  
     xlab = "Time", ylab = "log(SMR)", ylim = c(min(inter),  
         max(inter + 21 * slope)))  
for (i in 1:n.space) {  
  abline(inter[i], slope[i])  
}
```



Ohio temporal trends

What we learn from examining time patterns in different areas:

- ▶ What is the general trend?
- ▶ Is the trend similar across different areas, i.e.~is there space-time interaction?

It would be nice to have a plot in which trends in close by areas can be compared – if more similar then it suggests an interaction model.

Ohio temporal trends

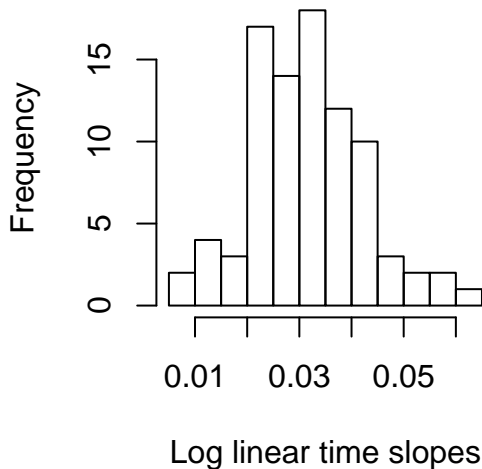
We plot the fitted lines $\hat{\alpha}_i + \hat{\beta}_i t$ for each county below.

The histogram of the slope estimates $\exp(\hat{\beta}_i)$ is also given below; all the slopes are greater than zero and no outliers.

We map the slopes to examine whether the sizes tend to cluster — this informs on the level of space-time interaction — no indication of a pattern (though hard to tell for sure).

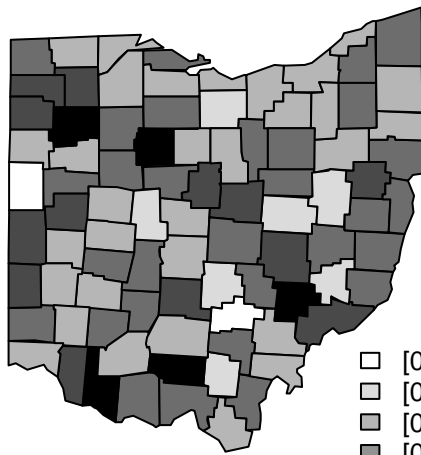
Ohio temporal trends

```
hist(slope, xlab = "Log linear time slopes", main = "")
```



Ohio temporal trends, by county

```
source("http://faculty.washington.edu/jonno/SISMIDmaterial/PolyMap.R")  
OhioMap(slope, ncol = 8, type = "e", figmain = "")
```

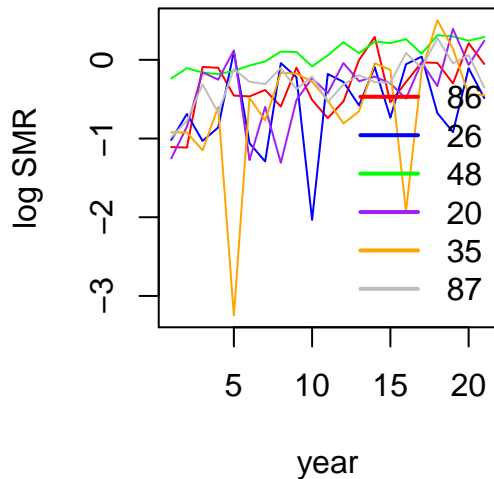


Ohio temporal trends

We now examine more closely the 6 areas in the North West of Ohio (for illustration).

```
nplot <- 6
ind <- c(86, 26, 48, 20, 35, 87)
maxlSMR <- max(log.SMR[ind, ])
minlSMR <- min(log.SMR[ind, ])
plot(1:n.time, log.SMR[ind[1], ], type = "n", ylab = "log SMR",
     xlab = "year", ylim = c(minlSMR, maxlSMR))
colp <- c("red", "blue", "green", "purple", "orange",
          "grey")
for (i in 1:nplot) {
  lines(1:n.time, log.SMR[ind[i], ], col = colp[i])
  sel <- which(tmp$county == ind[i], )
  data.temp <- tmp[sel, ]
}
legend("bottomright", legend = ind, col = colp, lwd = 2,
      lty = 1, bty = "n")
```

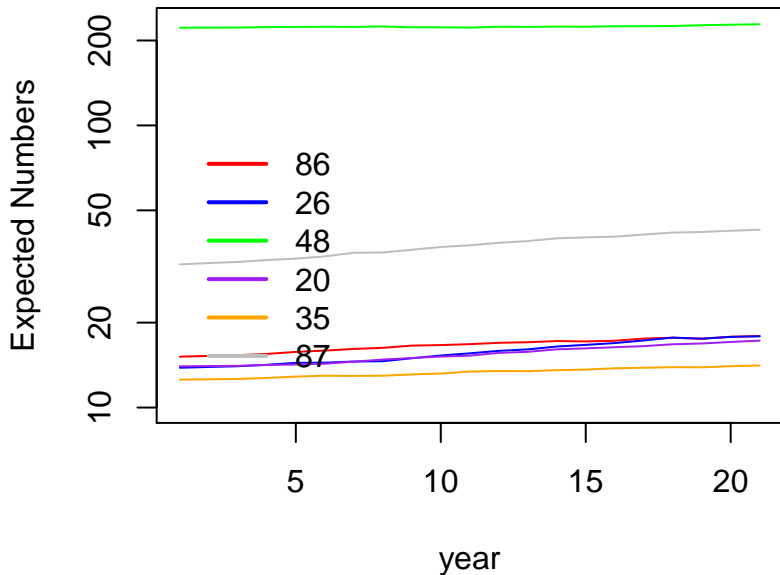
Ohio temporal trends: log SMRs



Space-time modeling of Ohio data

```
plot(1:n.time, log.SMR[1, ], type = "n", ylab = "Expected Number  
  xlab = "year", ylim = c(10, 230), log = "y")  
colp <- c("red", "blue", "green", "purple", "orange",  
  "grey")  
for (i in 1:nplot) {  
  sel <- which(tmp$county == ind[i], )  
  data.temp <- tmp[sel, ]  
  lines(1:n.time, data.temp$E, col = colp[i])  
}  
legend(1, 100, legend = ind, col = colp, lwd = 2, lty = 1,  
  bty = "n")
```


Space-time modeling of Ohio data: Expected numbers



Space-time modeling of Ohio data

We fit the model

$$\log \mu_{it} = \beta_0 + \beta_1 t + V_i + U_i + \gamma_t$$

where

- ▶ $V_i \sim_{iid} N(0, \sigma_v^2)$ are spatially unstructured,
- ▶ U_i are spatially structured ICAR,
- ▶ $\gamma_t \sim_{iid} N(0, \sigma_\gamma^2)$ are temporally unstructured.
- ▶ $\exp(\beta_1)$ is the change in the relative risk associated with a 1-year increase in time.

Space-time modeling of Ohio data

```
data.inla <- read.table("http://faculty.washington.edu/jonno/SIS
  header = T)
head(data.inla)
##      county time          E  Y
## 1         1    1 10.27767   6
## 2         2    1 47.13125  32
## 3         3    1 19.13895  15
## 4         4    1 43.81020  27
## 5         5    1 20.29211  12
## 6         6    1 17.16256   7
data.inla$region.struct <- data.inla$region.unstruct <- data.inla$region
data.inla$time.struct <- data.inla$time.unstruct <- data.inla$time
```

Space-time modeling of Ohio data

Set up inla main effects model with a temporal log-linear trend.

```
formula <- Y ~ time.struct + f(region.struct, model = "besag",  
  graph = "examples/ohio.graph", param = c(1, 0.026)) +  
  f(region.unstruct, model = "iid", param = c(1,  
    0.026)) + f(time.unstruct, model = "iid", param = c(1,  
    0.026))  
mod.t = inla(formula, data = data.inla, family = "poisson",  
  E = E, control.predictor = list(compute = TRUE))
```

Space-Time Modeling of Ohio Data

```
summary(mod.t)
##
## Call:
## c("inla(formula = formula, family = \"poisson\", data = data.inla, ", "      E = E, control.predictor =
##
## Time used:
##   Pre-processing      Running inla Post-processing      Total
##         1.2270          5.5368          0.2286          6.9924
##
## Fixed effects:
##              mean      sd 0.025quant 0.5quant 0.975quant      mode kld
## (Intercept) -0.4116 0.0312   -0.4734  -0.4116   -0.3502 -0.4115   0
## time.struct  0.0276 0.0023    0.0231   0.0276    0.0321  0.0276   0
##
## Random effects:
## Name      Model
## region.struct  Besags ICAR model
## region.unstruct  IID model
## time.unstruct  IID model
##
## Model hyperparameters:
##              mean      sd 0.025quant 0.5quant 0.975quant
## Precision for region.struct    52.30 25.95    19.06   46.70   118.30
## Precision for region.unstruct   84.43 23.89    46.71   81.37   139.93
## Precision for time.unstruct    287.05 90.13   144.06  276.30   494.22
##              mode
## Precision for region.struct    37.46
## Precision for region.unstruct   75.65
## Precision for time.unstruct   256.03
##
## Expected number of effective parameters(std dev): 97.88(1.548)
## Number of equivalent replicates : 18.88
##
## Marginal log-Likelihood: -5969.99
## Posterior marginals for linear predictor and fitted values computed
```

Space-time modeling of Ohio data

The posterior median (95% interval) is 1.028 (1.023,1.033). The mean of the exponentiated slopes calculated earlier was 1.031.

The various precisions (reciprocal variances) are difficult to interpret since the unstructured components are marginal (σ_v^{-2} , σ_γ^{-2}) while the structured component (σ_u^{-2}) is conditional.

Space-Time Modeling of Ohio Data

On the next slide we give summaries of the precisions: the posterior median of σ_ϵ (spatial) is $1/\sqrt{81.19} = 0.11$ and for σ_γ (temporal) is $1/\sqrt{276.59} = 0.06$.

Hence, the spatial unstructured makes around twice the contribution to the temporal unstructured. But over the 21 years the linear temporal trend (on the log relative risk scale) increases $21 \times 0.032 = 0.67$ and so is quite large.

Space-Time Modeling of Ohio Data

Extract the spatial and non-spatial random effects

```
gamma.t <- mod.t$summary.random$time.unstruct$mean
S.t <- mod.t$summary.random$region.struct$mean
epsilon.t <- mod.t$summary.random$region.unstruct$mean
range(S.t)
## [1] -0.1975292  0.1867911
range(epsilon.t)
## [1] -0.3076826  0.2004370
```


Mapping spatial and non-spatial random effects

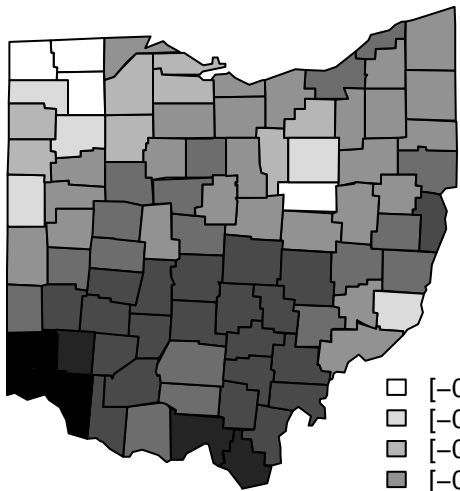
From main effects model plots of

- ▶ maps of spatial and non-spatial
- ▶ smooth term in time

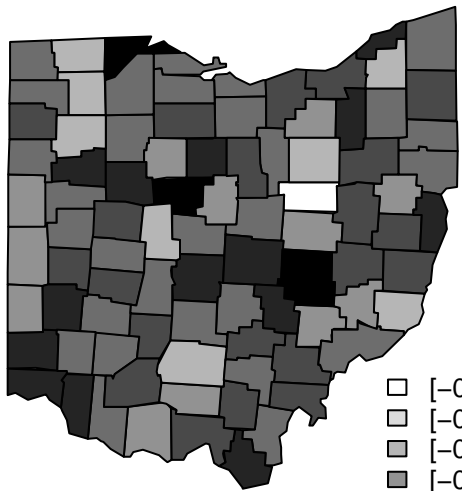
```
OhioMap(S.t, ncol = 8, type = "e", figmain = "Structured sp  
OhioMap(epsilon.t, ncol = 8, type = "e", figmain = "Unstruc  
plot(1:n.time, gamma.t, type = "b", lwd = 2, xlab = "year",  
      ylab = "gamma", main = "Unstructured temporal term")  
abline(h = 0, col = "red", lwd = 2)
```

Space-time modeling of Ohio data

Structured spatial component

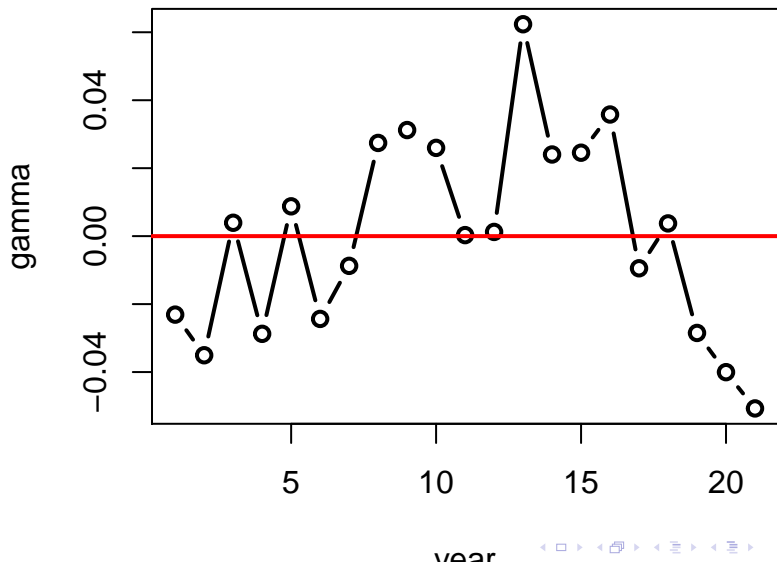


Unstructured spatial component



Space-time modeling of Ohio data

Unstructured temporal term



Space-time modeling of Ohio data

The latter plot shows that the “unstructured” random effects have some temporal structure, which suggests that the linear term is not capturing the temporal trend well.

Space-Time Modeling of Ohio Data

We now examine a couple of the Knorr-Held (2000) interaction models.

Recall the Type I interaction model with

$$\mu_{it} = \beta_0 + \epsilon_i + S_i + \gamma_t + \eta_t + \delta_{it}$$

and independent interaction parameters in time and space, i.e. $\delta_{ij} \sim_{iid} N(0, \sigma_\delta^2)$, or

$$f(\delta | \sigma_\delta^2) \propto \exp \left(-\frac{1}{2\sigma_\delta^2} \sum_{i=1}^I \sum_{t=1}^T \delta_{it}^2 \right).$$

Very flexible, but lack of structure is a drawback.

To specify this model in `inla` is not straightforward as we need to construct the δ specification from first principles.

Space-Time Modeling of Ohio Data

[illegible]

Space-time model

```
# --- add all the necessary columns to the data
# ---#
k <- 88
ntime <- 21
ohiosp <- data.frame(fips = data.inla$county, county = data.inla
  year = data.inla$time, expected = data.inla$E,
  deaths = data.inla$Y)
ohiosp$time.unstruct <- ohiosp$time.struct <- ohiosp$year
ohiosp$region.unstruct <- ohiosp$region.struct <- ohiosp$county
mu0 <- log(0.1/0.9)
sig2.0 <- 10000
# Prior parameters for the gamma on the precisions.
a <- 1
b <- 0.01
```


Space-time model

```
# --- Main Effects (no space-time interaction), ~7
# seconds ---#
formula <- deaths ~ 1 + f(time.unstruct, model = "iid",
  param = c(a, b)) + f(time.struct, model = "rw1",
  param = c(a, b), constr = T) + f(region.unstruct,
  model = "iid", param = c(a, b)) + f(region.struct,
  model = "besag", param = c(a, b), graph = k.theta)
#
mod <- inla(formula, data = ohiosp, family = "poisson",
  E = expected, control.fixed = list(mean.intercept = c(mu0),
  prec.intercept = c(1/sig2.0)), control.predictor = list(
```

Model summary: main effects

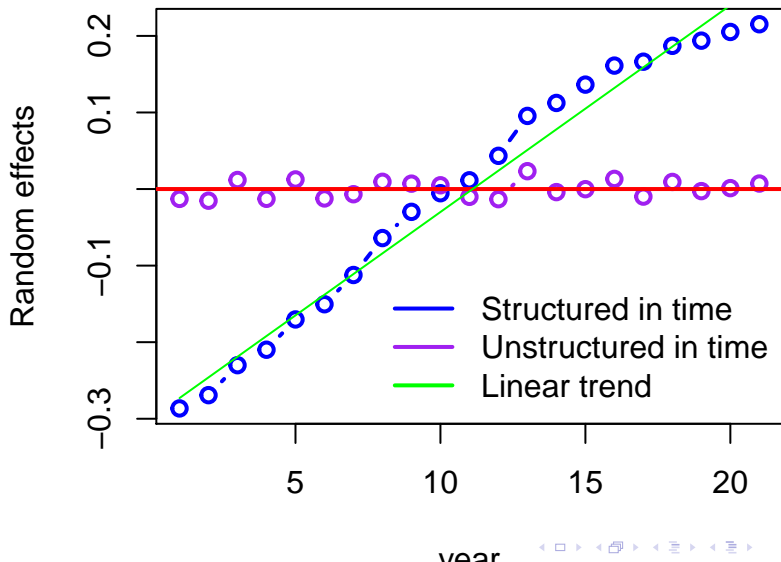
```
tau.0 <- mod$summary.random$time.struct$mean  
gamma.0 <- mod$summary.random$time.unstruct$mean  
S.0 <- mod$summary.random$region.struct$mean  
epsilon.0 <- mod$summary.random$region.unstruct$mean
```

Main effects plot

```
plot(1:ntime, tau.0, type = "b", lwd = 2, xlab = "year",  
     ylab = "Random effects", main = "Smooth temporal term: Main  
     col = "blue")  
lines(1:ntime, gamma.0, type = "b", lwd = 2, col = "purple")  
abline(h = 0, col = "red", lwd = 2)  
lines(1:ntime, -0.3 + 0.027 * (1:ntime), col = "green")  
legend("bottomright", legend = c("Structured in time",  
    "Unstructured in time", "Linear trend"), col = c("blue",  
    "purple", "green"), lwd = 2, bty = "n")
```

Main effects plot

Smooth temporal term: Main effects



Model summary

The command `summary(mod.1)` gives the precisions for the unstructured space, time and interaction terms as 81.59, 221.34 and 787.96 showing that the interactions terms are very small.

Model summary

Now the type I interaction model

```
# - Type I -#
ohiosp$spacetime.unstruct <- 1:1848
#
formulaI <- deaths ~ 1 + f(time.unstruct, model = "iid",
  param = c(a, b)) + f(time.struct, model = "rw1",
  param = c(a, b), constr = T) + f(region.unstruct,
  model = "iid", param = c(a, b)) + f(region.struct,
  model = "besag", param = c(a, b), graph = k.theta) +
  f(spacetime.unstruct, model = "iid", hyper = list(prec = list(
    b))))

modI <- inla(formulaI, data = ohiosp, family = "poisson",
  E = expected, , control.fixed = list(mean.intercept = c(mu0)
  prec.intercept = c(1/sig2.0)), control.predictor = list(
```

Model summary

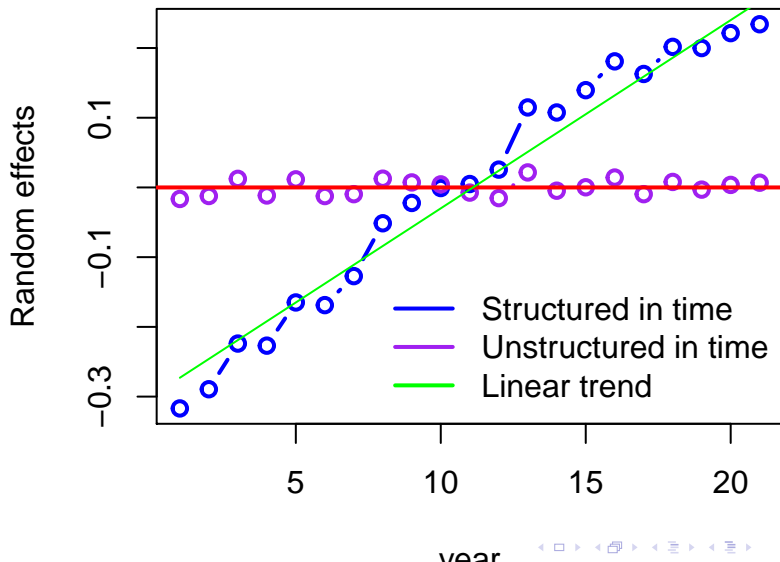
Code to extract summaries to produce Figure below

```
# random effect posterior mean estimates
tau.1 <- modI$summary.random$time.struct$mean
gamma.1 <- modI$summary.random$time.unstruct$mean
S.1 <- modI$summary.random$region.struct$mean
epsilon.1 <- modI$summary.random$region.unstruct$mean
delta.1 <- matrix(modI$summary.random$spacetime.unstruct$mean,
  byrow = F, ncol = ntime)
```

Model summary

```
plot(1:ntime, tau.1 + gamma.1, type = "b", lwd = 2,  
     xlab = "year", ylab = "Random effects", main = "Smooth tempo  
     col = "blue")  
lines(1:ntime, gamma.1, type = "b", lwd = 2, col = "purple")  
abline(h = 0, col = "red", lwd = 2)  
lines(1:ntime, -0.3 + 0.027 * (1:ntime), col = "green")  
legend("bottomright", legend = c("Structured in time",  
    "Unstructured in time", "Linear trend"), col = c("blue",  
    "purple", "green"), lwd = 2, bty = "n")
```


Smooth temporal terms: Type I

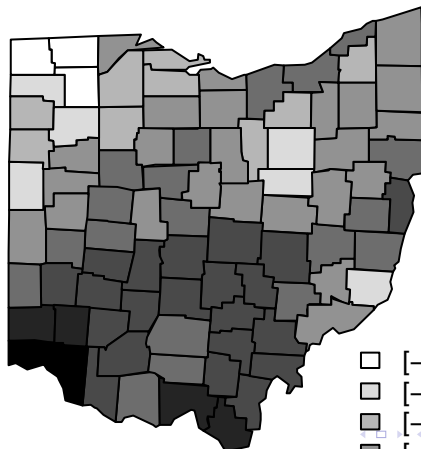


Interaction summaries

```
# plot of the spatial component
```

```
OhioMap(S.1, ncol = 8, type = "e", figmain = "Structured sp
```

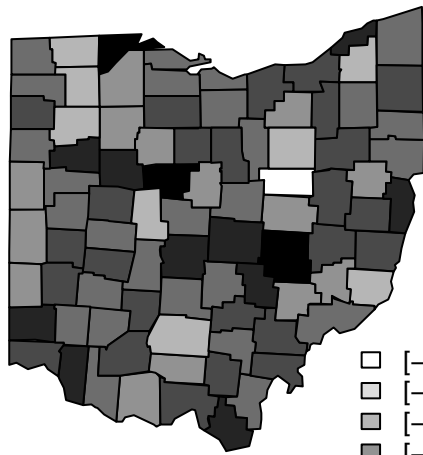
Structured spatial component



Interaction summaries

```
OhioMap(epsilon.1, ncol = 8, type = "e", figmain = "Unstruc
```

Unstructured spatial component

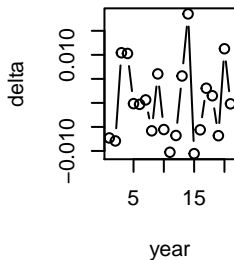


Interaction summaries for 6 areas

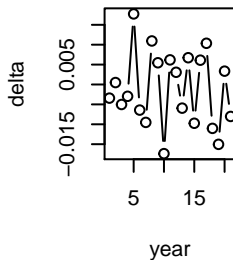
```
# plot of the NW delta's (the interaction terms)
par(mfrow = c(2, 3))
plot(1:ntime, delta.1[86, ], type = "b", ylab = "delta",
     xlab = "year", main = "county 86")
plot(1:ntime, delta.1[26, ], type = "b", ylab = "delta",
     xlab = "year", main = "county 26")
plot(1:ntime, delta.1[48, ], type = "b", ylab = "delta",
     xlab = "year", main = "county 48")
plot(1:ntime, delta.1[20, ], type = "b", ylab = "delta",
     xlab = "year", main = "county 20")
plot(1:ntime, delta.1[35, ], type = "b", ylab = "delta",
     xlab = "year", main = "county 35")
plot(1:ntime, delta.1[87, ], type = "b", ylab = "delta",
     xlab = "year", main = "county 87")
```

Interaction summaries for 6 areas

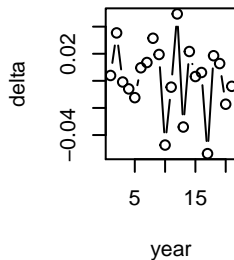
county 86



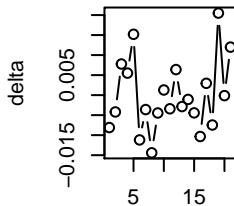
county 26



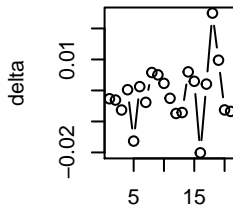
county 48



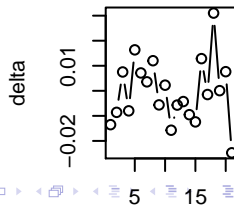
county 20



county 35



county 87



Examination of Space-Time Interaction

To see if a more complex interaction model is warranted we evaluate the autocorrelation function for the δ_{it} , with one each i (i.e.~each area).

We then map the auocorrelations at lag 1.

The rationale for this is that:

- ▶ We would expect to see positive correlations if there is temporal structure in the residuals (Type II interaction).

-If there were no evidence of temporal structure but a spatial pattern in the autocorrelations then we would see clustering in the map (Type III interaction).

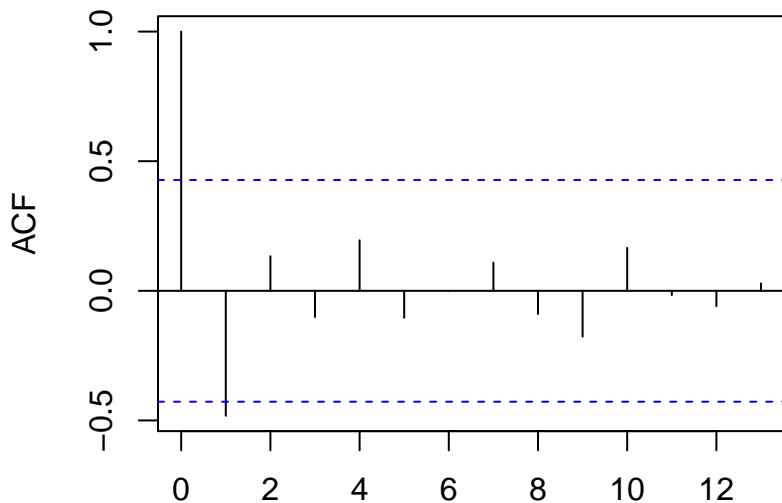
- ▶ If there were positive correlations and spatial clustering then Type IV is appropriate.

Space time interactions

```
acfobj <- rep(0, 88)
for (i in 1:88) {
  temp <- acf(delta.1[i, ])
  acfobj[i] <- temp$acf[2, , 1]
}
OhioMap(acfobj, ncol = 4, type = "e", figmain = "ACF",
  lower = -0.6, upper = 0.6)
```

Space time interactions

Series delta.1[i,]



Space time modeling

To follow up we examine the δ 's in the North West more closely.

With respect to the interaction figure.

There appears to be no/little temporal structure in each of the panels — if there were, Type II interaction would be suggested.

There appears to be no/little spatial structure (eg. plot (1,1) does not appear similar to plot (1,2) or plot (2,1)) in each of the panels — if there were, Type III interaction would be suggested.

If there were temporal and spatial structure a Type IV interaction model be suggested.

The δ 's are small here which suggests no strong interactions.

Conclusion for Ohio: Spatial effects and temporal effects which are not linear, but little interaction.