Implementing the models in WinBUGS

The dataset we will use contains the following information:

Table: Summary of pollutants measured, and periods of operation, at eight sites in London, 1994–97. The total number of days of operation are given for each pollutant at each site together with the percentage of missing observations. The units are μgm^{-3} for PM$_{10}$, parts per billion for SO$_2$ and NO and parts per million for CO.

	Period	Total	Missing	%	Mean	Min.	25%	Med.	75%	Max.
Bexley										
PM$_{10}$	1994-97	1461	211	14.4	24.0	4.0	15.0	20.0	29.0	92.0
SO$_2$	1994, 1996-97	1095	178	16.3	6.9	1.0	3.0	4.0	8.0	76.0
NO	-	-	-	-	-	-	-	-	-	-
CO	1994-97	1461	192	13.1	0.5	0.1	0.3	0.4	0.5	4.4
Bloomsbury										
PM$_{10}$	1994-97	1461	61	4.2	28.0	7.0	19.0	24.0	34.0	103.0
SO$_2$	1994-97	1461	115	7.9	8.3	1.0	4.0	6.0	11.0	48.0
NO	1994-97	1461	44	3.0	42.4	4.0	19.0	30.0	50.0	467.0
CO	1994-97	1461	68	4.7	0.7	0.1	0.4	0.6	0.8	4.3
Brent										
PM$_{10}$	1996-97	731	120	16.4	20.8	6.0	14.0	18.0	25.0	82.0
SO$_2$	1996-97	731	33	4.5	4.4	1.0	2.0	3.0	5.2	20.0
NO	1996-97	731	57	7.8	23.8	1.0	5.0	8.0	22.5	414.0
CO	1996-97	366	15	4.1	0.5	0.1	0.2	0.3	0.7	5.0
Eltham										
PM$_{10}$	1996-97	731	166	22.7	21.2	8.0	15.0	18.0	25.0	81.0
SO$_2$	1996-97	731	91	12.4	4.6	1.0	2.0	3.0	5.0	40.0
NO	1996-97	731	95	13.0	21.7	1.0	5.0	9.0	20.0	339.0
CO	-	-	-	-	-	-	-	-	-	-

	Period	Total	Missing	%	Mean	Min.	25%	Med.	75%	Max.
Harringey										
PM_{10}	1996-97	731	161	22.0	26.2	8.0	18.0	22.0	32.0	89.0
SO_2	-	-	-	-	-	-	-	-	-	-
NO	1996-97	731	139	19.0	63.3	5.0	28.0	43.0	68.6	562.0
CO	-	-	-	-	-	-	-	-	-	-
Hillingdon										
PM_{10}	1996-97	731	225	30.8	24.5	6.0	16.0	21.0	31.0	88.0
SO_2	1996-97	731	230	31.5	5.1	1.0	3.0	4.0	6.0	28.0
NO	1996-97	731	252	34.5	81.9	2.0	31.0	67.0	105.0	506.0
CO	1996-97	731	268	36.7	0.8	0.2	0.5	0.6	0.9	4.3
N. Kensington										
PM_{10}	1996-97	731	99	13.5	23.6	9.0	16.0	20.0	27.2	89.0
SO_2	1996-97	731	91	12.4	4.6	1.0	2.0	3.0	6.0	32.0
NO	1996-97	731	106	14.5	27.6	1.0	6.0	11.0	25.0	442.0
CO	1996-97	731	93	12.7	1.2	0.1	0.4	0.7	1.3	16.6
Sutton										
PM_{10}	1996-97	731	92	12.6	25.1	9.0	17.0	22.0	29.0	250.0
SO_2	1996-97	731	96	13.1	4.9	1.0	2.7	4.0	6.0	28.4
NO	1996-97	731	106	14.5	51.1	3.0	26.3	39.0	57.0	404.0
CO	1996-97	731	104	14.2	1.1	0.2	0.8	1.0	1.3	6.7

Single pollutant, single monitoring site

- **Stage One, Observed Data Model:**

  $$Y_t = X_t^T \beta_1 + \theta_t + v_t,$$

  v_t is referred to as *measurement error*, and assumed to be are independent and identically distributed (i.i.d.) as $N(0, \sigma_v^2)$
- In WinBUGS (ignoring the covariates for simplicity)

```
model {
       for (t in 2:(n-1)) {
# observation model
              y[t] ~ dnorm(theta[t],tau.v)
.
}  # t loop
y[1]~dnorm(theta[1],tau.v)
y[n]~dnorm(theta[n],tau.v)
.
   tau.v ~ dgamma(1,0.01)
}  # end of model
```

- **Stage Two, Temporal Model:**

$$\theta_t = \rho\theta_{t-1} + w_t$$

  w_t i.i.d. as $N(0, \sigma_w^2)$.
- From here, we use $\rho = 1$, i.e. a first order random walk, for clarity of explanation.
- Recall that from a Bayesian perspective, the second (temporal) stage may be viewed as a prior distribution for $\theta' = (\theta_1, ..., \theta_T)$, and that $p(\theta|\sigma_w^2)$, can be expressed as

$$p(\theta_t|\theta_{-t}, \sigma_w^2) \sim \begin{cases} N(\theta_{t+1}, \sigma_w^2) & \text{for } t = 1, \\ N\left(\frac{\theta_{t-1}+\theta_{t+1}}{2}, \frac{\sigma_w^2}{2}\right) & \text{for } t = 2, ..., T-1, \\ N(\theta_{t-1}, \sigma_w^2) & \text{for } t = T. \end{cases}$$

  where θ_{-t} represents the vector of θ's with θ_t removed. It is noted that σ_w^2 is a *conditional* variance and so it is not comparable to σ_v^2.
- This is the reason for

```
 for (t in 2:(n-1))
```

  and defining the end points separately.

- In WinBUGS

```
model {
       for (t in 2:(T-1)) {
.
# system model
              tmp.theta[t] <- (theta[t-1]+theta[t+1])/2
              theta[t] ~ dnorm(tmp.theta[t],tau.w2)
.
} # t loop
.
theta[1]~dnorm(theta[2],tau.w)
theta[T]~dnorm(theta[n-1],tau.w)
.
tau.w ~ dgamma(r.w,d.w)
sigma.w <- 1 / sqrt(tau.w)
  } # end of model
```

- Note that because we are dealing with dealing with the cyclical graph at this stage, unless we make specific allowance there will be double counting of the likelihood terms (where for example θ will appear as both a parent of θ_{t-1} and as a child of θ_{t+1} and so we have to either
  - explicitly specify some of the full conditional distributions (using the RW structure). It is possible to do this in WinBUGS, although not widely documented. On the previous slide we need to explicitly find the contribution of the likelihood (the data) to the posterior for σ_w^2, i.e. `r.w` and `d.w` in

    ```
    tau.w ~ dgamma(r.w,d.w)
    ```
  - use an in-built WinBUGS function which allows for this, using the equivalence with a intrinsic CAR (conditionally autoregressive) model.

- Note that Gamma prior and with Normal likelihood combine to give a Gamma posterior.

$p(\theta|\tau_w) \sim N(\theta_{t-1}, \tau_w)$. Note use of $\tau_w = 1/\sigma_w^2$.
$p(\tau_w) \sim Ga(r, d)$

$$\begin{aligned} p(\tau_w|\theta) &\propto d^r \tau_w^{(r-1)} \exp(-d\tau_w) \\ &\times \tau_w^{(n/2)} \exp\{\tau/2 \sum_{t=2}^{N} (\theta_t - \theta_{t-1})^2\} \\ &\propto d^r \tau_w^{(r+n/2-1)} \exp(\tau_w \{d + \sum_{t=2}^{N} (\theta_t - \theta_{t-1})^2)\} \end{aligned}$$

- So the posterior
$p(\tau_w|\theta) \sim Ga(r + n/2, d + \sum_{t=2}^{N} (\theta_t - \theta_{t-1})^2/2)$

Specifying the full conditionals

- We need to calculate the contribution of the likelihood ourselves and then combine this with the prior to give the posterior.
- In WinBUGS

```
model {
       for (t in 2:(T-1)) {
.
# calculate the contribution to the likelihood for
# full conditionals
tau.w.like[t] <-pow((theta[t]-tmp.theta[t]),2)
.
} # t loop
.
tau.w.like[1] <-  0
tau.w.like[T] <-  pow((theta[T]-theta[T-1]),2)
.
} # model
```

- In WinBUGS

```
.
tau.w2 <- tau.w*2
d <-1
r <- 0.01
     d.w <- d+sum(tau.w.like[])/2
     r.w <- r + n/2
      tau.w ~ dgamma(r.w,d.w)
      .
```

- Note this uses a prior of $Ga(1, 0.01)$ for τ_w which is ‘hard-wired’ into the code at this point, the values of r and d could also be an input to the model in the form of data.

- The whole model in WinBUGS, `model1.odc`

```
# Single site, one pollutant (note likelihood calculations because of the cyclical model)
model {

       for (t in 2:(T-1)) {

# observation model
              y[t] ~ dnorm(theta[t],tau.v)

# system model
              tmp.theta[t] <- (theta[t-1]+theta[t+1])/2
              theta[t] ~ dnorm(tmp.theta[t],tau.w2)

# calculate the contribution to the likelihood for full conditionals
tau.w.like[t] <-pow((theta[t]-tmp.theta[t]),2)
} # t loop
# need to define the end points separately
theta[1]~dnorm(theta[2],tau.w)
theta[T]~dnorm(theta[T-1],tau.w)
y[1]~dnorm(theta[1],tau.v)
y[T]~dnorm(theta[T],tau.v)

# calculate the contributions to likelihood & full conditionals
tau.w.like[1] <-     0
tau.w.like[T] <-      pow((theta[T]-theta[T-1]),2)
tau.w2 <- tau.w*2
d <-1
r <- 0.01
     d.w <- d+sum(tau.w.like[])/2
     r.w <- r + T/2
      tau.v ~ dgamma(1,0.01)
      tau.w ~ dgamma(r.w,d.w)
      sigma2.v<-1/tau.v
      sigma.v<-sqrt(sigma2.v)
      sigma2.w <- 1 / tau.w
      sigma.w<-sqrt(sigma2.w)

 } # end model
```

- Data for single site: PM$_{10}$ at Bloomsbury site, `model1-data.odc`.

```
list(T = 1461, y = c(66, 49, 35, 40, NA, NA, 22, 32, 17, 14, 17, 18, 20, 21, 26,
 24, 24, 29, 23, 25, 23, 28, 28, 38,
49, 51, 48, 46, 55, 41, 37, 24, 33,
75, 76, 70, 46, 55, 61, 29, 24, 24,
...,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA))
```

- Initial values (for chain 1), `model1-init1.odc`

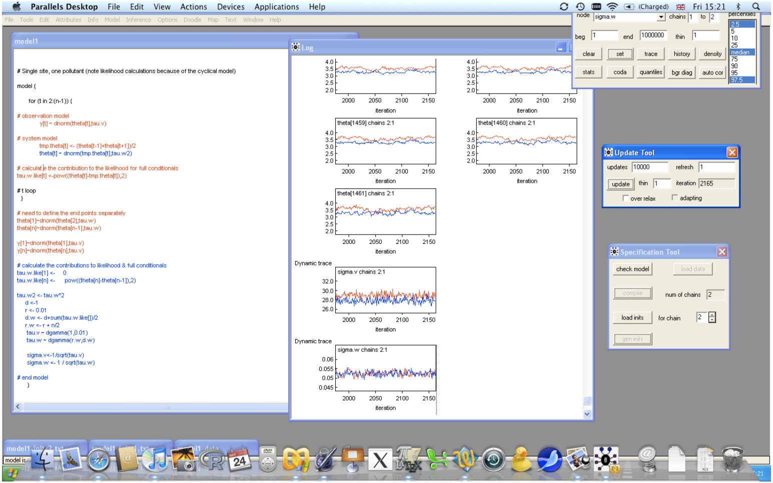
  ```
  list(tau.v = 1, tau.w = 1, theta = c(3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  ...,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3),

  y = c(NA, NA, NA, NA, 3, 3, NA, NA, NA, NA,
  NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
   NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
  ...,
  NA, NA, NA, NA, NA, NA, NA, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3))
  ```

- Note requirement to provide initial values for the missing values of `y`. Where there is data, i.e. not a random variable, need to put `NA`.

Need to set the parameters which you want to keep

- `theta` - if you are interested in keeping all of them (there are 1461 of them, one for each day)
- `theta[i]` - if you want to keep a single one of them
- `theta[i:j]` or `theta[c(3,56,987)]` - if you want to keep a selection
- `sigma.v` - the variance of the random error from the first level of the model
- `sigma.w` - the variance of the random walk process from the second level of the model

Note that convergence is likely to take much longer than in simple examples!



model1

```
# Single site, one pollutant (note likelihood calculations because of the cyclical model)

model {

      for (t in 2:(n-1)) {

# observation model
            y[t] ~ dnorm(theta[t],tau.v)

# system model
            tmp.theta[t] <- (theta[t-1]+theta[t+1])/2
            theta[t] ~ dnorm(tmp.theta[t],tau.w2)

# calculate the contribution to the likelihood for full conditionals
tau.w.like[t] <-pow((theta[t]-tmp.theta[t]),2)

# t loop
  }

# need to define the end points separately
theta[1]~dnorm(theta[2],tau.w)
theta[n]~dnorm(theta[n-1],tau.w)

y[1]~dnorm(theta[1],tau.v)
y[n]~dnorm(theta[n],tau.v)

# calculate the contributions to likelihood & full conditionals
tau.w.like[1] <-     0
tau.w.like[n] <-      pow((theta[n]-theta[n-1]),2)

tau.w2 <- tau.w*2
    d <-1
    r <- 0.01
    d.w <- d+sum(tau.w.like[])/2
    r.w <- r + n/2
     tau.v ~ dgamma(1,0.01)
     tau.w ~ dgamma(r.w,d.w)

     sigma.v<-1/sqrt(tau.v)
     sigma.w <- 1 / sqrt(tau.w)

# end model
     }
```

node sigma.w   chains 1 to 2

beg 1   end 1000000   thin 1

Update Tool

updates 10000   refresh 1

thin 1   iteration 2165

□ over relax   □ adapting

Specification Tool

num of chains 2

for chain 2

Exercises

Without WinBUGS

1. Show that a random walk process of order 1 can be expressed in terms of an intrinsic CAR model, i.e. if $p(\theta_t|\theta_{t-1}) \sim N(\theta_{t-1}, \sigma_w^2)$ then

$$p(\theta_t|\theta_{-t}, \sigma_w^2) \sim \begin{cases} N(\theta_{t+1}, \sigma_w^2) & \text{for } t = 1, \\ N\left(\frac{\theta_{t-1}+\theta_{t+1}}{2}, \frac{\sigma_w^2}{2}\right) & \text{for } t = 2, ..., T-1, \\ N(\theta_{t-1}, \sigma_w^2) & \text{for } t = T. \end{cases}$$

   where θ_{-t} represents the vector of θ's with θ_t removed. Pay particular attention to any assumptions that need to be made when $t = 1$ and $t = T$.

2. Show that a Gamma prior, $\tau_w \; Ga(a, b)$ combines with normal likelihood, $[\theta_t|\theta_{t-1}, \tau_2] \sim N(\theta_{t-1}, \tau_w)$, to give a Gamma posterior, paying particular attention to the form of the updated parameters.

Exercises

Using WinBUGS

1. Open `model1.odc` and load the data (`model1-data.odc`) and compile the model with two chains. Initial values can be found for two chains in `model1-inits1.odc` and `model2-inits2.odc`.
2. Run the model for a suitable number of iterations and calculate summary statistics for the posterior distributions of `theta`, `sigma.v` and `sigma.w`.
3. In R (or other package) plot the estimated values `theta` against the observed data, `y`. What do you conclude? Note that you may have to deal with the different lengths of the two series, remember that `theta` has no missing values in it.
4. Plot a suitable summary of the posterior values of `theta` (including their uncertainty) against time. What do you conclude about the uncertainty in the values of `theta` when the original data is missing?

Conditional (Spatial) Models

- Remember (or look up in the notes) the Scottish lip cancer model in which we proposed a simple Poisson-Gamma regression model.
- Before we considered an empirical Bayes approach, which has the advantage of being easy to fit but cannot be expanded to do spatial smoothing and is not quite 'right' statistically.
- Now we consider a fully Bayesian approach, which requires a prior distribution on regression parameters and variance parameters of random effects distribution.

$$\begin{aligned} Y_i|\theta_i, \beta_0 &\sim \text{Poisson}(E_i \mathrm{e}^{\beta_0}\theta_i) \\ \theta_i &\sim \text{Ga}(\alpha, \alpha) \end{aligned}$$

We require priors for β_0 and α. For example:

$$\beta_0 \sim \text{N}(m, v)$$
$$\alpha \sim \text{Ga}(a, b)$$

with m, v, a, b picked to reflect beliefs about β_0 and α.

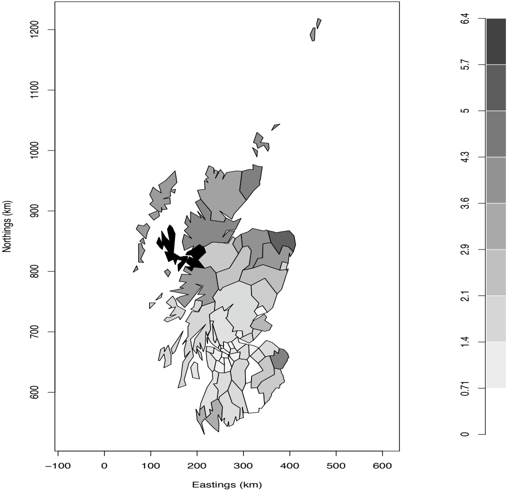


Figure: SMRs for Scottish counties.

Empirical Bayes for Scotland

We recap on the previous analyses – this involved maximum likelihood estimation for β_0 and α in a negative binomial model and produced:

```
> emp0 <- eBayes(z$Y,z$E)
> emp0$beta
  0.3521065
> emp0$alpha
[1] 1.87949
> emp0$RR
 [1] 3.9973624 4.0791107 2.9802133 2.8467916 3.0025773 2.6545872 2.9590825
 [8] 2.4517687 2.3721492 2.7619805 2.6005515 2.2037872 2.0149301 2.1376464
...
[43] 0.6900960 0.4948910 0.4013614 0.5124617 0.5604849 0.4593902 0.3319144
[50] 0.3766186 0.6098460 0.5850639 0.4100864 0.3460232 0.3403845 0.6020789
> emp0$RRmed
 [1] 3.8755781 4.0458981 2.9034476 2.7600608 2.9434956 2.5655788 2.9237792
 [8] 2.3603697 2.2725880 2.7200177 2.5425312 2.0979757 1.8790820 2.0659710
...
[43] 0.6317935 0.4741200 0.3949723 0.4779112 0.5131326 0.4284178 0.3282190
[50] 0.3608116 0.5408883 0.5189084 0.3637163 0.3068970 0.2822885 0.4993176
```

WinBUGS analysis of the Poisson-Gamma model

In the example that follows we specify a flat prior for β_0, and a Ga(1,1) prior for α.

The iterative algorithm is run for 10,000 iterations, with the first 4,000 discarded as “burn-in”.

We summarize the posteriors for the relative risks:

$$\text{RR}_i = \exp(\beta_0)\theta_i$$

and for β_0 and α. The posterior mean for β_0 is 0.36, compared to 0.35 under empirical Bayes, and the posterior mean for α is 1.79, compared to 1.88 under empirical Bayes.

Similarly the posterior means and posterior medians agree very closely.

```
model
{
  for (i in 1 : N) {
      Y[i]  ~ dpois(mu[i])
      mu[i] <- E[i]*exp(beta0)*theta[i]
      RR[i] <- exp(beta0)*theta[i]
      theta[i] ~ dgamma(alpha,alpha)
  }
# Priors
  alpha ~ dgamma(1,1)
  beta0 ~ dflat()
# Functions of interest:
  sigma.theta <- sqrt(1/alpha)   # standard deviation of non-spatial
  base <- exp(beta0)
}
```

```
DATA
list(N = 56,
  Y = c( 9, 39, 11, 9, 15, 8, 26, 7, 6, 20, 13, 5, 3, 8, 17, 9, 2, 7,
  9, 7, 16, 31, 11, 7, 19, 15, 7, 10, 16, 11, 5, 3, 7, 8, 11, 9, 11,
  8, 6, 4, 10, 8, 2, 6, 19, 3, 2, 3, 28, 6, 1, 1, 1, 1, 0, 0), E = c(
  1.4, 8.7, 3.0, 2.5, 4.3, 2.4, 8.1, 2.3, 2.0, 6.6, 4.4, 1.8, 1.1,
  3.3, 7.8, 4.6, 1.1, 4.2, 5.5, 4.4, 10.5,22.7, 8.8, 5.6,15.5,12.5,
  6.0, 9.0,14.4,10.2, 4.8, 2.9, 7.0, 8.5,12.3,10.1,12.7, 9.4, 7.2,
  5.3, 18.8,15.8, 4.3,14.6,50.7, 8.2, 5.6, 9.3,88.7,19.6, 3.4, 3.6,
  5.7, 7.0, 4.2, 1.8))

INTIAL ESTIMATES
list(alpha = 1, beta0 = 0,
  theta=c(1,1,1,1,1,1,1,1,1,1,1,1,1,
  1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
  1,1,1,1,1,1,1,1,1,1,1,1,1))
```

```
node    mean    sd      MC error 2.5%   median  97.5%   start   sample
RR[1]   4.07    1.297   0.01877 1.959   3.92    7.001   4000    6001
RR[2]   4.105   0.6469  0.00864 2.938   4.068   5.48    4000    6001
RR[3]   3.006   0.858   0.01159 1.607   2.915   4.937   4000    6001
RR[4]   2.875   0.8995  0.01019 1.391   2.773   4.886   4000    6001
RR[5]   3.016   0.7406  0.01114 1.754   2.955   4.668   4000    6001
RR[6]   2.68    0.8865  0.01325 1.227   2.568   4.696   4000    6001
RR[7]   2.975   0.5666  0.00830 1.994   2.929   4.236   4000    6001
RR[8]   2.476   0.8492  0.01224 1.082   2.379   4.412   4000    6001
....
RR[49]  0.3321  0.06051 7.88E-4 0.2261  0.3286  0.4612  4000    6001
RR[50]  0.3685  0.1334  0.00162 0.1603  0.3522  0.6725  4000    6001
RR[51]  0.6     0.3539  0.00424 0.1112  0.5327  1.45    4000    6001
RR[52]  0.5702  0.3425  0.00519 0.1034  0.5017  1.4     4000    6001
RR[53]  0.4021  0.2446  0.00316 0.07137 0.3546  0.9934  4000    6001
RR[54]  0.3327  0.2042  0.00227 0.05706 0.2924  0.8143  4000    6001
RR[55]  0.3259  0.2533  0.00345 0.02491 0.2646  0.9605  4000    6001
RR[56]  0.5814  0.4538  0.00636 0.04737 0.4723  1.745   4000    6001
alpha   1.79    0.3985  0.00792 1.129   1.753   2.682   4001    6000
beta0   0.3567  0.1188  0.00591 0.1315  0.353   0.5966  4000    6001
```

Poisson-Lognormal Model

The Poisson-gamma model offers analytic tractability, but does not easily allow the incorporation of spatial random effects.
A Poisson-lognormal non-spatial random effect model is given by:

$$Y_i|\beta, V_i \sim_{ind} \text{Poisson}(E_i \mu_i \mathrm{e}^{V_i}) \quad V_i \sim_{iid} N(0, \sigma_v^2)$$

where V_i are area-specific random effects that capture the residual or unexplained (log) relative risk of disease in area i, $i = 1, ..., n$.
Whereas in the Poisson-Gamma model we have $\theta \sim \text{Ga}(\alpha, \alpha)$, here we have $\theta = \mathrm{e}^{V_i} \sim \text{LogNormal}(0, \sigma^2)$.
This model does not give a marginal distribution of known form, but does naturally lead to the addition of spatial random effects.
The marginal variance is of the same quadratic form as with the negative-binomial model.

Non-Spatial Analysis of the Scottish Lip Cancer Data

We now report a fully Bayesian version of the normal model, with log-linear cubic model.

The covariates are centered here in order to reduce dependence in the parameter estimates, which reduces the computational burden; this model was fitted using so-called Markov chain Monte Carlo via the `WinBUGS` software.

Flat priors were placed on $\beta_0, \beta_1, \beta_2, \beta_3$ and a Ga$(1, 0.0260)$, was assumed for σ_v^{-2}.

WinBUGS code

```
model {
  for (i in 1 : N) {
      Y[i]  ~ dpois(mu[i])
      X1c[i] <- X[i]-mean(X[1:N])
      X2c[i] <- X1c[i]*X1c[i]
      X3c[i] <- X1c[i]*X1c[i]*X1c[i]
      log(mu[i]) <- log(E[i]) + beta0 +
                    beta1*X1c[i] + beta2*X2c[i] + beta3*X3c[i] + V[i]
      RR[i] <- exp(beta0 + beta1*X1c[i] + beta2*X2c[i]+ beta3*X3c[i] + V[i])
      V[i] ~ dnorm(0,tau.V)
  }
# The gamma prior corresponds to df=2, q=0.95, R=log 2.
  tau.V ~ dgamma(1,0.0260)
  beta0 ~ dflat()
  beta1 ~ dflat()
  beta2 ~ dflat()
  beta3 ~ dflat()
# Functions of interest:
  sigma.V <- sqrt(1/tau.V)       # standard deviation of non-spatial
  RRRlo <- exp(-1.96*sigma.V)
  RRRhi <- exp(1.96*sigma.V) }
```

Spatial Models

- In general we might expect residual relative risks in areas that are "close" to be more similar than in areas that are not "close".
- We would like to exploit this information in order to provide more reliable relative risk estimates in each area.
- This is analogous to the use of a covariate x, in that areas with similar x values are likely to have similar relative risks.
- Unfortunately the modelling of spatial dependence is much more difficult since spatial location is acting as a surrogate for unobserved covariates.
- We need to choose an appropriate spatial model, but do not directly observe the covariates whose effect we are trying to mimic.

We first consider the model

$$Y_i|\beta, \gamma, \mathbf{U_i}, \mathbf{V_i} \sim_{\mathbf{ind}} \text{Poisson}(\mathbf{E_i}\mu_{\mathbf{i}}\text{e}^{\mathbf{U_i}+\mathbf{V_i}})$$

with

$$\log \mu_i = g(\mathbf{S}_i, \gamma) + \mathbf{f}(\mathbf{x_i}, \beta), \tag{1}$$

where

- $\mathbf{S}_i = (S_{i1}, S_{i2})$ denotes spatial location, the centroid of area i,
- $f(\mathbf{x_i}, \beta)$ is a regression model,
- $g(\mathbf{S}_i, \gamma)$ is an expression that we may include to capture large-scale spatial trend – the form

  $$f(\mathbf{S}_i) = \gamma_1 S_{i1} + \gamma_2 S_{i2},$$

  is a simple way of accommodating long-term spatial trend.
- The random effects $V_i \sim_{iid} N(0, \sigma_v^2)$ represent non-spatial overdispersion,
- U_i are random effects with spatial structure.

- In spatial epidemiology and disease mapping, one approach is to specify the distribution of the random effect in a particular area, U_i, as if we knew the values of the spatial random effects, U_j, in "neighboring areas"
- We therefore need to specify a rule for determining the "neighbours" of each area.
- Spatial models that start with the n area-specific residual spatial random effects all suffer from a level of arbitrariness in their specification – in an epidemiological context the areas are not regular in shape (as opposed to images for example, which are on a regular grid).
- To define *neighbours*, a number of authors have taken the neighborhood scheme to be such that areas i and j are taken to be neighbors if they share a *common boundary*. This is reasonable if all regions are of similar size and arranged in a regular pattern (as is the case for pixels in image analysis where these models originated), but is not particularly attractive otherwise.

- Various other neighborhood/weighting schemes are possible.
- We could take the neighborhood structure to depend on the distance between area centroids and determine the extent of the spatial correlation (i.e. the distance within which regions are considered neighbors).
- In typical applications it is difficult to assess whether the spatial model chosen is appropriate, which argues for a simple form, and to assess the sensitivity of conclusions to different choices.
- In Figure **??** we show a close-up of a portion of the Birmingham study. One of the wards in the center of the Birmingham region is such that it ‘just’ shares a common boundary with a number of close-by wards. In terms of the common-boundary prior, it could be considered to have between four and ten neighbors.

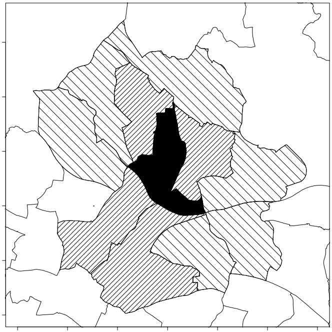


Figure: Close-up of a region of the Birmingham study.

The ICAR model

- A common model is to assign the spatial random effects an intrinsic conditional autorgressive (ICAR) prior.
- Under this specification it is assumed that

$$U_i | U_j, j \in \partial_i \sim N\left(\overline{U}_i, \frac{\omega_u^2}{m_i}\right),$$

  where ∂i is the set of neighbors of area i, m_i is the number of neighbours, and $\overline{U}_i$ is the mean of the spatial random effects of these neighbors.
- The parameter ω_u^2 is a conditional variance and its magnitude determines the amount of spatial variation.
- The variance parameters σ_v^2 and ω_u^2 are on different scales, σ_v is on the log odds scale while ω_u is on the log odds scale, *conditional* on $U_j, j \in \partial i$; hence they are not comparable.

- Notice that if ω_u^2 is "small" then although the residual is strongly dependent on the neighboring value the overall contribution to the residual relative risk is small.
- This is a little counterintuitive but stems from spatial models having two aspects, strength of dependence and total amount of spatial dependence, and in the ICAR model there is only a single parameter which controls both aspects.

`WinBUGS` *representation*

The ICAR model can be specified via the function:

```
U[1:N] ~ car.normal(adj[],weights[],num[],tau)
```

where:

- `adj[]`: A vector listing the ID numbers of the adjacent areas for each area (this can be generated using the Adjacency Tool from the Map menu in `GeoBUGS`).
- `weights[]`: A vector the same length as `adj[]` giving unnormalized weights associated with each pair of areas.
- `num[]`: A vector of length N (the total number of areas) giving the number of neighbors n_i for each area.
- The `car.normal` distribution is parameterized to include a sum-to-zero constraint on the random effects. A separate intercept term must be used in the model and this must be assigned an improper uniform prior using the `dflat()` distribution (see full code below).

The WinBUGS code for the ICAR model

```
model {
  for (i in 1 : N) {
      Y[i]  ~ dpois(mu[i])
      X1c[i] <- X[i]-mean(X[1:N])
      X2c[i] <- X1c[i]*X1c[i]
      X3c[i] <- X1c[i]*X1c[i]*X1c[i]
      log(mu[i]) <- log(E[i]) + beta0 + beta1*X1c[i]  +
                    beta2*X2c[i] + beta3*X3c[i] + V[i] + U[i]
      RR[i] <- exp(beta0 + beta1*X1c[i] +
                   beta2*X2c[i] + beta3*X3c[i] + V[i] + U[i])
      V[i] ~ dnorm(0,tau.V)
  }
# ICAR prior distribution for spatial random effects:
  U[1:N] ~ car.normal(adj[], weights[], num[], tauomega.U)
  for(k in 1:sumNumNeigh) {
      weights[k] <- 1
  }
  tau.T ~ dgamma(1,0.0260)
  p ~ dbeta(1,1)
  sigma.Z <- sqrt(p/tau.T)
  omega.U <- sigma.Z/sqrt(1.164)
  sigma.V <- sqrt((1-p)/tau.T)
  tau.V <- 1/(sigma.V*sigma.V)
  tauomega.U <- 1/(omega.U*omega.U)
  beta0 ~ dflat()
  beta1 ~ dflat()
  beta2 ~ dflat()
  beta3 ~ dflat()
  sd.U <- sd(U[1:N])
  vratio <- sd.U*sd.U/(sd.U*sd.U+sigma.V*sigma.V)
}
```

```
DATA

list(N = 56, Y = c( 9, 39, 11, 9, 15, 8, 26, 7, 6, 20, 13, 5, 3, 8,
17, 9, 2, 7, 9, 7, 16, 31, 11, 7, 19, 15, 7, 10, 16, 11, 5, 3, 7, 8,
11, 9, 11, 8, 6, 4, 10, 8, 2, 6, 19, 3, 2, 3, 28, 6, 1, 1, 1, 1, 0,
0), E = c( 1.4, 8.7, 3.0, 2.5, 4.3, 2.4, 8.1, 2.3, 2.0, 6.6, 4.4, 1.8,
1.1, 3.3, 7.8, 4.6, 1.1, 4.2, 5.5, 4.4, 10.5,22.7, 8.8, 5.6,15.5,12.5,
6.0, 9.0,14.4,10.2, 4.8, 2.9, 7.0, 8.5,12.3,10.1,12.7, 9.4, 7.2, 5.3,
18.8,15.8, 4.3,14.6,50.7, 8.2, 5.6, 9.3,88.7,19.6, 3.4, 3.6, 5.7, 7.0,
4.2, 1.8), X = c(0.16,0.16,0.10,0.24,0.10,0.24,0.10, 0.07, 0.07,0.16,
0.07,0.16,0.10,0.24, 0.07,0.16,0.10, 0.07, 0.07,0.10, 0.07,0.16,0.10,
0.07, 0.01, 0.01, 0.07, 0.07,0.10,0.10, 0.07,0.24,0.10, 0.07, 0.07,
0,0.10, 0.01,0.16, 0, 0.01,0.16,0.16, 0, 0.01, 0.07, 0.01, 0.01, 0,
0.01, 0.01, 0, 0.01, 0.01,0.16,0.10),
num = c(3, 2, 2, 3, 4, 2, 5, 1, 5, 4, 1, 2, 3, 3, 2, 6, 6, 6, 5, 3,
3, 2, 4, 8, 3, 3, 4, 4, 11, 6, 7, 3, 4, 9, 4, 2, 4, 6, 3, 4,
5, 5, 4, 5, 4, 6, 6, 4, 9, 2, 4, 4, 4, 5, 6, 5),
adj = c(
19, 9, 5,
10, 7,
12, 6,
28, 20, 18,
19, 12, 11, 1,
3,8,
17, 16, 13, 10, 2,
6,
29, 23, 19, 17, 1,
22, 16, 7, 2,
5,
5, 3,
19, 17, 7,
35, 32, 31,
29, 25,
...
53, 49, 48, 46, 31, 24,
49, 47, 44, 24,
54, 53, 52, 48, 47, 44, 41, 40, 38,
29, 21,
54, 42, 38, 34,
54, 49, 40, 34,
49, 47, 46, 41,
52, 51, 49, 38, 34,
56, 45, 33, 30, 24, 18,
55, 27, 24, 20, 18
),
sumNumNeigh = 240))

INITIAL ESTIMATES

list(tau.T = 1, p=0.5, beta0 = 0, beta1 = 0, beta2 = 0, beta3 = 0,
V=c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
U=c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0))
```

Figure **??** shows the centroids for each area, allowing us to confirm the number and labels of the neighbors of each area.

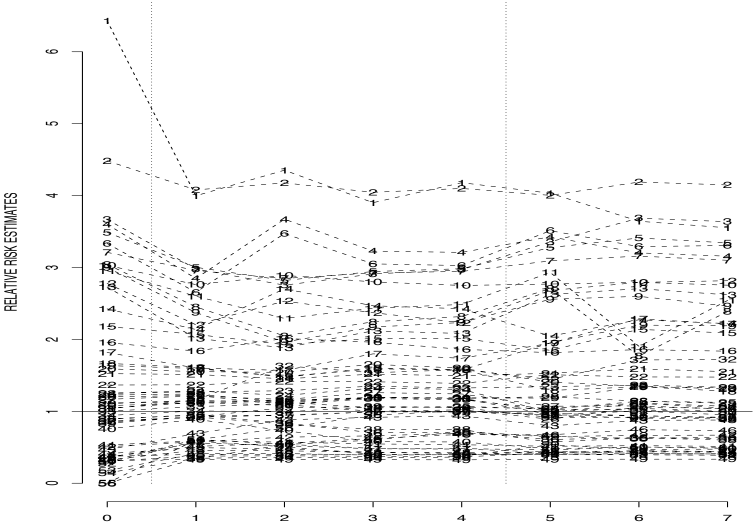


Figure:

Figure details: Relative risk estimates for Scottish lip cancer data:

0 denote the SMRs;

1 the empirical Bayes estimates without the use of AFF;

2 the empirical Bayes estimates with log link and a linear model in AFF;

3 the empirical Bayes estimates with a log-linear cubic model in AFF;

4 the fully Bayes non-spatial estimates with a log-linear cubic model in AFF;

5 estimates under the joint model;

6 estimates under the initial ICAR model;

7 estimates under the refined ICAR model. Estimates 5–7 are based upon a log-linear cubic covariate model.

Plotting symbol is county number.

Back to the temporal pollution model - using the `car.normal` distribution to represent the RW(1) process.

$$p(\theta_t|\theta_{-t},\sigma^2_w) \sim \begin{cases} N(\theta_{t+1},\sigma^2_w) & \text{for } t=1, \\ N\left(\frac{\theta_{t-1}+\theta_{t+1}}{2},\frac{\sigma^2_w}{2}\right) & \text{for } t=2,...,T-1, \\ N(\theta_{t-1},\sigma^2_w) & \text{for } t=T. \end{cases}$$

where θ_{-t} represents the vector of θ's with θ_t removed.

This is equivalent to specifying $\theta_t|\theta_{-t} \sim N(\sum_k C_{tk}\theta_k, \sigma^2_w M_{tt})$ where $C_{tk} = W_{tk}/W_{t+}, W_{t+} = \sum_k W_{tk}$ and $W_{tk} = 1$ if $k = (t-1)$ or $(t+1)$ and 0 otherwise; $M_{tt} = 1/W_{t+}$

Hence the RW(1) prior may be fitted using the `car.normal distribution` in WinBUGS, with appropriate specification of the weight and adjacency matrices, and vector representing the number of neighbours.

Note that if the observed time points are not equally spaced, it is necessary to include missing values (NA) for the intermediate time points.

This prior may be specified in WinBUGS using the `car.normal` distribution,

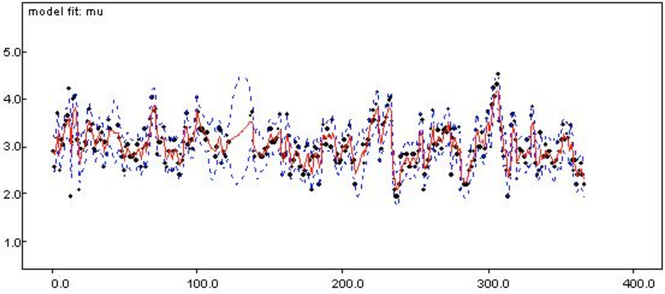
- with adjacency vector `adj[]` listing neighbouring time points, i.e. $(t-1)$ and $(t+1)$ are neighbours of time point t,
- corresponding weight vector `weight[]` set to a sequence of 1's,
- and a vector giving the number of neighbours, `num[]`, set to 2 for all time points except `num[1]` and `num[T]` which are set to 1.

Model 1 using `car.normal`, in file `model1CARNORMAL.odc`.

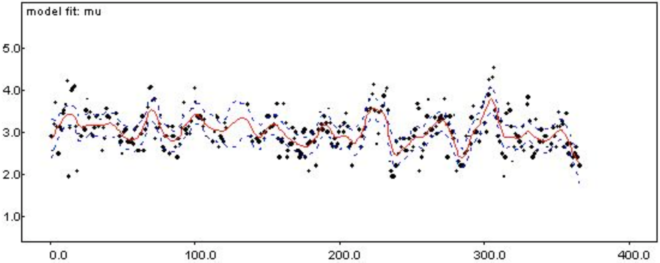
```
model {

# likelihood
for(t in 1:T) {
y[t] ~ dnorm(mu[t], tau.v)
mu[t] <- beta + theta[t]
}

# prior for temporal effects
# RW prior for theta[t] - specified using car.normal with neighbours (t-1) and (t+1)
# for theta[2],....,theta[T-1], and neighbours (t+1) for theta[1] and (t-1) for theta[T]

theta[1:T] ~ car.normal(adj[], weights[], num[], tau)
beta~dflat()
.
.
# Specify weight matrix and adjacency matrix corresponding to RW(1) prior
# (Note - this could be given in the data file instead)

for(t in 1:1) {
weights[t] <- 1; adj[t] <- t+1; num[t] <- 1
}
for(t in 2:(T-1)) {
weights[2+(t-2)*2] <- 1; adj[2+(t-2)*2] <- t-1
weights[3+(t-2)*2] <- 1; adj[3+(t-2)*2] <- t+1; num[t] <- 2
}
for(t in T:T) {
weights[(T-2)*2 + 2] <- 1; adj[(T-2)*2 + 2] <- t-1; num[t] <- 1
}

# other priors
tau.err  ~ dgamma(0.01, 0.01) # measurement error precision
sigma.v <- 1 / sqrt(tau.v)
sigma2.v <- 1/tau.v

tau.w  ~ dgamma(0.01, 0.01) # random walk precision
sigma.w <- 1 / sqrt(tau.w)
sigma2.w <- 1/tau.w
} # model
```

- The following is a plot of posterior median (red line) and posterior 95% intervals (dashed blue lines) for `mu[t]` (the underlying mean daily pollutant concentration), with observed concentrations shown as black dots.
- This plot was produced by selecting the model fit option from the Compare menu (available from the Inference menu), with `mu` specified as the node, day as the axis and y as other).
- Note that the dashed blue line shows the posterior 95% interval for the estimated mean daily concentration, and is not a predictive interval - hence we would not necessarily expect all of the observed data points to lie within the interval.

Using RW(1) model



Equivalent plot assuming an RW(2) prior. Note the greater amount of smoothing imposed by this prior



Exercises

1. Run the changed version of model1 using the `car.normal` distribution using the same PM_{10} data from the Bloomsbury site (`model1-data.odc`).
2. Show that a random walk of order two $\theta_t \sim N(\theta_{t-1} + \theta_{t-2}, \sigma_w^2)$ can be expressed as an intrinsic CAR model and defined the contents of the matrices `adj[]` and `weights[]` and the vector `num[]` required to fit this within WinBUGS.
3. Change the model you fitted in the first question to fit a second order random walk, compile the model and comment on the differences in the results from those observed when fitting a first order random walk.

Recall the model

$$Y_i|\beta, \gamma, \mathbf{U_i}, \mathbf{V_i} \sim_{\mathbf{ind}} \text{Poisson}(\mathbf{E_i}\mu_\mathbf{i}\text{e}^{\mathbf{U_i}+\mathbf{V_i}})$$

with

$$\log \mu_i = g(\mathbf{S}_i, \gamma) + \mathbf{f}(\mathbf{x_i}, \beta), \tag{2}$$

where

- $\mathbf{S}_i = (S_{i1}, S_{i2})$ denotes spatial location, the centroid of area i,
- $f(\mathbf{x_i}, \beta)$ is a regression model,
- $g(\mathbf{S}_i, \gamma)$ is an expression that we may include to capture large-scale spatial trend – the form

  $$f(\mathbf{S}_i) = \gamma_1 S_{i1} + \gamma_2 S_{i2},$$

  is a simple way of accommodating long-term spatial trend.
- The random effects $V_i \sim_{iid} N(0, \sigma_v^2)$ represent non-spatial overdispersion,
- U_i are random effects with spatial structure which we now consider ‘jointly’ rather than considering neighbours.

A Joint Model

- Assume that $\mathbf{U} = (U_1, ..., U_n)$ arise from a zero mean multivariate normal distribution with variances var$(U_i) = \sigma_u^2$ and correlations corr$(U_i, U_j) = \exp(-\phi d_{ij}) = \rho^{d_{ij}}$ where d_{ij} is the distance between the centroids of areas i and j, and $\rho > 0$ is a parameter that determines the extent of the correlation.
- This model is *isotropic* since it assumes that the correlation is the same in all spatial directions. We refer to this as the *joint* model, since we have specified the joint distribution for $\mathbf{U}$.
- More generally the correlations can be modeled as corr$(U_i, U_j) = \exp(-(\phi d_{ij})^\kappa)$.

WinBUGS *representation*

The above model with

$$\text{cov}(U_i, U_j) = \tau_u^{-1} \exp(-(\phi d)^\kappa)$$

and $\phi > 0$, $0 < \kappa < 2$ can be specified via the function:

$$\texttt{U[1:N]} \sim \texttt{spatial.exp(mu[],x[],y[],tau,phi,kappa)}$$

where:

- `mu[]`: A vector giving the mean for each area.
- `x[]` and `y[]`: Vectors of length n (the number of areas) giving the x and y coordinates of the centroid of each area.
- `phi` $= \phi$.
- `kappa` $= \kappa$.
- This model can be very slow for even moderate sized datasets (because a matrix inversion is required at each iteration).

Single pollutant, multiple monitoring site

- S monitoring sites measuring a single pollutant.
- The underlying autoregressive structure remains constant across sites with a constant adjustment in the mean level for site s by an amount m_s, $s = 1, ..., S$.
- **Stage One, Observed Data Model:**

  $$Y_{st} = X'_{st}\beta_1 + X'_s\beta_2 + m_s + \theta_t + v_{st}$$

  with v_{st} i.i.d. as $N(0, \sigma^2_{vs})$ and β_1, β_2, $q_1 \times 1$ and $q_2 \times 1$ vectors of site/day and site only regression coefficients.
- Dropping the covariate terms for clarity of explanation.

- **Stage Two (a), Temporal Model:**

  $$\theta_t = \rho\theta_{t-1} + w_t$$

  with w_t i.i.d. as $N(0, \sigma_w^2)$.
- Considering a RW(1) process for clarity of explanation, i.e. $\rho = 1$ in the AR(1) process.

```
 model {

              for (t in 2:(n-1)) {
for (site in 1:8) {

# y arises from the underlying theta, plus site parameter& measurement error
                y.mat[t,site] ~ dnorm(mean.site[t,site],tau.v[site])
mean.site[t,site] <- theta[t]  +m.adj[site]
} # site loop

# the underlying theta is an average of the two neighbours
            tmp.theta[t] <- (theta[t-1]+theta[t+1])/2
                 theta[t] ~ dnorm(tmp.theta[t],tau.w2)
                tau.w.like[t] <-pow((theta[t]-theta[t-1]),2)

                 # the underlying theta is an average of the two neighbours
            tmp.theta[t] <- (theta[t-1]+theta[t+1])/2
                 theta[t] ~ dnorm(tmp.theta[t],tau.w2)
                tau.w.like[t] <-pow((theta[t]-theta[t-1]),2)
                } # t loop
.
```

- **Stage Two (b), Spatial Model:**
  The random effects $m = (m_1, ..., m_S)'$ arise from the multivariate normal distribution

  $$m \sim MVN(0_S, \sigma_m^2 \Sigma_m),$$

  where 0_S is an $S \times 1$ vector of zeros,
  σ_m^2 the between-site variance and
  Σ_m is the $S \times S$ correlation matrix, in which element (s, s') represents the correlation between sites s and s'.
- Note: the site effects are constrained to sum to zero.
- This model is stationary and assumes an isotropic covariance model in which the correlation between sites s and s' is assumed to be a function of the distance between them

  $$f(d_{ss'}, \phi) = \exp\left(-\phi d_{ss'}\right)$$

  where $\phi > 0$ describes the strength of the correlation and in this case $\kappa = 1$
- A simpler model assumes that the site-specific levels are (conditionally) independent

  $$m_s \sim \text{ i.i.d } N(0, \sigma_m^2),$$

```
 # set the spatial effects  up as spatial.exp prior


m[1:8] ~ spatial.exp(mu[], xcoords[],ycoords[],tau.m,phi1,phi2)

# and to constrain the sums to be zero
for (site in 1:8) {
mu[site]<-0
m.adj[site] <- m[site]-mean(m[1:8])
}


phi2 <- 1
phi1 ~ dunif(0.005,0.115)

tau.m ~ dgamma(1,0.01)
sigma.m <- 1/sqrt(tau.m)

sigma.m.adj <- sqrt(pow(sigma.m,2.0)*8.0/7.0)

      } # model
```

- **Stage Three, Hyperpriors:**
  - Unless there is specific information to the contrary, i.e. that a monitor with different characteristics is used at a particular site, we will assume $\sigma_{vs}^{-2} \sim Ga(a_v, b_v)$.
  - The between site precision has prior $\sigma_m^{-2} \sim Ga(a_m, b_m)$.
  - A uniform prior is used for ϕ, with the limits being based on beliefs about the relationship between correlation and distance.
  - The distance, d, at which the correlation, ρ, between two sites might be expected to fall to a particular level would be $d = -\log(\rho)/\phi$.

```
 # Set up the priors for 'edges' of the underlying  process for theta
      theta[1]~dnorm(theta[2],tau.w)
      theta[n]~dnorm(theta[n-1],tau.w)

# Set up the priors for the 'edges' of the y's
for (site in 1:8) {
y.mat[1,site] ~ dnorm(theta[1],tau.v[site])
y.mat[n,site] ~ dnorm(theta[n],tau.v[site])
      }

# Set up the priors for the 'edges' of the precisions
      tau.w.like[1] <- pow(theta[2]-theta[1],2)
      tau.w.like[n] <- pow(theta[n]-theta[n-1],2)

# Set up the likelihood calculations (because of cyclical graph) and priors
      tau.w2 <- tau.w*2
      d.w <- 1+sum(tau.w.like[])/2
      r.w <- 1 + n/2
 tau.w ~ dgamma(r.w,d.w)
      sigma.w <- 1 / sqrt(tau.w)

      # Set up the site specific observation precisions
for (site in 1:8) {
    tau.v[site] ~ dgamma(1,0.001)
   sigma.v[site] <-1/sqrt(tau.v[site])
      }
      .
```

- Data

```
list(n = 1461,
 xcoords = c(551.8, 530.2, 520, 544, 530, 506.9, 524, 525.6),
 ycoords = c(176.3, 182, 184, 174.7, 189.2, 178.6, 181.7, 164.6),
y.mat = structure(.Data = c(NA, 2.83321334405622, NA, NA, NA,
NA, NA, NA, NA, 2.89037175789616, NA, NA, NA, NA, NA, NA, NA,
2.77258872223978, NA, NA, NA, NA, NA, NA, NA, 2.77258872223978,
...
2.83321334405622, 2.77258872223978, 2.56494935746154, 2.63905732961526,
2.19722457733622, 2.484906649788, 2.30258509299405,
2.30258509299405, 2.63905732961526, 2.484906649788, 2.30258509299405,
2.56494935746154), .Dim = c(1461,8)))
```

- Note: WinBUGS reads data into an array by filling the right-most index first, whereas the R fills the left-most index first. Therefore in R, before the data is exported the transpose function was used, `y.mat = t(y.mat)` before using `dput(y.mat, filename)`.
- If the data was in a three dimensional array `aperm` can be used, e.g. `y.array = aperm(y.array,c(1,3,2)` to achieve the same result.

- Initial values

```
list(tau.w = 1, tau.v = c(1, 1, 1, 1, 1, 1, 1, 1),
tau.m = 1, m = c(0, 0, 0, 0, 0, 0, 0, 0),
phi1 = 0.07,
y.mat = structure(.Data = c(3, NA, 3, 3, 3, 3, 3,
3, 3, NA, 3, 3, 3, 3, 3, 3, 3, NA, 3, 3, 3, 3, 3, 3, 3,
...
NA, NA, NA, NA, NA, NA, NA), .Dim = c( 1461,8)),
theta = c(2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5,
...
2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5,
2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5,
2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5))
```

Exercises

Using WinBUGS

1. Compare the results from the model using the joint spatial model (using `spatial.exp`) with one that assumes that the site effects are conditionally independent.
2. Replace the specifying the full conditional approach to implementing the temporal (random walk) part of the model with the `car.normal` approach developed in the single site model.
3. Implement an AR(1) process in place of the RW(1) process, i.e. $\theta_t \sim N(\rho\theta_{t-1}, \sigma_w^2)$. Note that for the RW(1) process, $\rho = 1$. Estimate the value of ρ and consider how appropriate the non-stationary model seems in this case.

If you can do 2. or 3. I would be very grateful!

THANK YOU!