Spatio-temporal modelling of exposures

Lecture 20

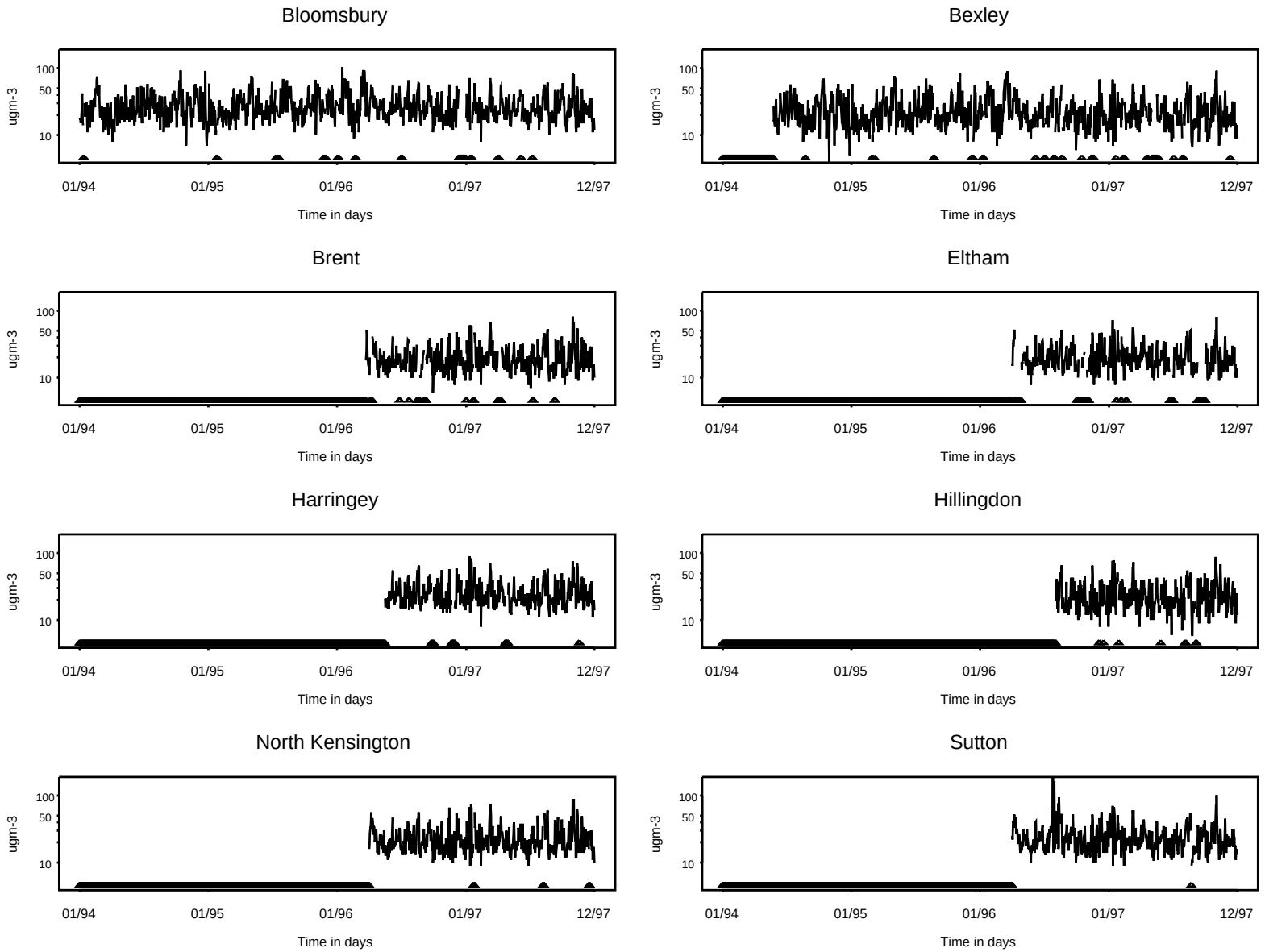
Background

- Daily measurements often available for different pollutants from a number of sites
- May be subject to measurement error
- Contain missing values
  - Pollutants not measured at all sites
  - Monitor being moved by design, e.g. six-day monitoring schedule
  - Unreliable or faulty monitors

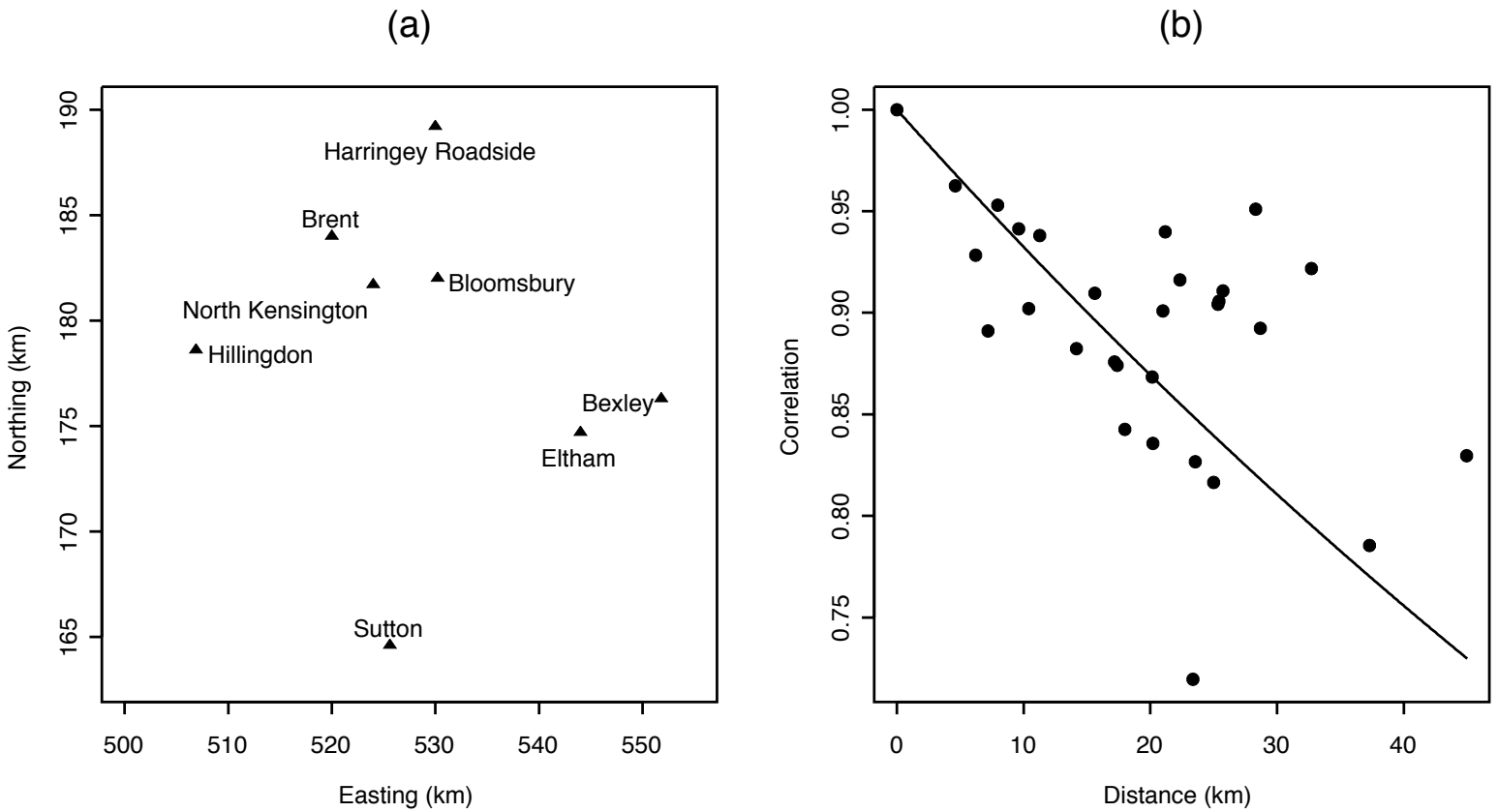
Data

- Eight sites within London, 1997-94
- PM_{10}, SO_2, NO and CO.
- All pollutants only measured at only 4 sites.
- Periods of operation between 1 and 4 years.
- Percentage of missing values as great as 37%.

Time series plots of (logged) values of PM_{10}



Locations of monitoring sites and correlations with distance



Model framework

- Bayesian hierarchical model.
- Pollutants modelled as a function of the true underlying level with measurement error.
- Can incorporate covariate information, e.g. temperature.
- Underlying level is a function of the previous day's level.
- Missing values treated as unknown parameters within the Bayesian framework and can be estimated.

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 $N(0, \sigma_v^2)$

- **Stage Two, Temporal Model:**
Autoregressive first order model

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

w_t i.i.d. as $N(0, \sigma_w^2)$.

- **Stage Three, Hyperprior:**
Normal prior $N(c, C)$ for β_1, where c is a $q_1 \times 1$ vector and C a $q_1 \times q_1$ variance-covariance matrix.
$\sigma_v^{-2} \sim Ga(a_v, b_v)$ and $\sigma_w^{-2} \sim Ga(a_w, b_w)$.

Posterior distribution

The posterior distribution is given by

$$p(\theta, \beta_1, \sigma_v^2, \sigma_w^2|y) = p(y)^{-1} \left\{ \prod_{t=1}^{T} p(y_t|\theta_t, \beta_1, \sigma_v^2) \right\} \times$$
$$\left\{ \prod_{t=2}^{T} p(\theta_t|\theta_{t-1}, \sigma_w^2) \right\} \times$$
$$p(\theta_1)p(\beta_1)p(\sigma_v^2)p(\sigma_w^2)$$

- Samples may be generated in a straightforward fashion using Markov chain Monte Carlo (using `WinBUGS`)
- Dealing with the cyclical graph that arises at stage two, requires some of the conditional distributions to be explicitly specified

- Missing values are treated as parameters and the posterior obtained over these values and the model parameters. Samples can be generated from the distribution of missing values

$$p(y_m|y_o) = \int p(y_m|\lambda)p(\lambda|y_o)d\lambda$$

where $\lambda = (\theta, \beta_1, \sigma_v^2, \sigma_w^2)'$

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:**

$$\theta_t = \rho\theta_{t-1} + w_t, w_t \text{i.i.d.} as N(0, \sigma_w^2)$$

- 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)
    ```

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

- 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`

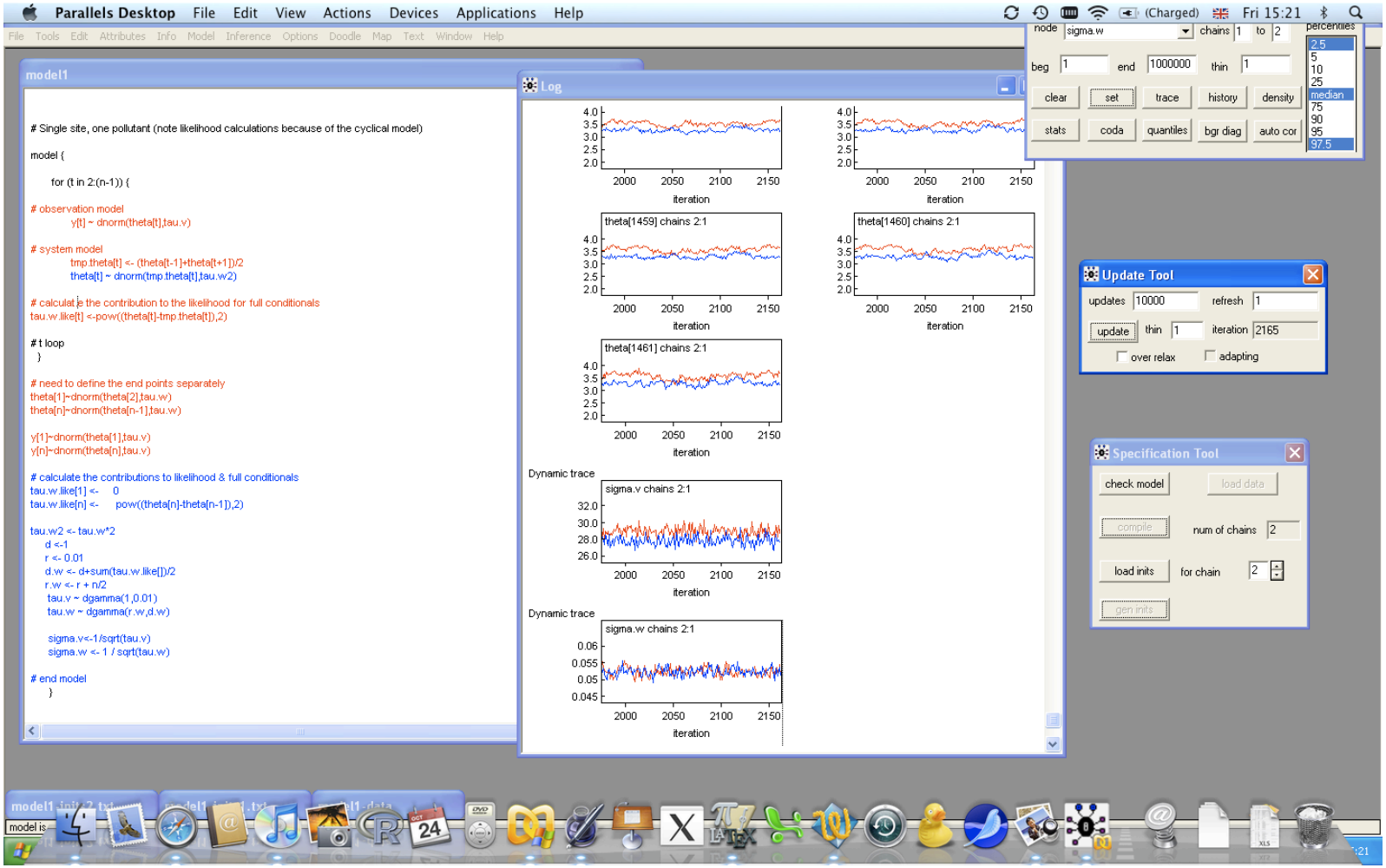
```
list(tau.v = 1, tau.w = 1, theta = c(3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
...,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3),

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

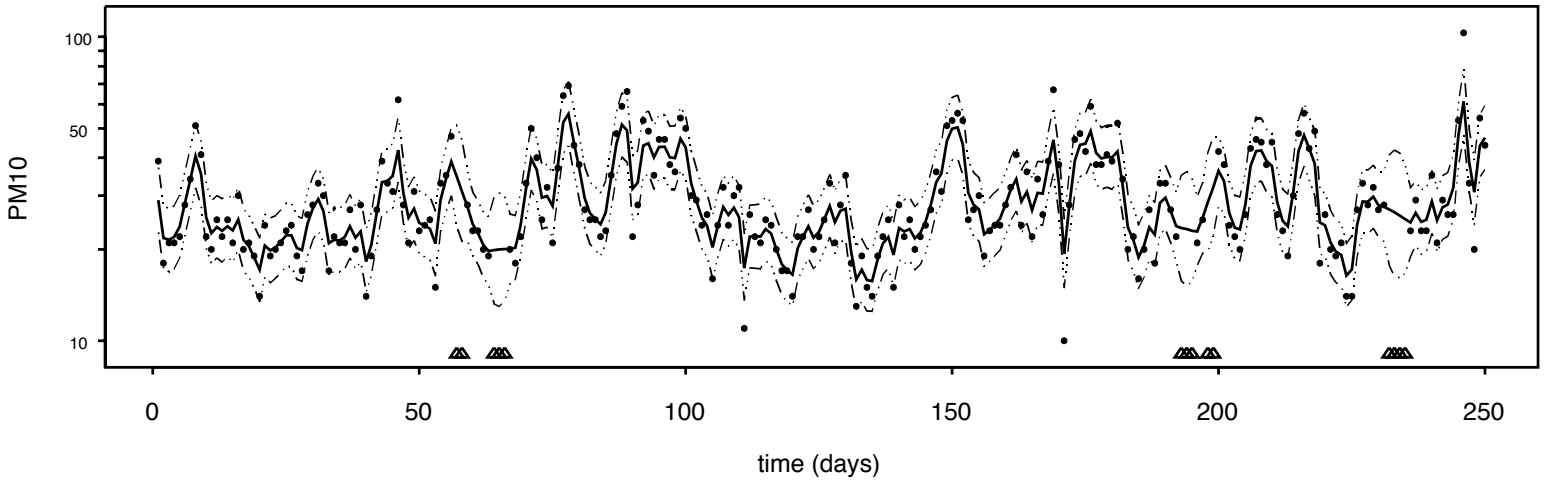
- `theta` - if you are interested in keeping all of them (there are 1461 of them, one for each day)
- `theta[i]` - if you want to keep a single one of them
- `theta[i:j]` or `theta[c(3,56,987)]` - if you want to keep a selection
- `sigma.v` - the variance of the random error from the first level of the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

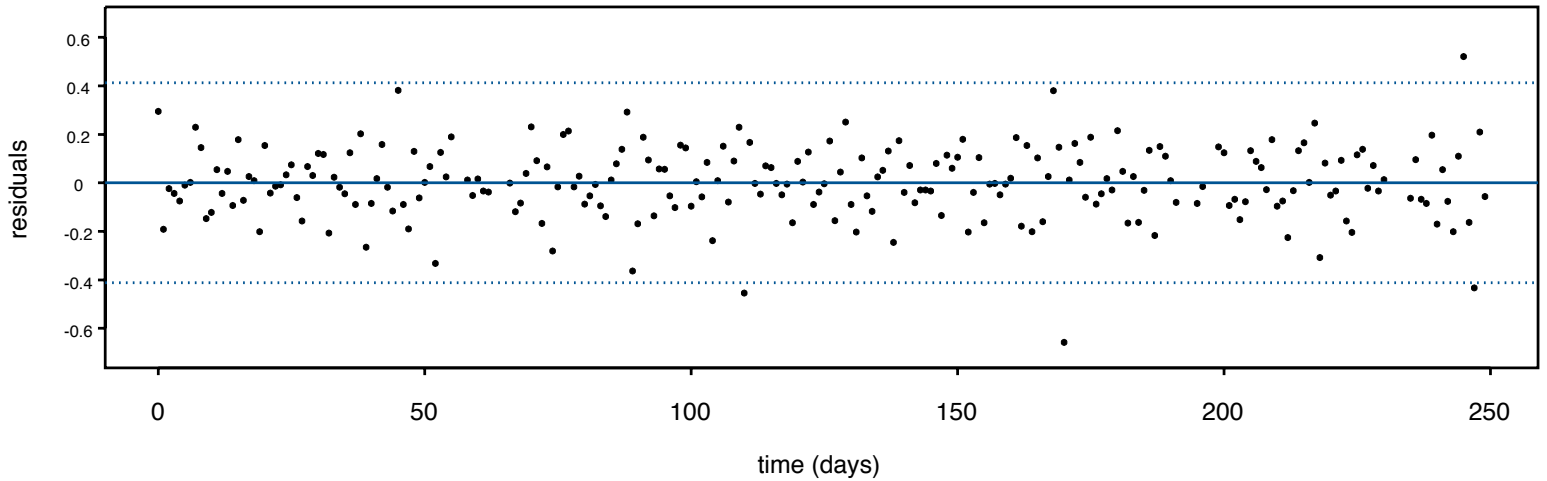

Parallels Desktop File Edit View Actions Devices Applications Help
(Charged) Fri 15:21
model1
# Single site, one pollutant (note likelihood calculations because of the cyclical model)
model {
for (t in 2:(n-1)) {
# observation model
y[t] ~ dnorm(theta[t],tau.v)
# system model
tmp.theta[t] <- (theta[t-1]+theta[t+1])/2
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
}
Log
theta[1459] chains 2:1
theta[1460] chains 2:1
theta[1461] chains 2:1
iteration
Dynamic trace
sigma.v chains 2:1
sigma.w chains 2:1
node sigma.w chains 1 to 2
beg 1 end 1000000 thin 1
clear set trace history density
stats coda quantiles bgr diag auto cor
percentiles
2.5 5 10 25 median 75 90 95 97.5
Update Tool
updates 10000 refresh 1
update thin 1 iteration 2165
over relax adapting
Specification Tool
check model load data
compile num of chains 2
load inits for chain 2
gen inits
model is

Time series of 250 days of observed and estimated levels (together with their differences) of PM_{10} at Bloomsbury

(a)



(b)



Adding a spatial component

- Assume that $\mathbf{m} = (\mathbf{m_1}, ..., \mathbf{m_n})$ arise from a zero mean multivariate normal distribution with variances $\text{var}(m_i) = \sigma_u^2$ and correlations $\text{corr}(m_i, m_j) = \exp(-\phi d_{ij}) = \rho^{d_{ij}}$ 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 $\mathbf{m}$.
- more generally the correlations can be modeled as $\text{corr}(m_i, m_j) = \exp(-(\phi d_{ij})^\kappa)$.

WinBmGS *representation*

The above model with

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

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

```
m[1:N] ~ 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).

Multiple monitoring sites

- 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.

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

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

with w_t i.i.d. as $N(0, \sigma^2_w)$.

- **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'.
- 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

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

  where $\phi > 0$ describes the strength of the correlation
- A simpler model assumes that the site-specific levels are (conditionally) independent

- **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 $d = -\log(\rho)/\phi$.

Multiple monitoring sites

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

$$Y_{st} = m_s + \theta_t + v_{st}$$

- Dropping the covariate terms for clarity of explanation.
- **Stage Two (a), Temporal Model:**

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

- Again, we consider a RW(1) process, i.e. $\rho = 1$ in the AR(1) process.

- **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'.
- We use the `spatial.exp` distribution in WinBUGS. In the code, ϕ and κ in $\text{corr}(U_i, U_j) = \exp(-(\phi d_{ij})^\kappa)$ are labelled $\phi = \phi_1$ and $\kappa = \phi_2$.

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

- Note: the site effects are constrained to sum to zero.

```
for (site in 1:8) {
mu[site]<-0
m.adj[site] <- m[site]-mean(m[1:8])
}
```

```
 # 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
```

- 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)))
```

- 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))
```

Estimating levels at unmeasured locations

- Based on the posterior estimates of the site effects, m_s and the variance-covariance matrix $\sigma_m^2 \Sigma_m$, it is possible to estimate the site effects, and thus pollution levels, at locations where there is no monitoring site.
- For a site at a new location, m_{S+1}, $(m_1, ..., m_S, m_{S+1})$ follows a multivariate normal distribution with zero mean and $(S+1) \times (S+1)$ variance-covariance matrix.
- Letting $m = (m_1, .., m_S)'$, the conditional distribution of $m_{S+1}|m$ is, normal with mean and variance given by

$$E[m_{S+1}|m] = \sigma_m^{-2} \Omega' \Sigma_m^{-1} m,$$

$$var(m_{S+1}|m) = \sigma_m^2 (1 - \Omega' \Sigma_m^{-1} \Omega),$$

- For exploratory purposes, the posterior medians may be substituted into these expressions (although this will ignore the inherent uncertainty in the estimates).

Site effects

	Median	2.5%	97.5%
Bexley	-0.0696	-0.0785	-0.0607
Bloomsbury	0.1341	0.1257	0.1426
Brent	-0.1210	-0.1294	-0.1125
Eltham	-0.1105	-0.1205	-0.1005
Harringey	0.1098	0.0999	0.1195
Hillingdon	0.0132	-0.0032	0.0300
North Kensington	0.0030	-0.0031	0.0090
Sutton	0.0410	0.0250	0.0572
σ_m	0.1019	0.0668	0.1794
ϕ	0.05675	0.02158	0.09778

spatial.pred and spatial.unipred

- Spatial interpolation or prediction at arbitrary locations can be carried out using the `spatial.pred` or `spatial.unipred` functions, in conjunction with fitting the `spatial.exp` model to a set of observed data.
- `spatial.pred` carries out joint or simultaneous prediction at a set of target locations
- `spatial.unipred` carries out single site prediction.
- The difference is that the single site prediction yields marginal prediction intervals (i.e. ignoring correlation between prediction locations) whereas joint prediction yields simultaneous prediction intervals for the set of target locations (which will tend to be narrower than the marginal prediction intervals).
- The predicted means should be the same under joint or single site prediction.
- The disadvantage of joint prediction is that it is very slow
  - computational time is of order P3, where P is the number of prediction sites

- The syntax for these predictive distributions is:
- Joint prediction:

```
m[1:P] ~ spatial.pred(mu.m[], x.m[], y.m[], S[])
```

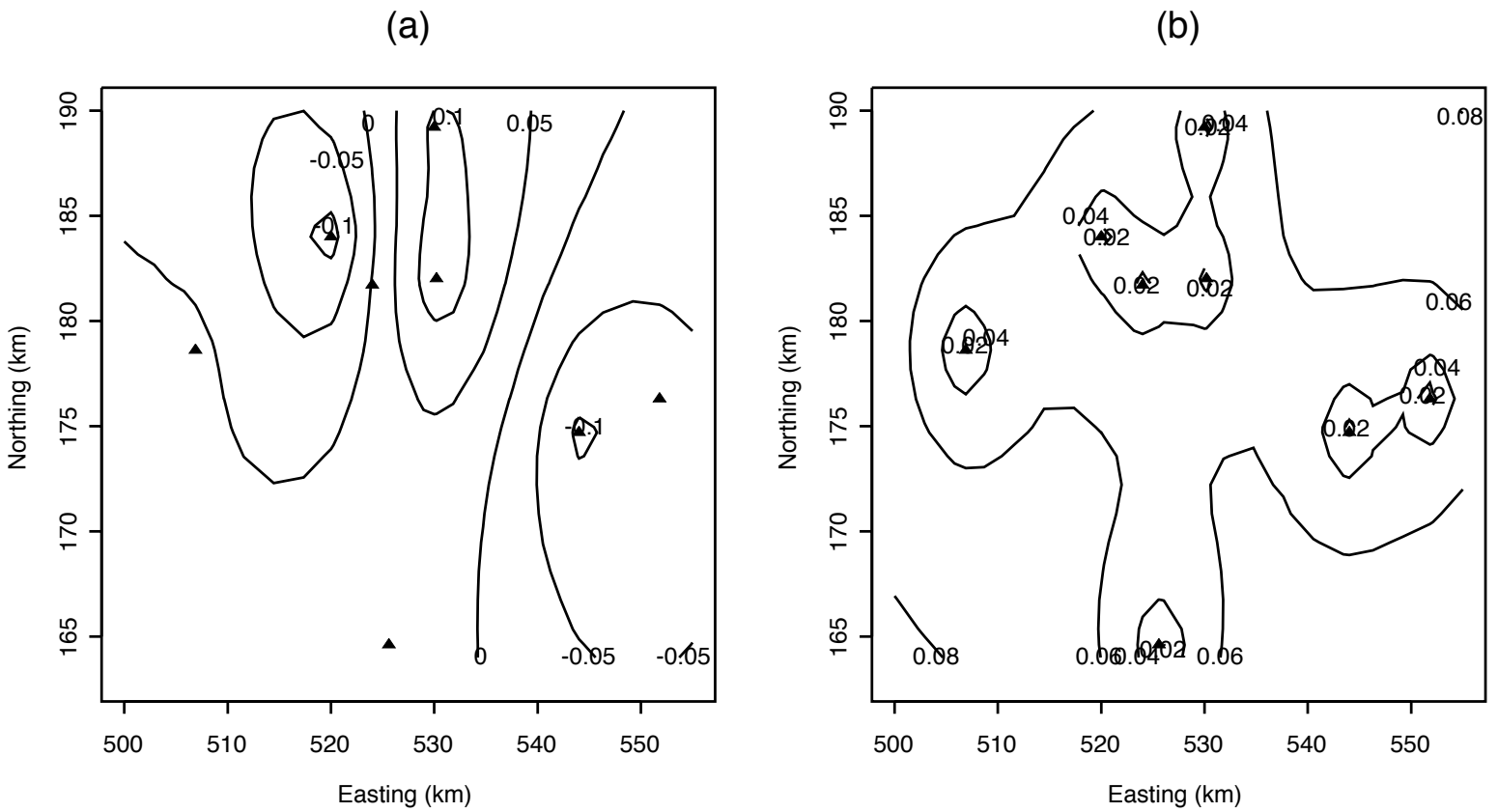
- Single site prediction:

```
for(j in 1:P) {
m[j] ~ spatial.unipred(mu.m[j], x.m[j], y.m[j], S[
                    }
```

where:

- `P` : Scalar giving the number of prediction locations
- `mu.m[]` : vector of length P (or scalar for single site version) specifying the mean for each prediction location (this should be specified in the same way as the mean for the observed data `S`).
- `x.m[]` and `y.m[]` : Vectors of length P (or scalars for single site version) giving the x and y coordinates of the location of each prediction point
- S : The vector of observations to which the `spatial.exp` model has been fitted.

Contour plot of site effects based on a 20x20 grid of locations without a pollution monitor with corresponding standard deviations



Health analysis

- PM_{10} and respiratory mortality (ICD 460-519) in London, 1994-97.
- Assess the effects of using modelled levels of pollutant on relative risks.
- Base model contains terms for trend, trend^2, year, month, year $\times$ month interaction, day of week, 12, 6, 4 and 2 monthly cycles and temperature (same day, lag 1, lag2).

Relative risks (and 95% CIs) associated with increase of 10$\mu g m^{-3}$ in PM_{10} (lag 1)

- Observed PM_{10} with missing values excluded
  - RR = 1.0116 (1.0046 - 1.0186)
- Modelled PM_{10} with missing values excluded
  - RR = 1.0166 (1.0064 - 1.0269)
- Modelled PM_{10} with estimated missing values
  - RR = 1.0182 (1.0084 - 1.0280)

using spatial model

- Modelled PM_{10} with missing values excluded
  - RR = 1.0134 (1.0066 - 1.0203)
- Modelled PM_{10} with estimated missing values
  - RR = 1.0128 (1.0062 - 1.0195)