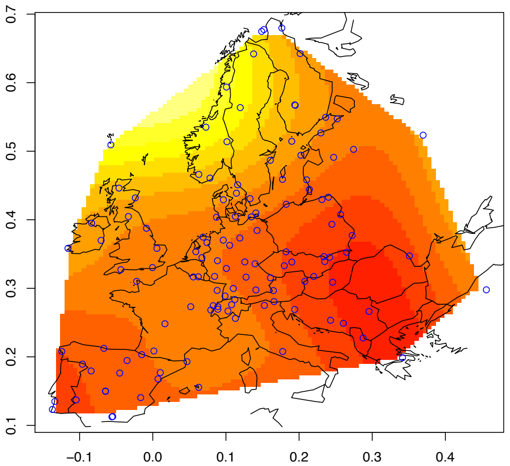
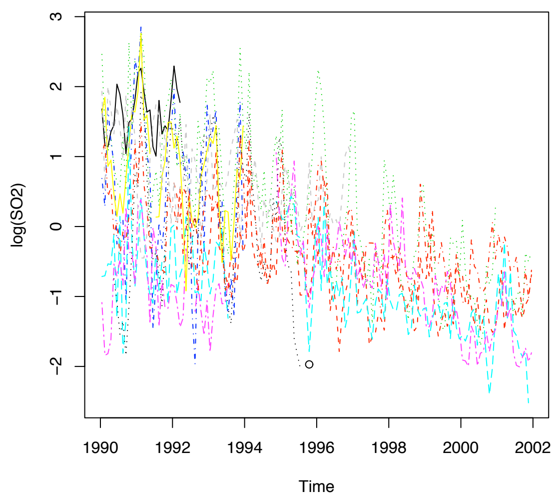




Regression models in space and time

Adrian Bowman

University of Glasgow | School of Mathematics & Statistics

with

Alastair Rushworth, David O'Donnell, Marco Giannitrapani,
Marian Scott, Rognvald Smith,
Mark Hallard, Mark Brewer and Simon Langan

Regression models in space and time

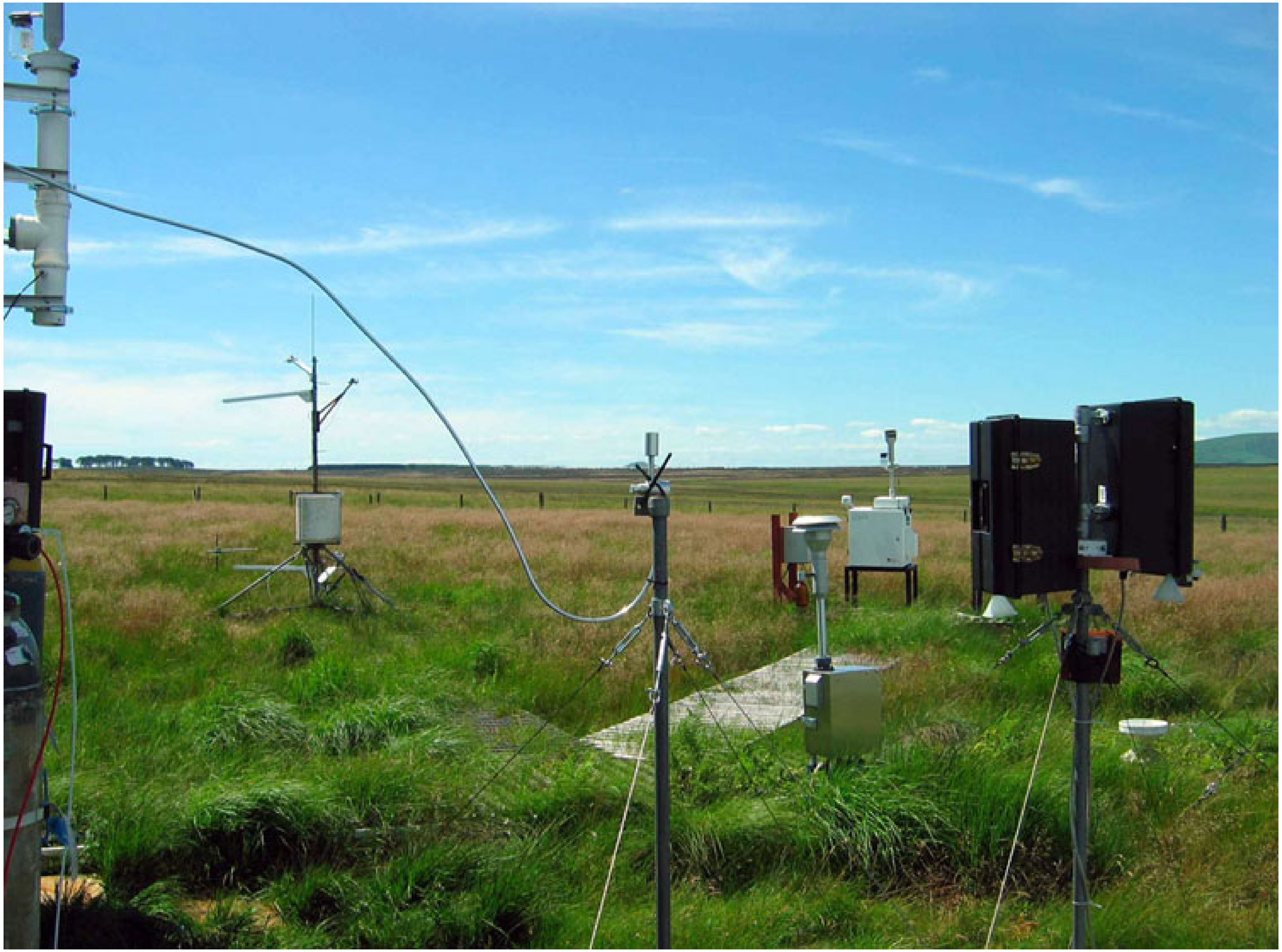
Blowing in the wind

air pollution over Europe

Going with the flow

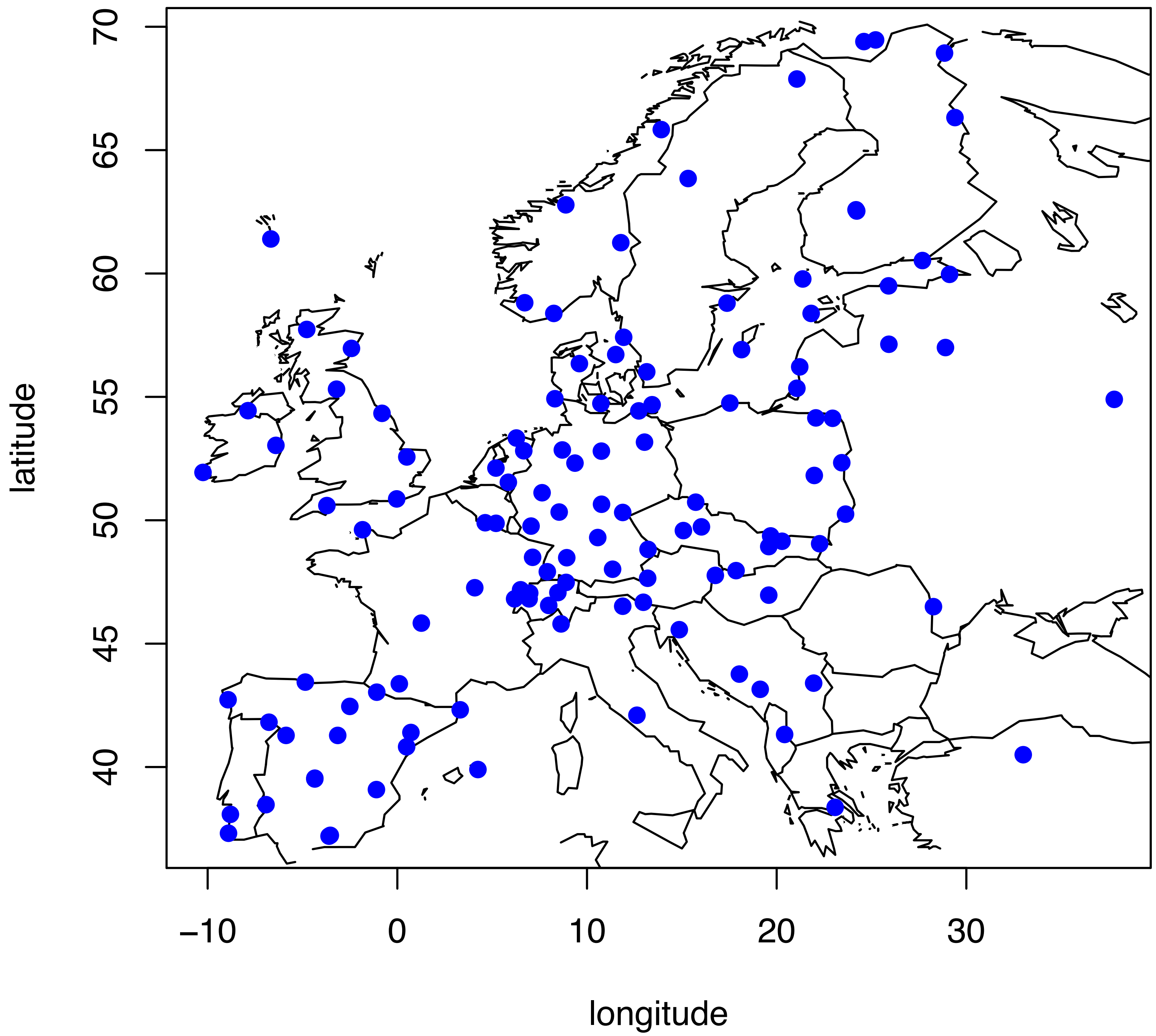
nitrates in the River Tweed

SO_2 pollution

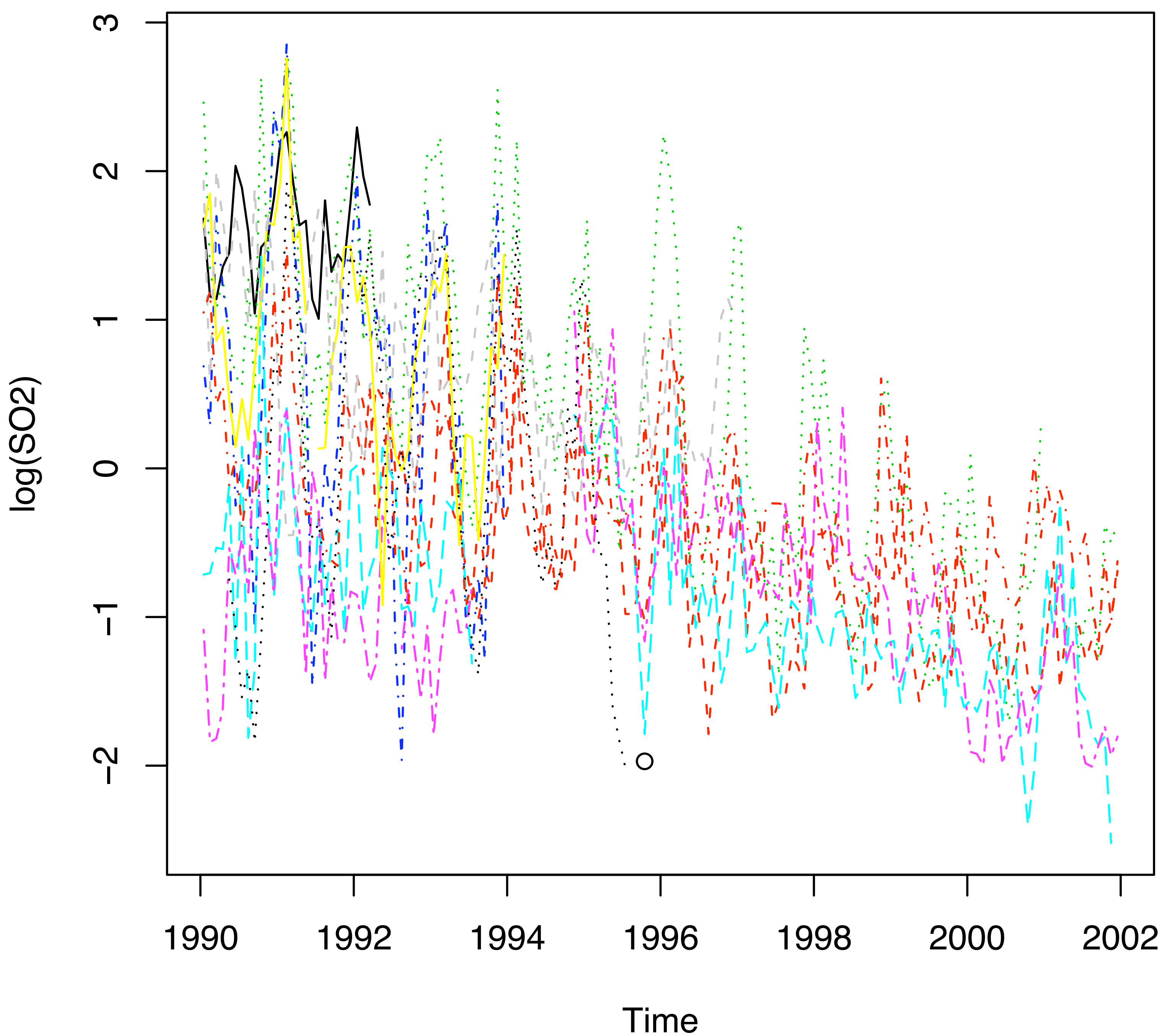


Marco Giannitrapani, Marian Scott, Rognvald Smith

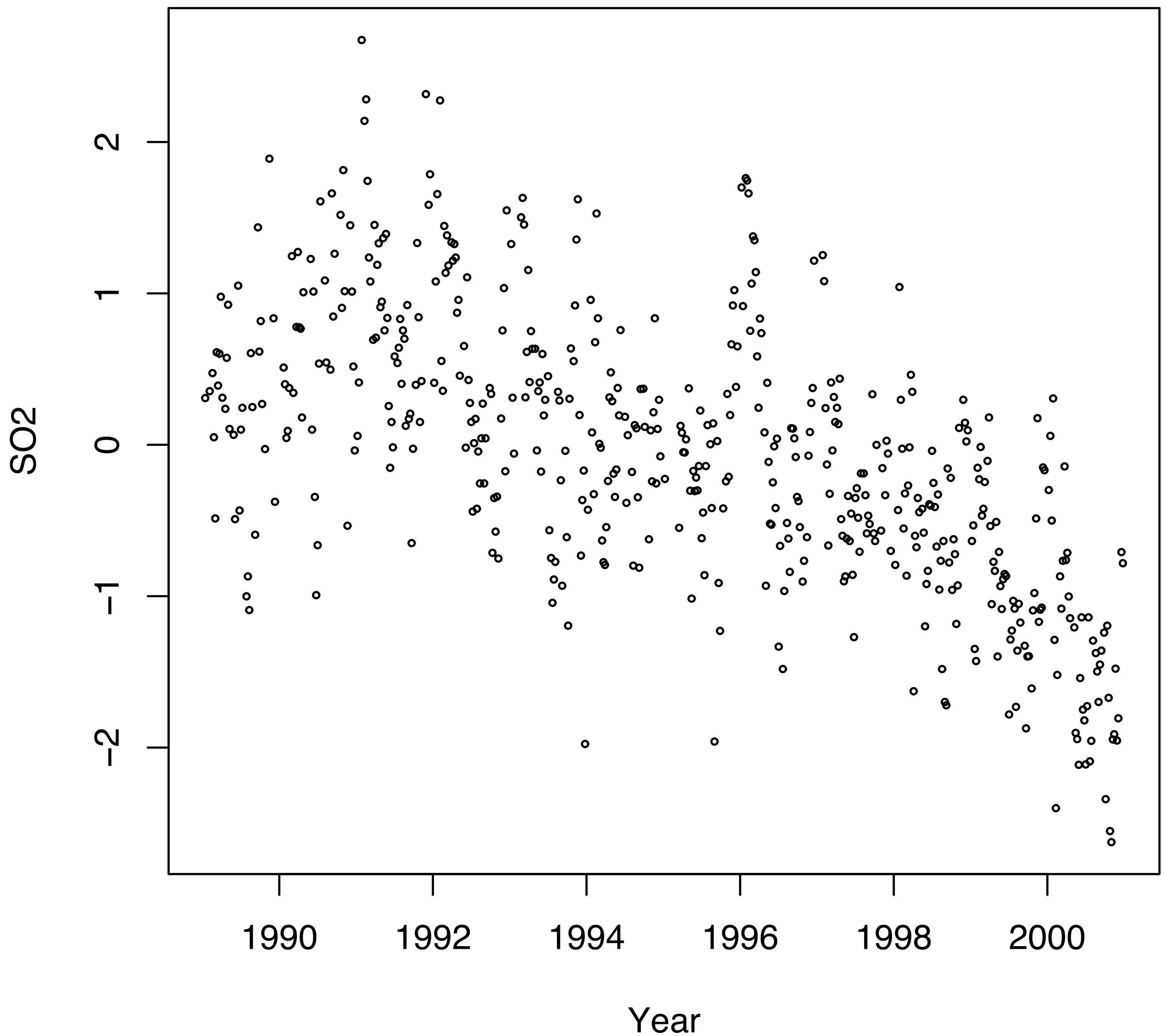
SO_2 pollution



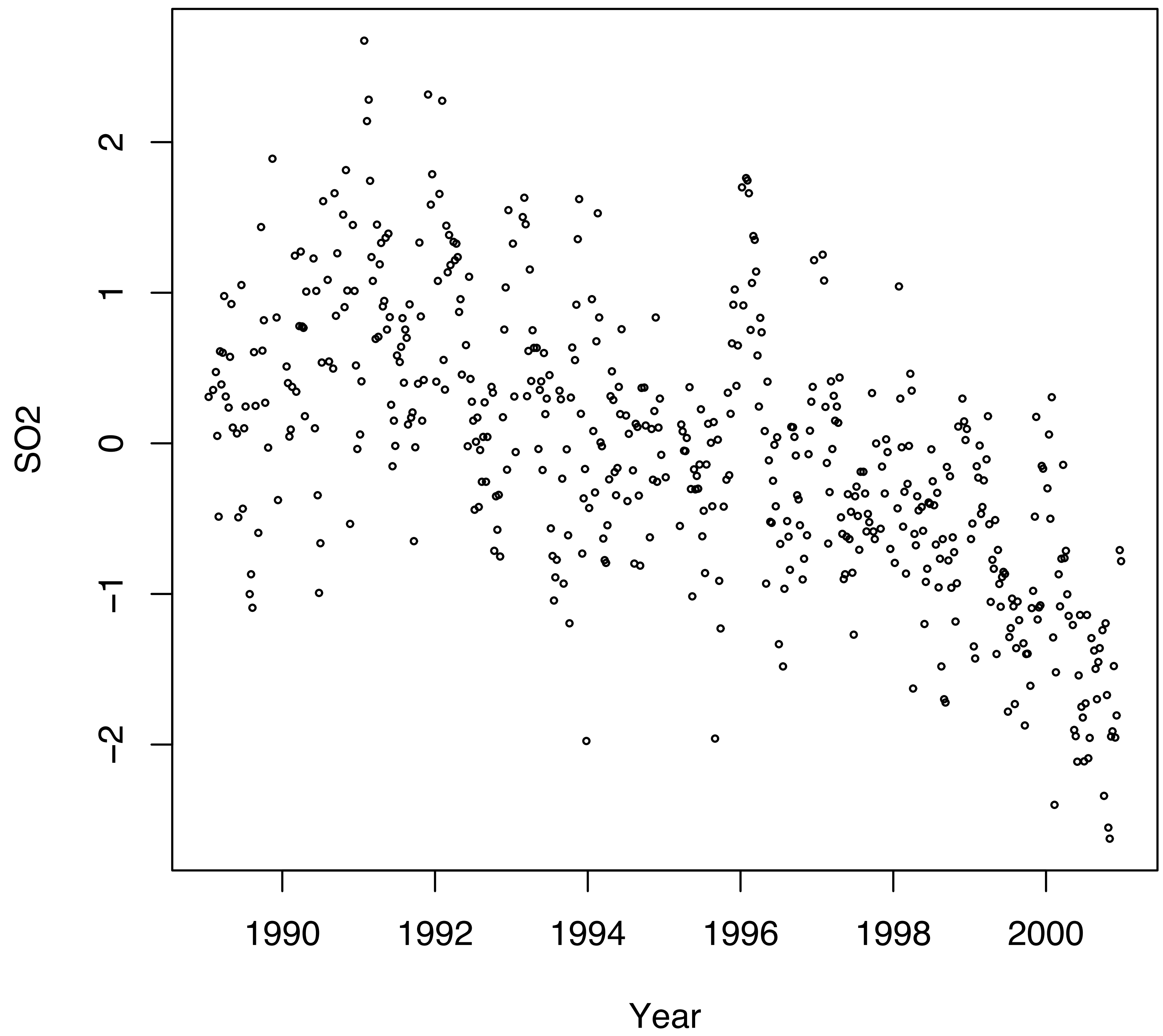
SO_2 pollution



SO_2 pollution



SO_2 pollution



$$\ln(SO_2) = \mu + m_1(\text{year}) + m_2(\text{week}) + \varepsilon, \varepsilon \sim AR(1)$$

Smoothing

Model:

$$y_i = m(x_i) + \varepsilon_i$$

$\hat{m}(x)$ is defined by minimising the weighted least squares

$$\sum_{i=1}^{n} \{y_i - \alpha - \beta(x_i - x)\}^2 w(x_i - x; h)$$

over α and β and setting $\hat{m}(x) = \hat{\alpha}$.
A set of fitted values can then be represented as

$$\hat{m} = Sy$$

Approximate degrees of freedom are defined by

$$\nu = \mathrm{tr}\,\{S\}$$

Additive models

Hastie & Tibshirani (1990), Wood (2006)

$$y = \mu + m_1(\text{year}) + m_2(\text{week}) + \varepsilon$$

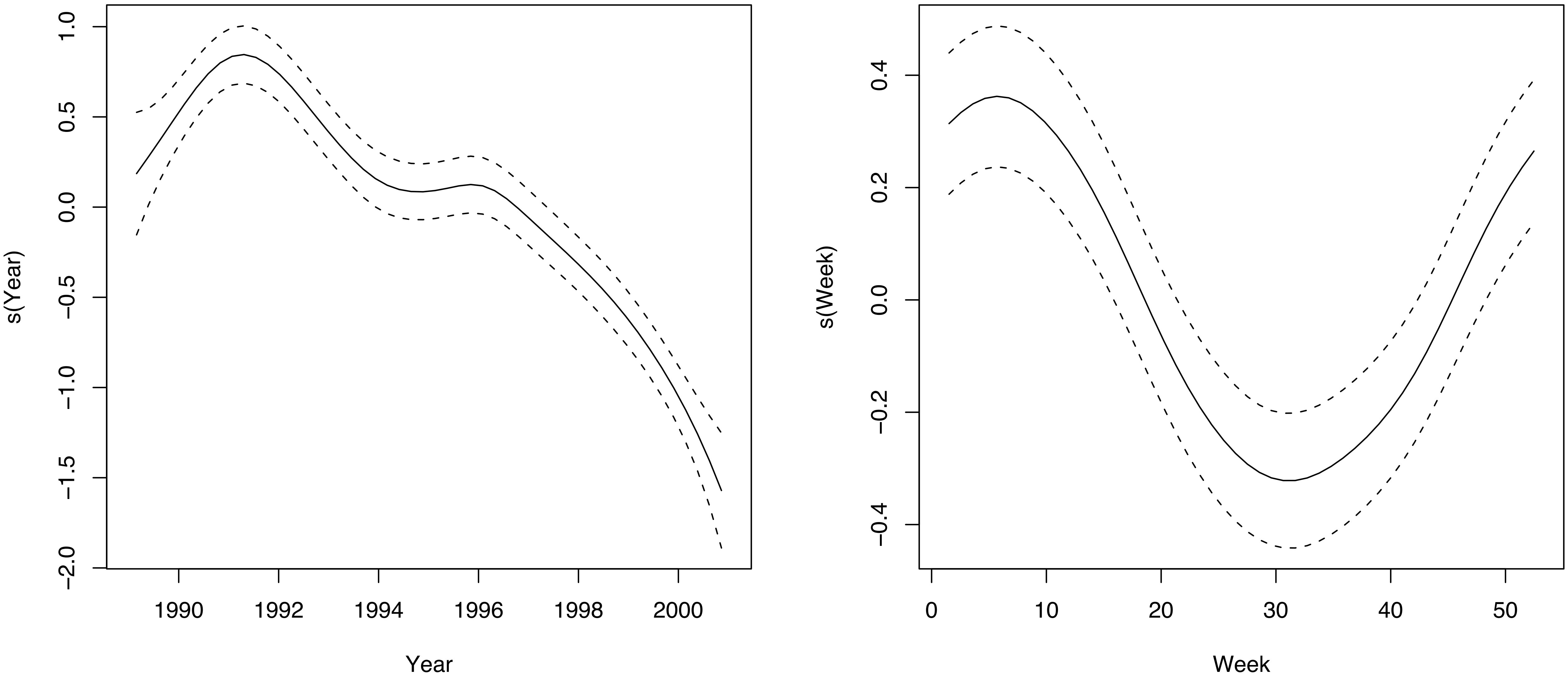
Backfitting algorithm

$$\hat{m}_1^{(r+1)} = S_1\left(y - \hat{\mu}\mathbf{1} - \hat{m}_2^{(r)}\right)$$

$$\hat{m}_2^{(r+1)} = S_2\left(y - \hat{\mu}\mathbf{1} - \hat{m}_1^{(r+1)}\right)$$

SO_2 pollution

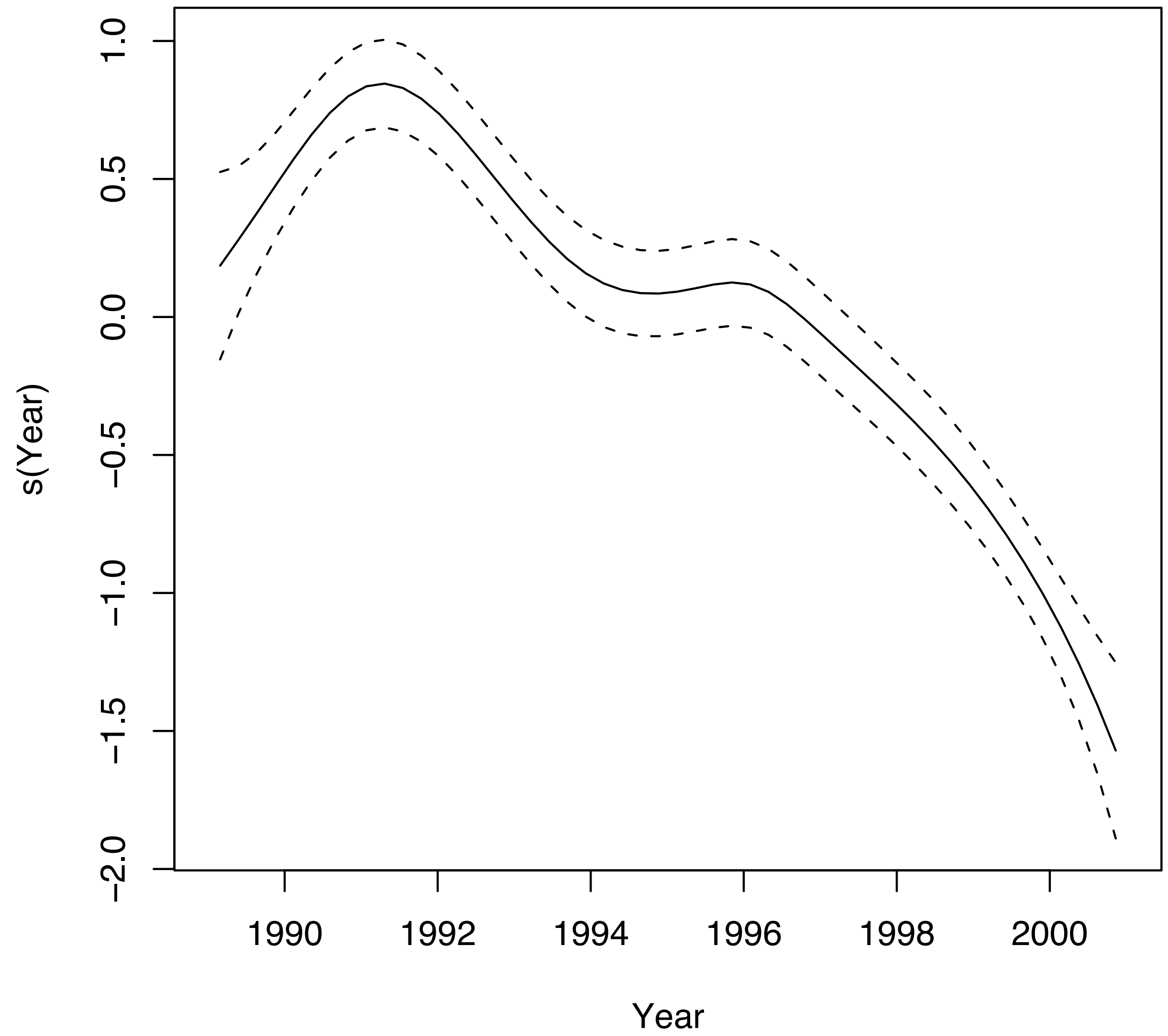
$$\ln(SO_2) = \mu + m_1(\text{year}) + m_2(\text{week}) + \varepsilon, \varepsilon \sim AR(1)$$

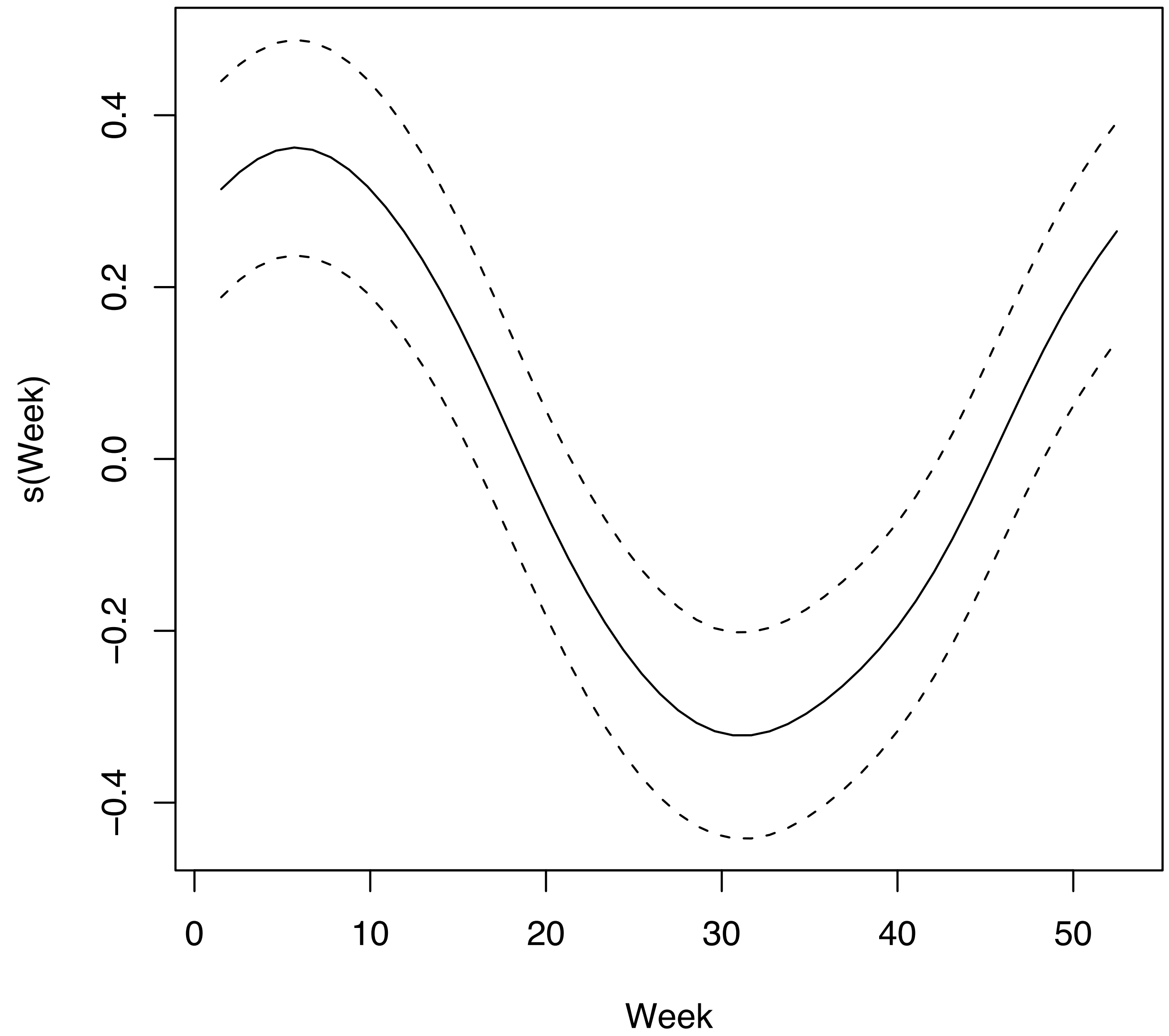


$$\text{s.e.}\{\hat{m}(x)\} = \sqrt{\text{var}\{s^\top y\}} = \sqrt{(s^\top V s)}\,\sigma$$

SO_2 pollution - adding meteorology

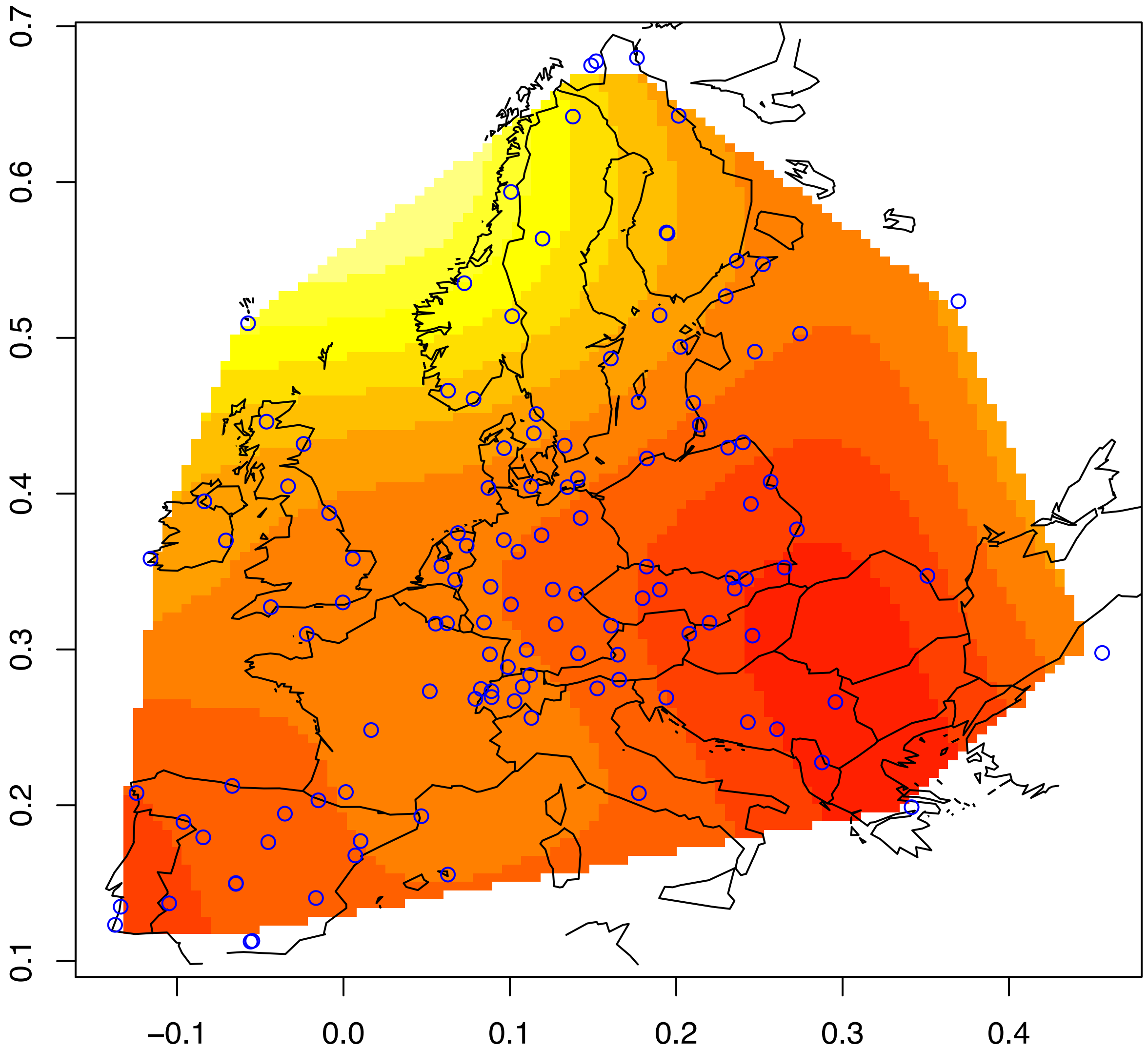
$$
\begin{aligned}
\textit{Model } 1: \quad \ln(SO_2) &= \mu + m_y(\text{year}) + m_w(\text{week}) + \varepsilon \\
\textit{Model } 2: \quad \ln(SO_2) &= \mu + m_y(\text{year}) + m_w(\text{week}) + m_r(\text{rain}) \\
&\quad + m_t(\text{temp}) + m_h(\text{humidity}) + m_a(\text{air}) + \varepsilon.
\end{aligned}
$$

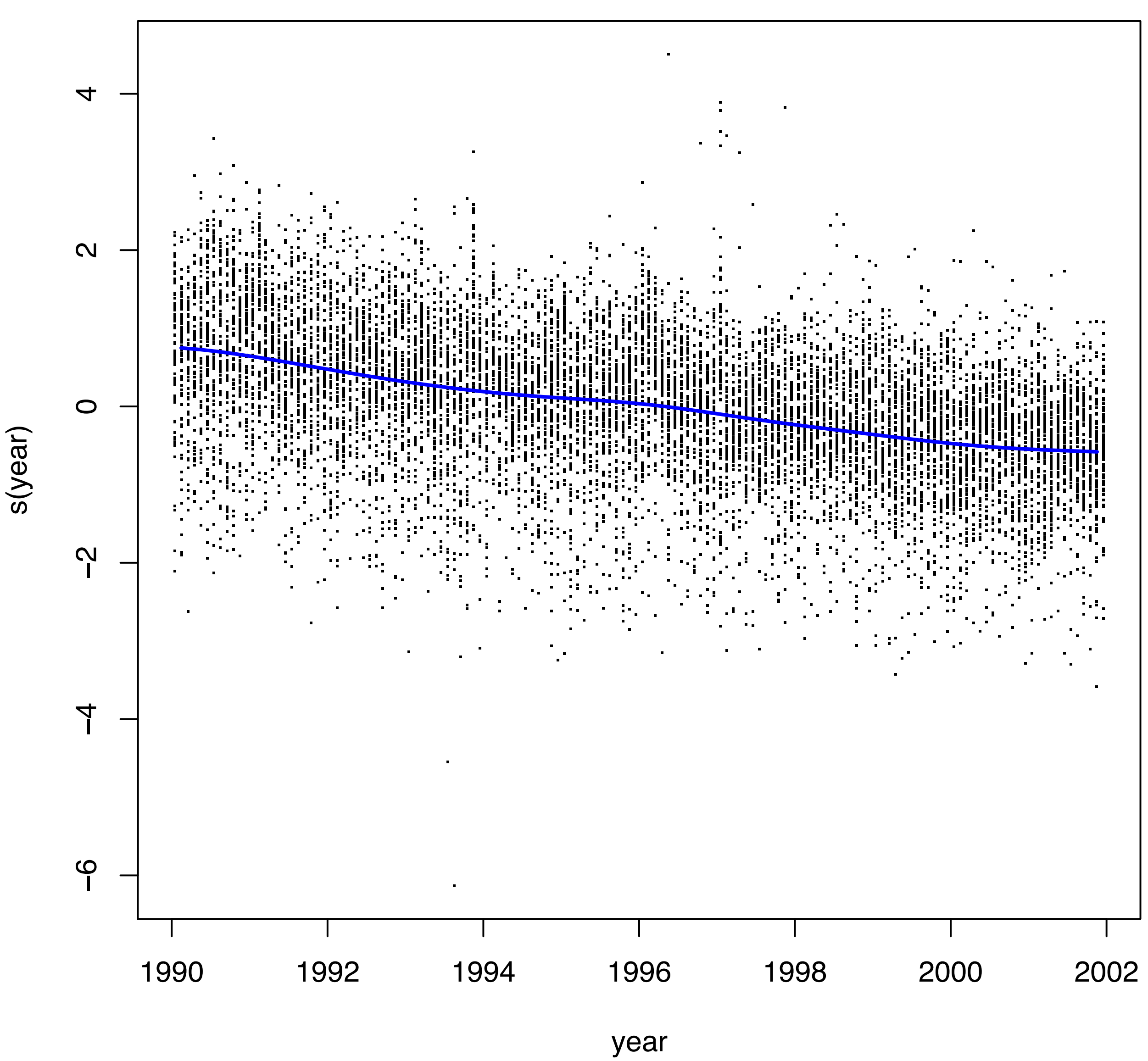




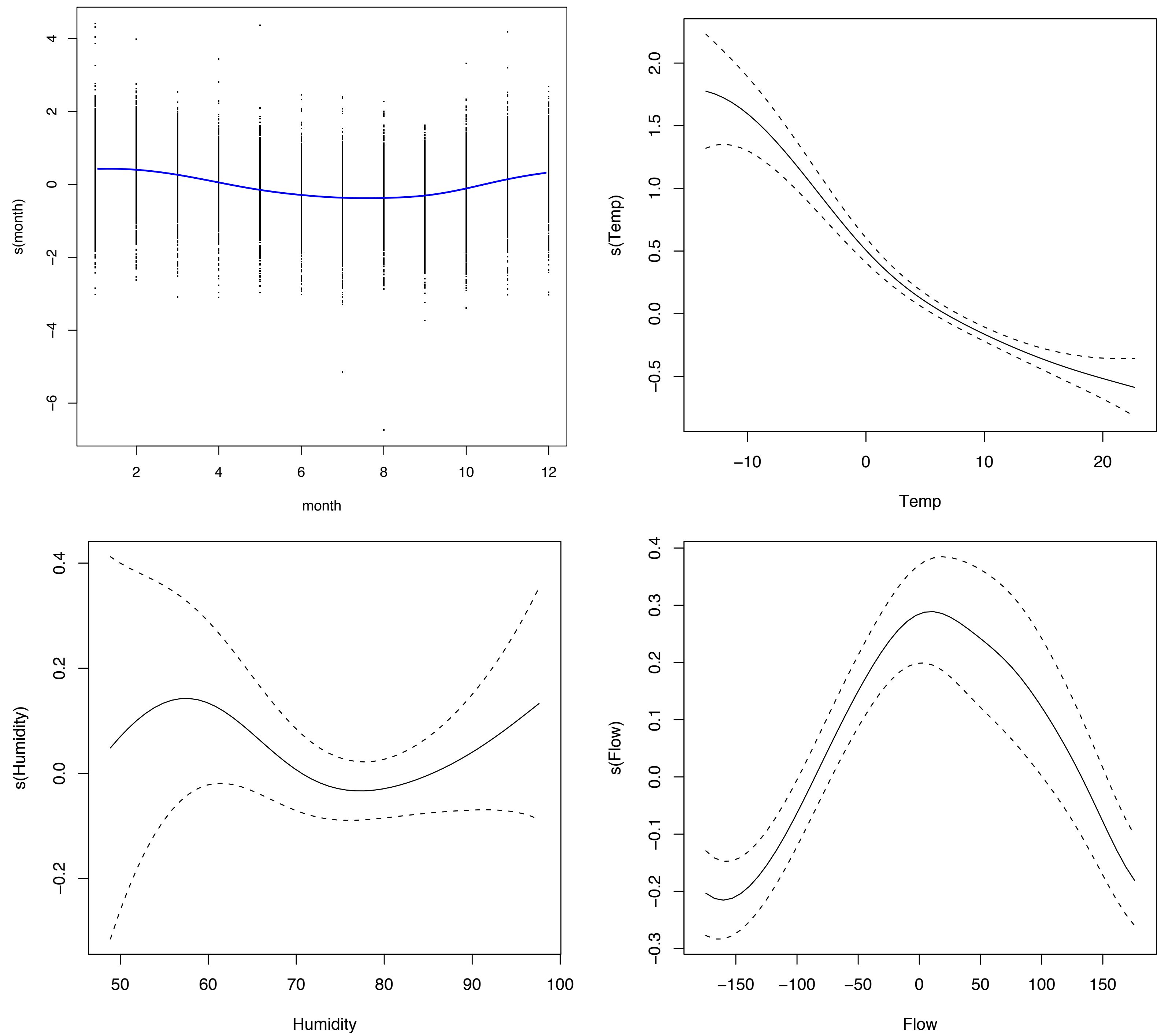
SO_2 pollution - adding meteorology

$$\begin{aligned} \textit{Model } 1: \quad \ln(SO_2) &= \mu + m_y(\text{year}) + m_w(\text{week}) + \varepsilon \\ \textit{Model } 2: \quad \ln(SO_2) &= \mu + m_y(\text{year}) + m_w(\text{week}) + m_r(\text{rain}) \\ &\quad + m_t(\text{temp}) + m_h(\text{humidity}) + m_a(\text{air}) + \varepsilon. \end{aligned}$$

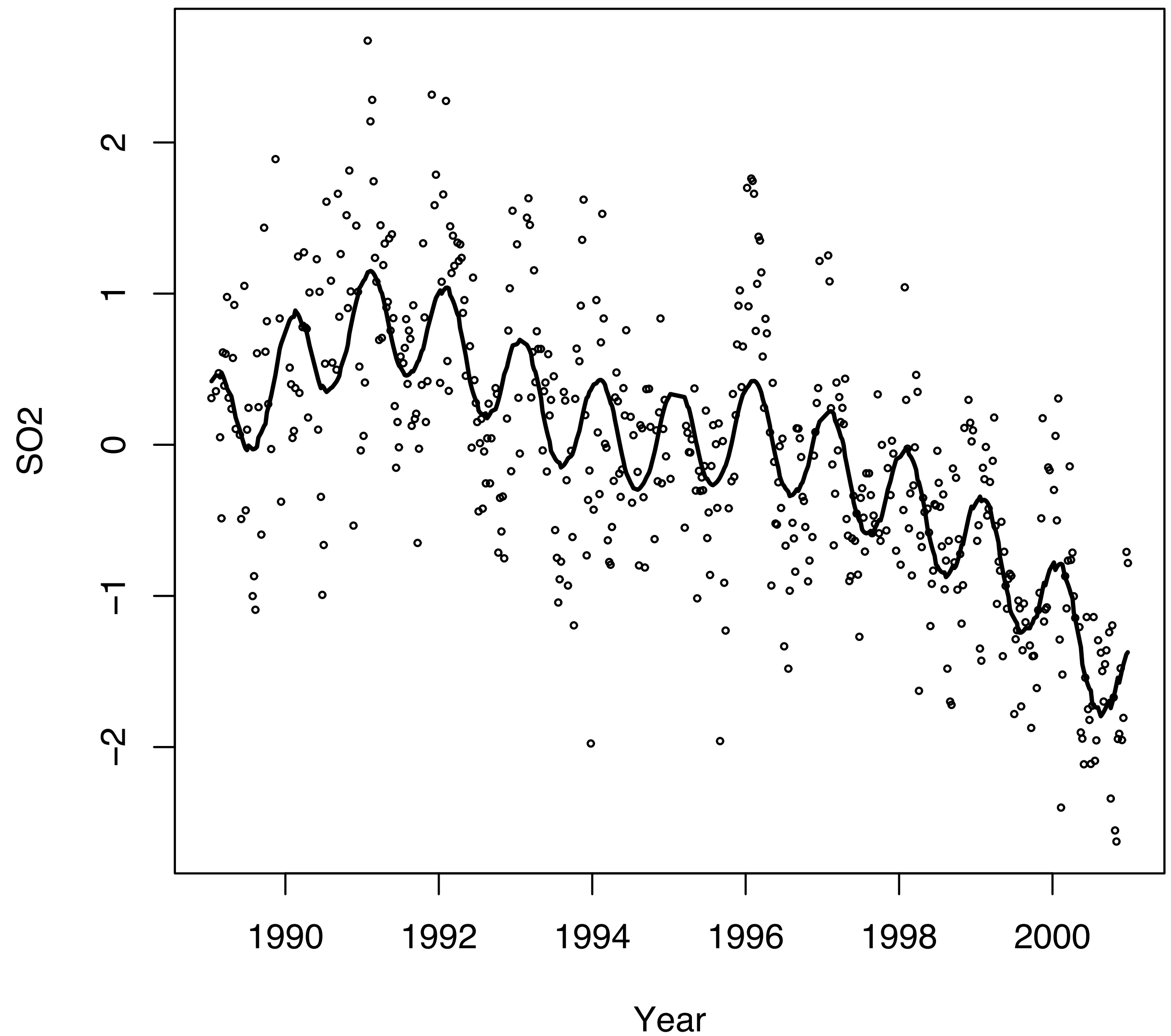


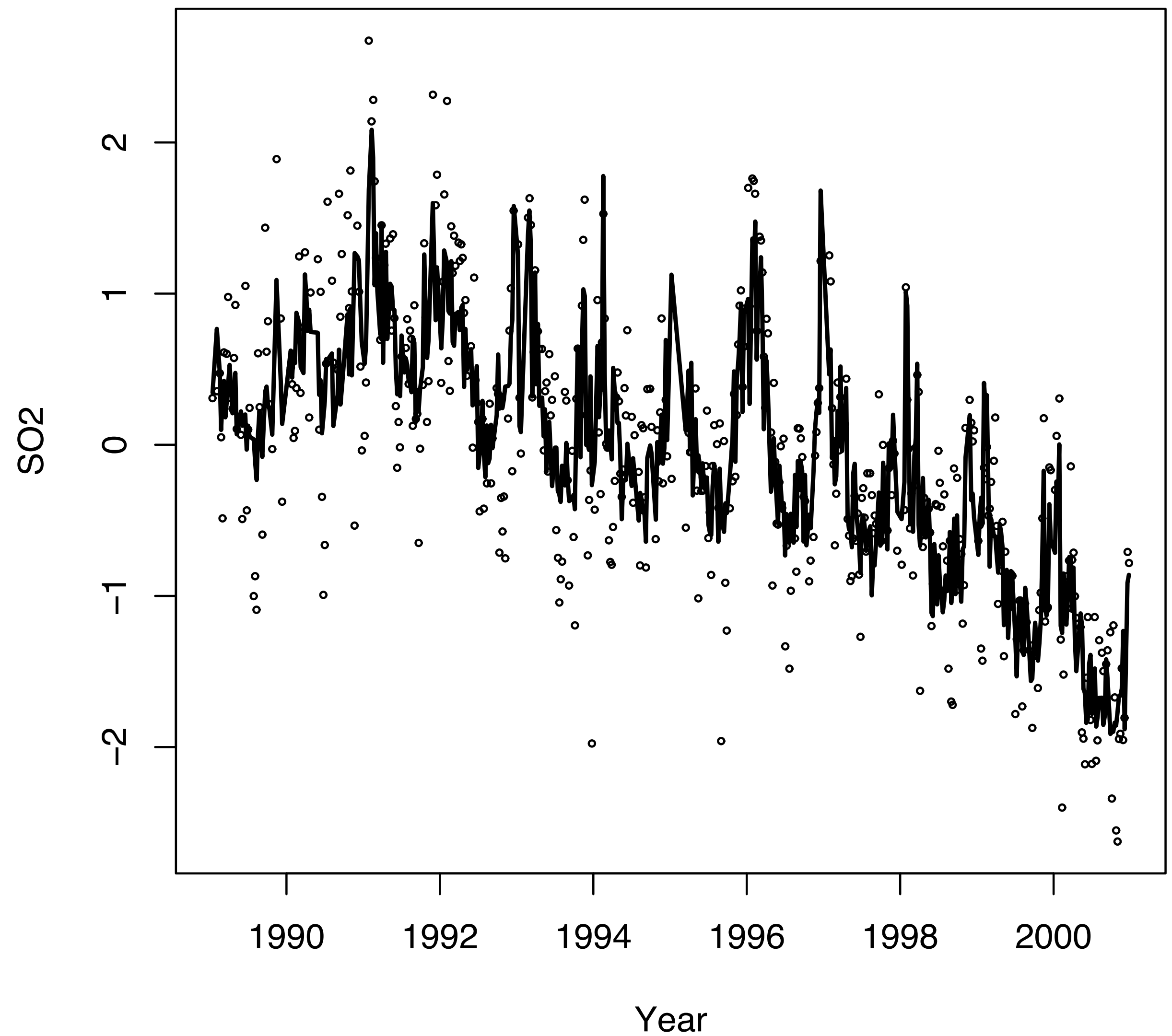


SO_2 pollution - adding meteorology



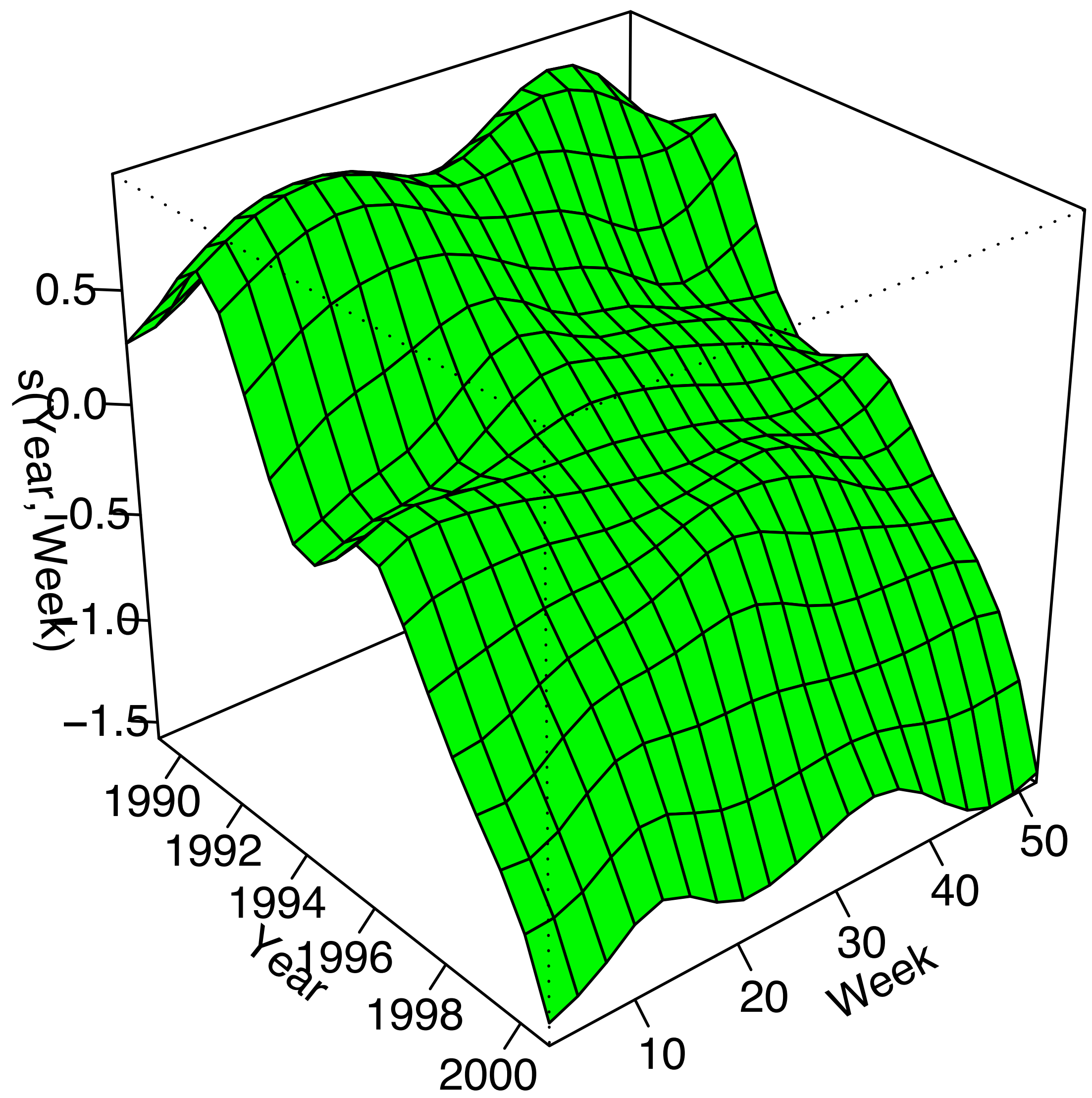
SO_2 pollution - fitted values



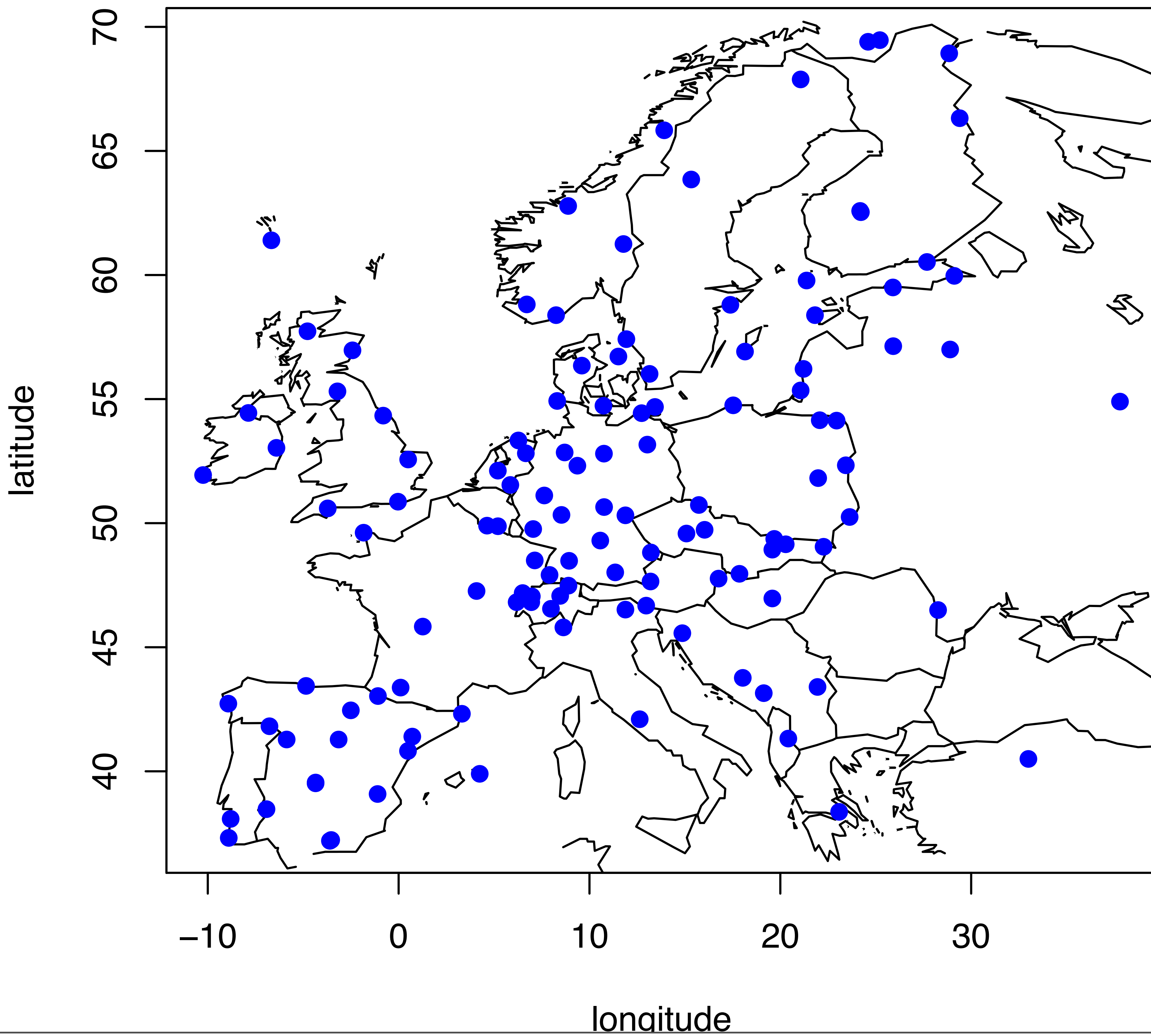


SO_2 pollution - a bivariate component

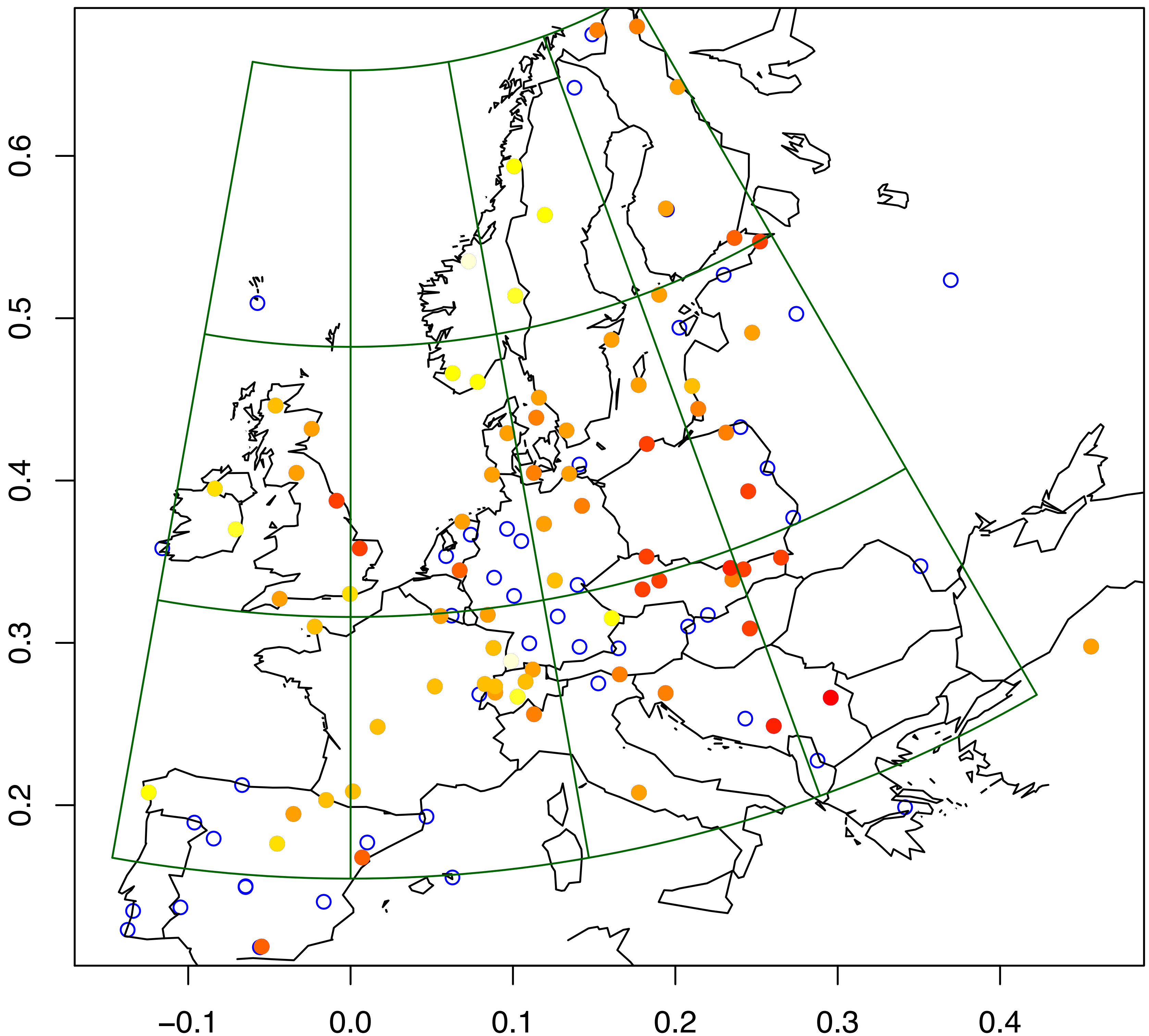
$$
\begin{aligned}
\ln(SO_2) &= \mu + m_{yw}(\text{year}, \text{week}) + m_r(\text{rain}) \\
&\quad m_t(\text{temp}) + m_h(\text{humidity}) + m_a(\text{air}) + \varepsilon
\end{aligned}
$$



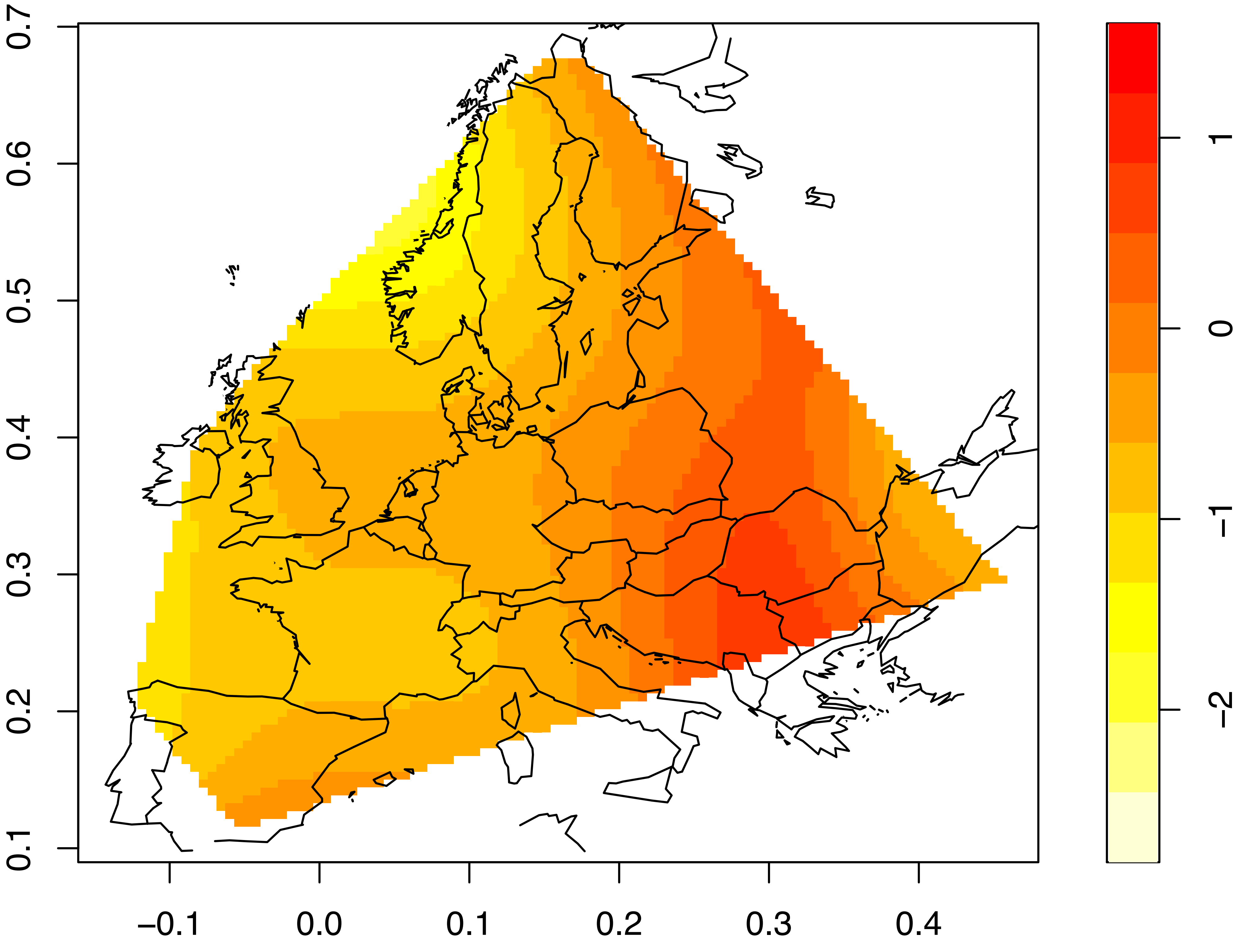
SO_2 pollution



SO_2 pollution



SO_2 pollution



Computational issues

$$P_j^{(l)} = S_j(I_n - \sum_{k<j} P_k^{(l)} - \sum_{k>j} P_k^{(l-1)}),$$

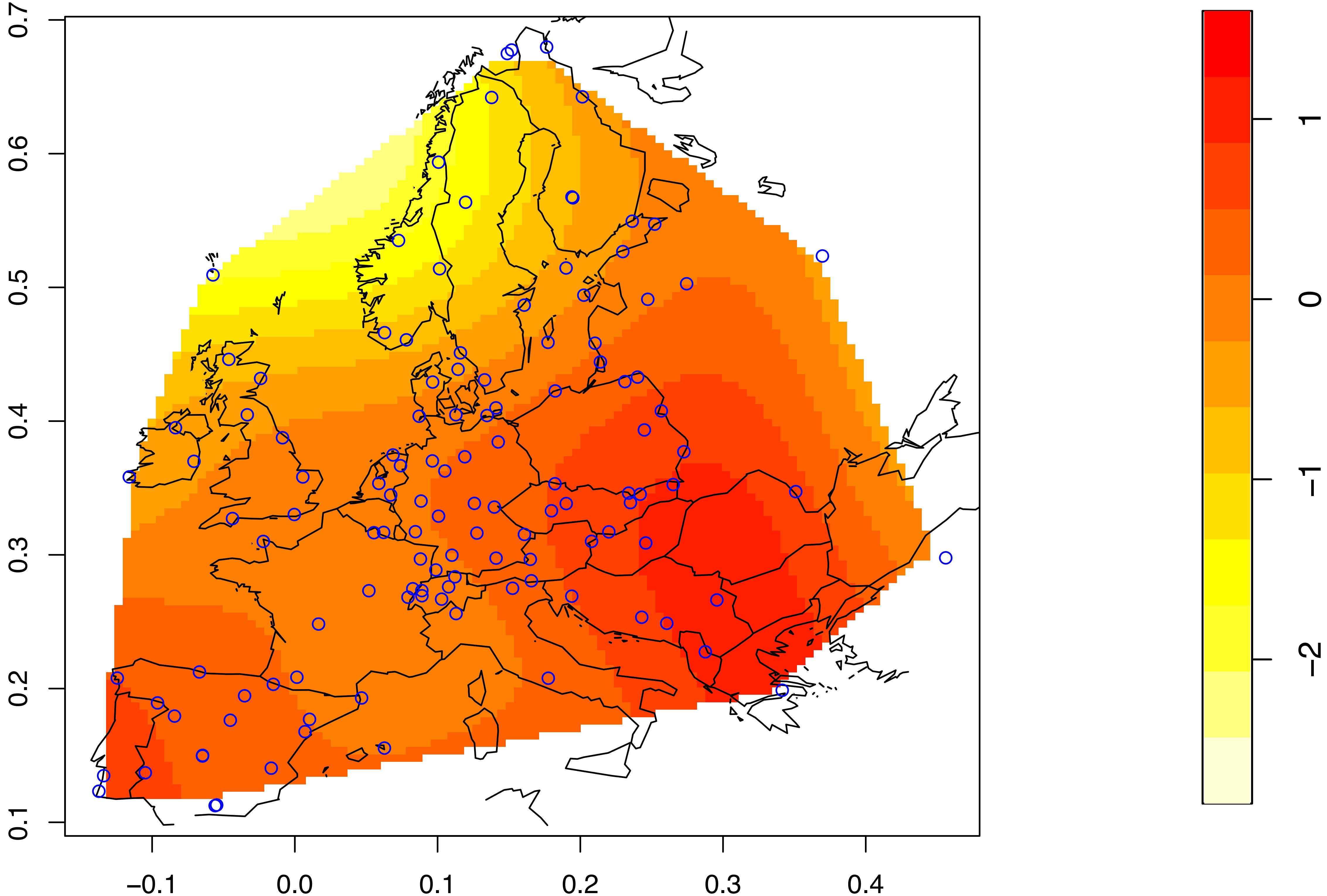
$$\tilde{P}_j^{(l)} = S_j C_j(I_n - \sum_{k<j} E_k \tilde{P}_k^{(l)} - \sum_{k>j} E_k \tilde{P}_k^{(l-1)}),$$

In the spatiotemporal case,

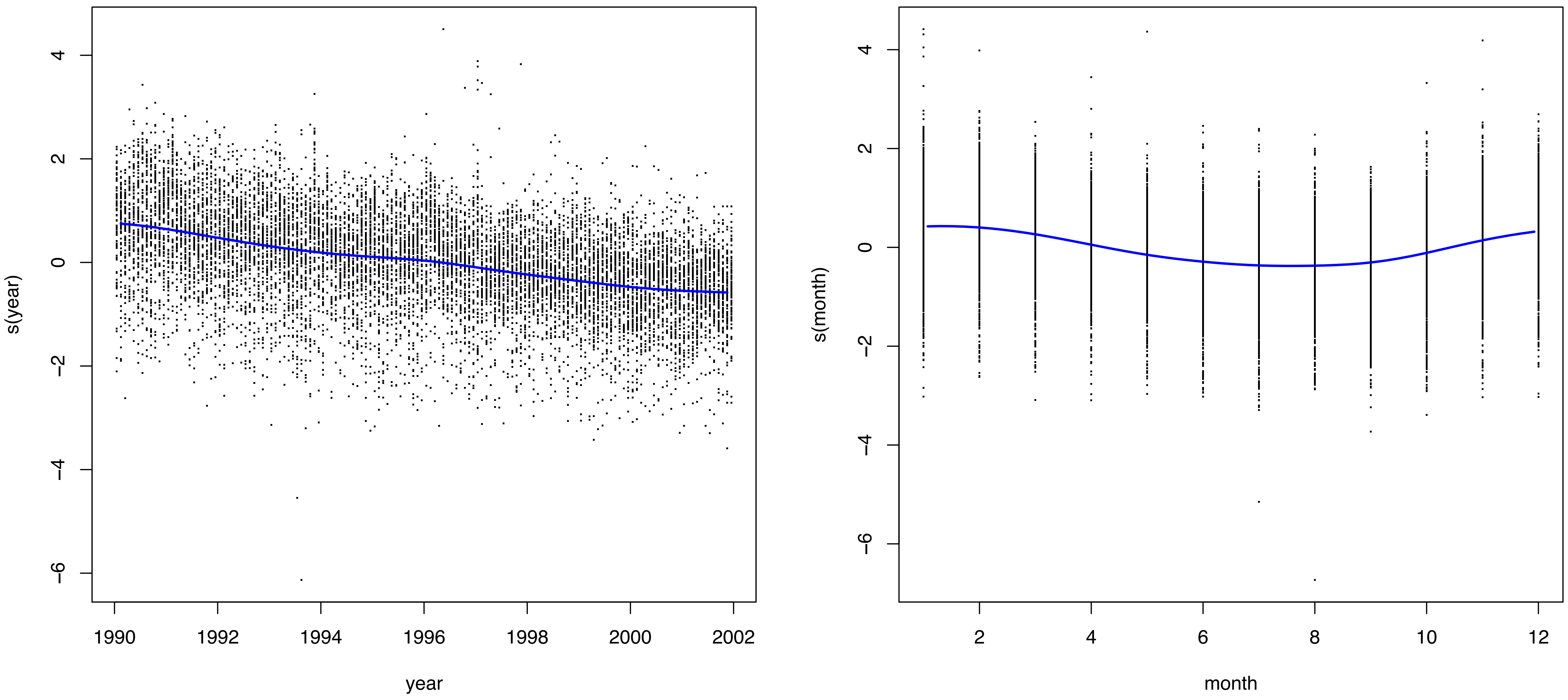
- see Currie, Durban & Eilers (2006);
- the data can be organised in a matrix form Y which is $s \times t$
- when a separable covariance structure is appropriate, efficient calculations are possible through the Kronecker product formulation

$$V = \Sigma \otimes \Gamma$$

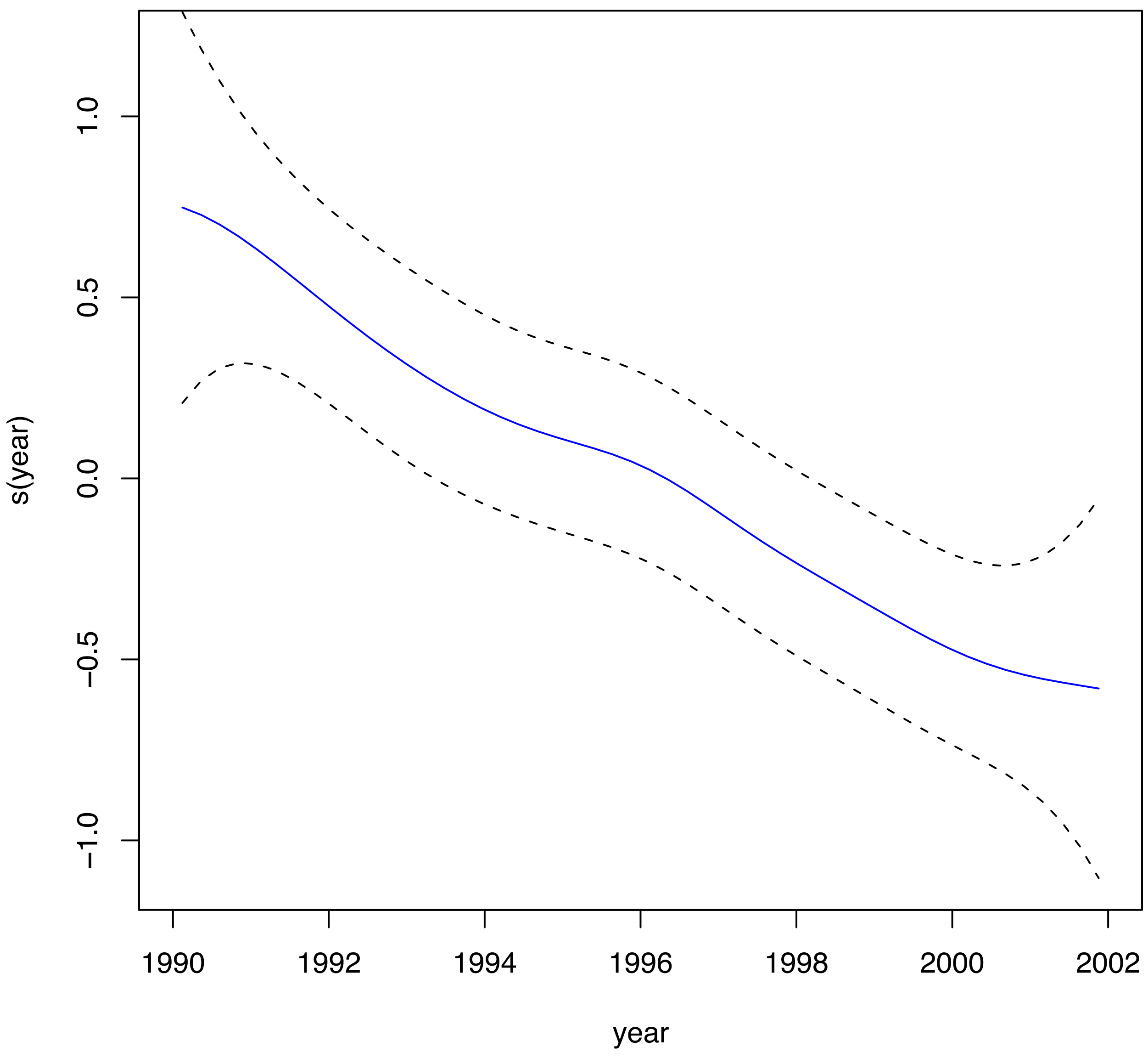
SO_2 pollution - additive model

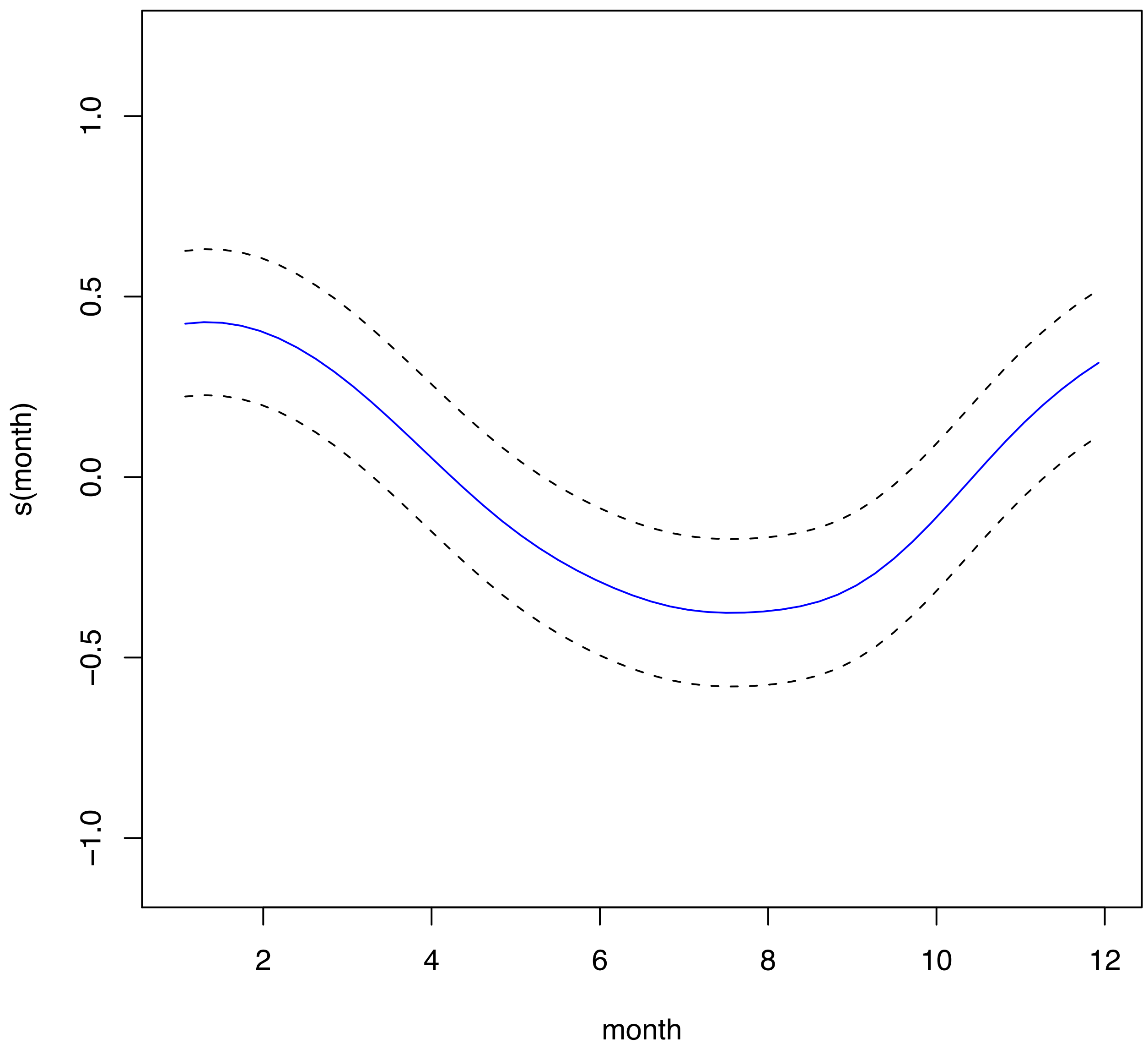


SO_2 pollution - additive model

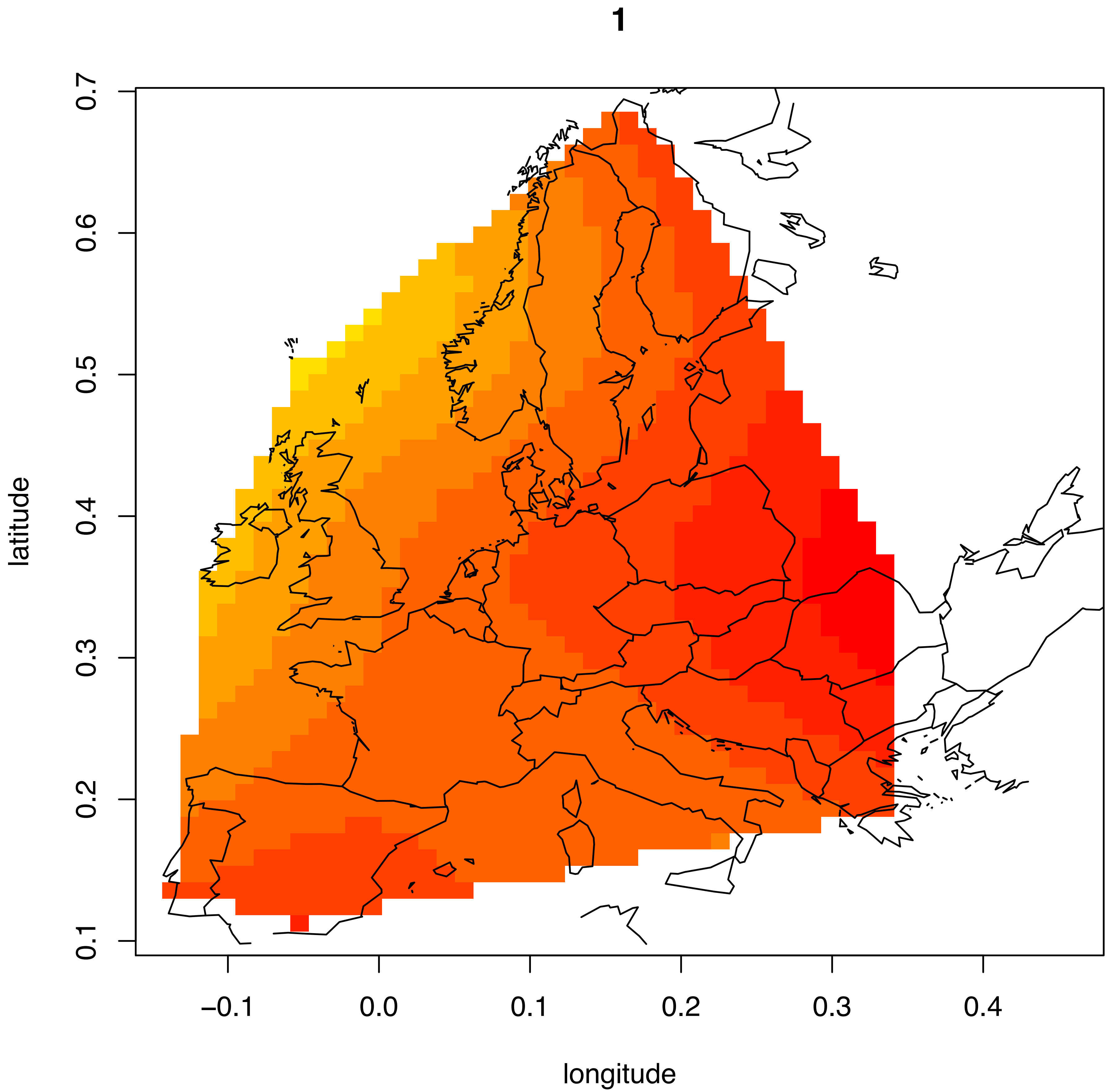


SO_2 pollution - additive model

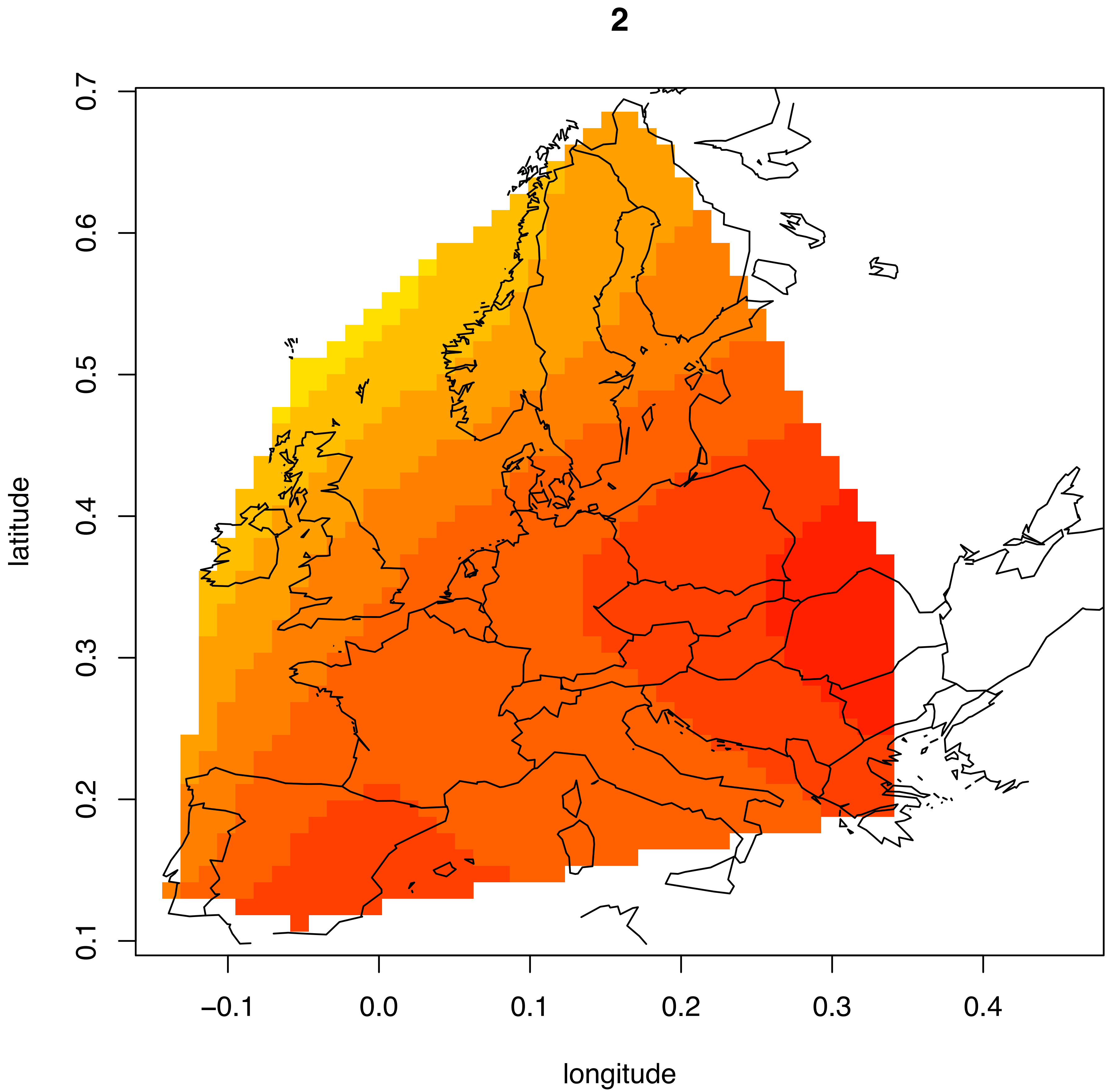




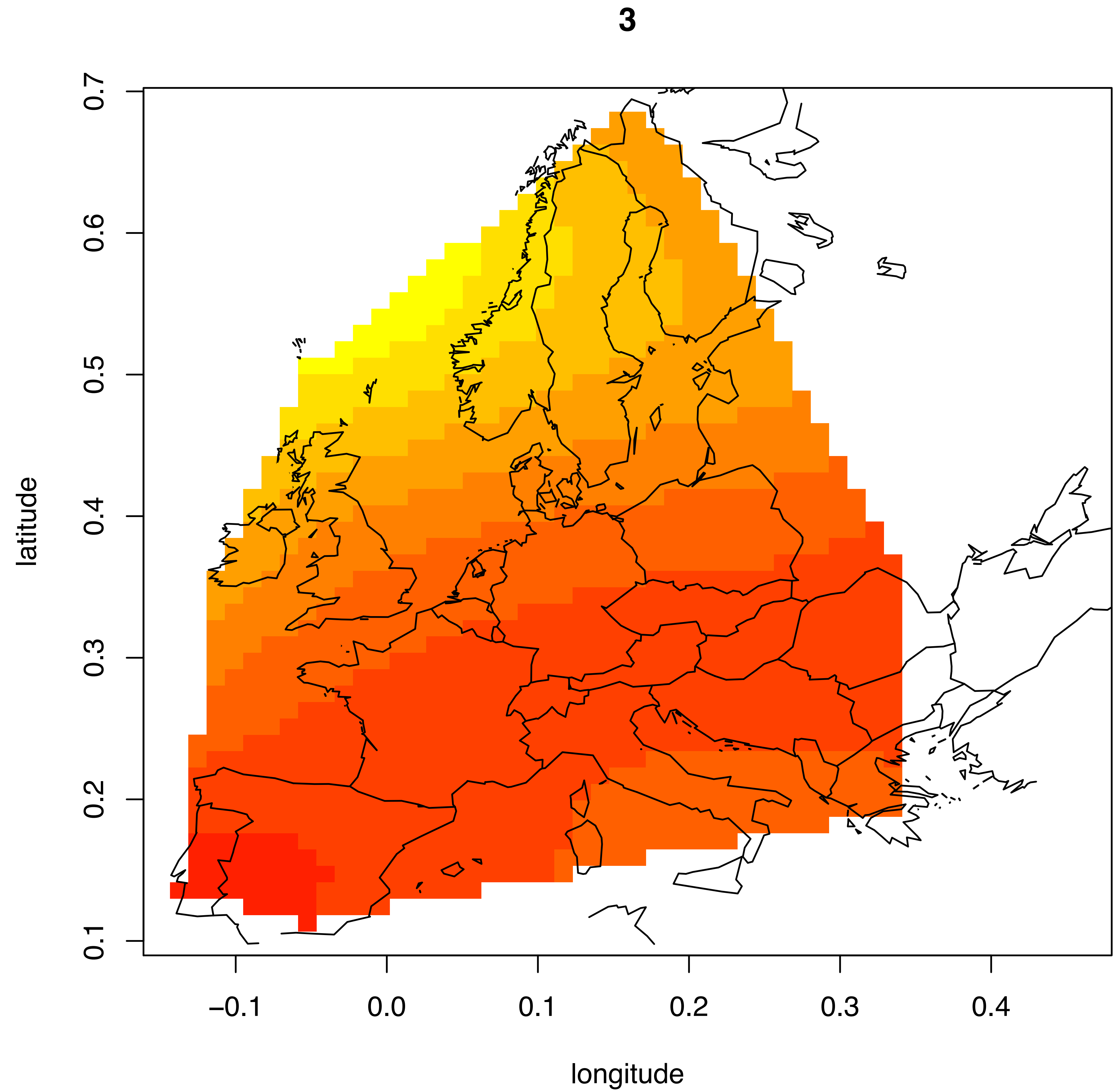
SO_2 pollution



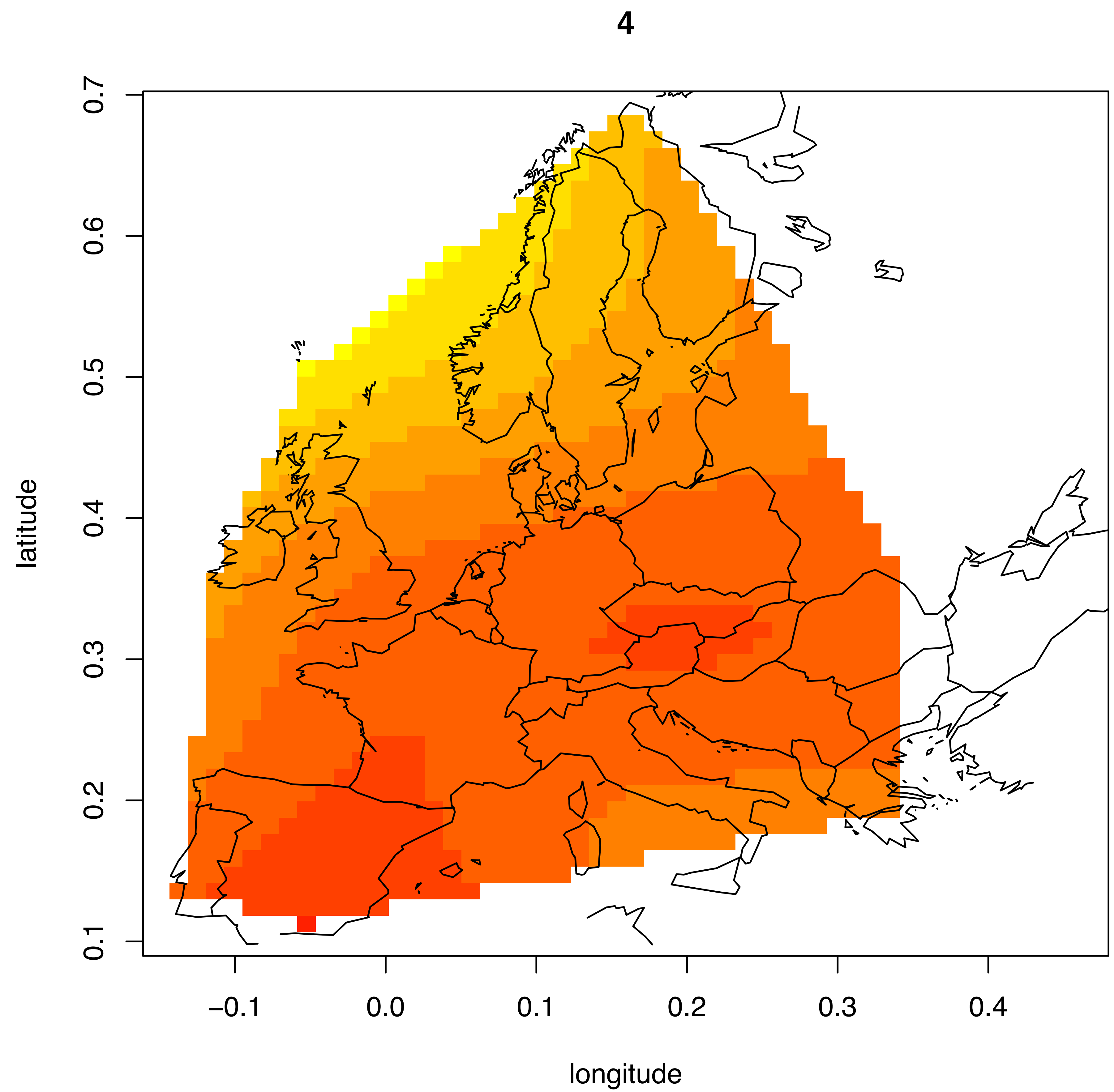
SO_2 pollution



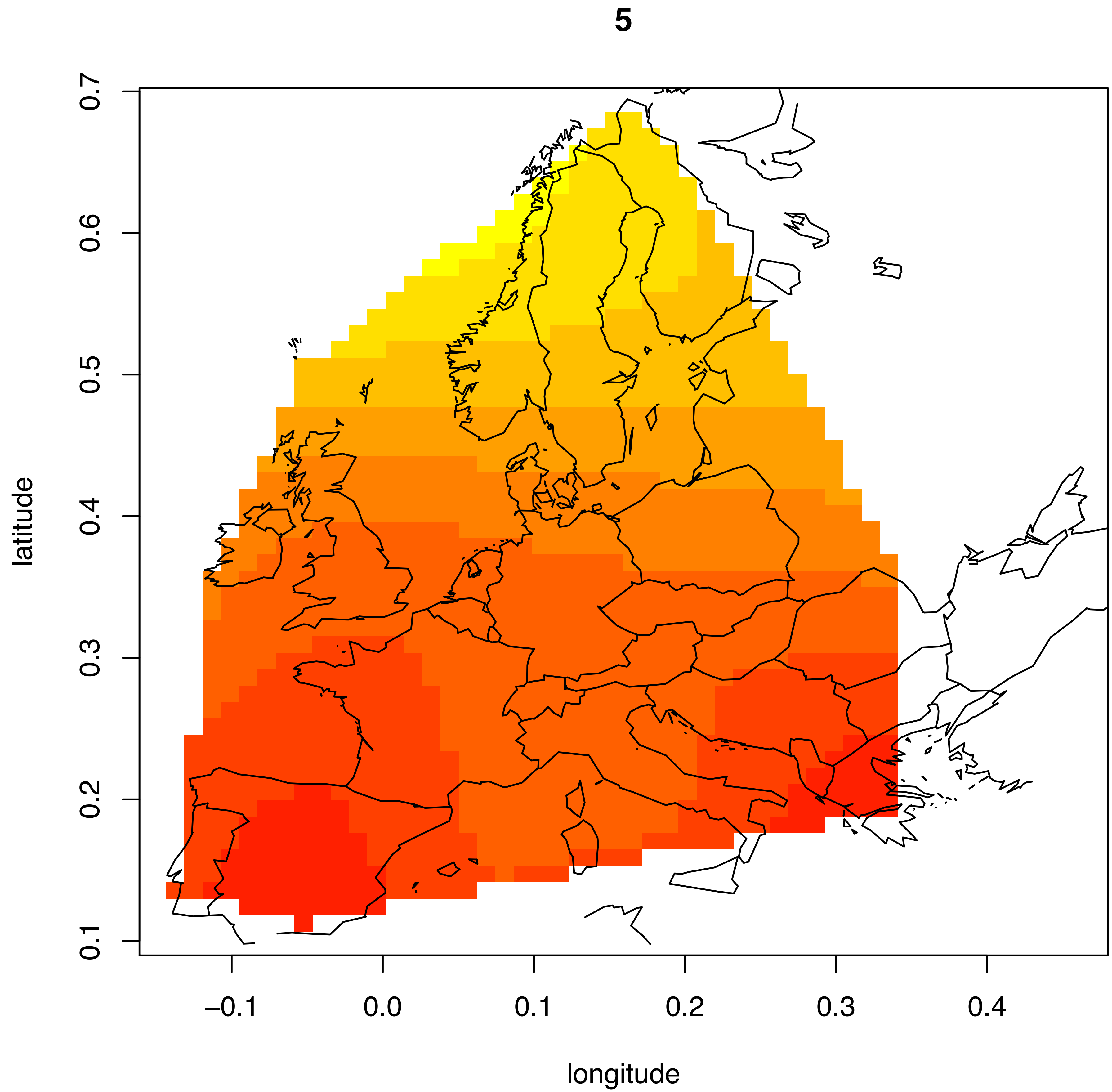
SO_2 pollution



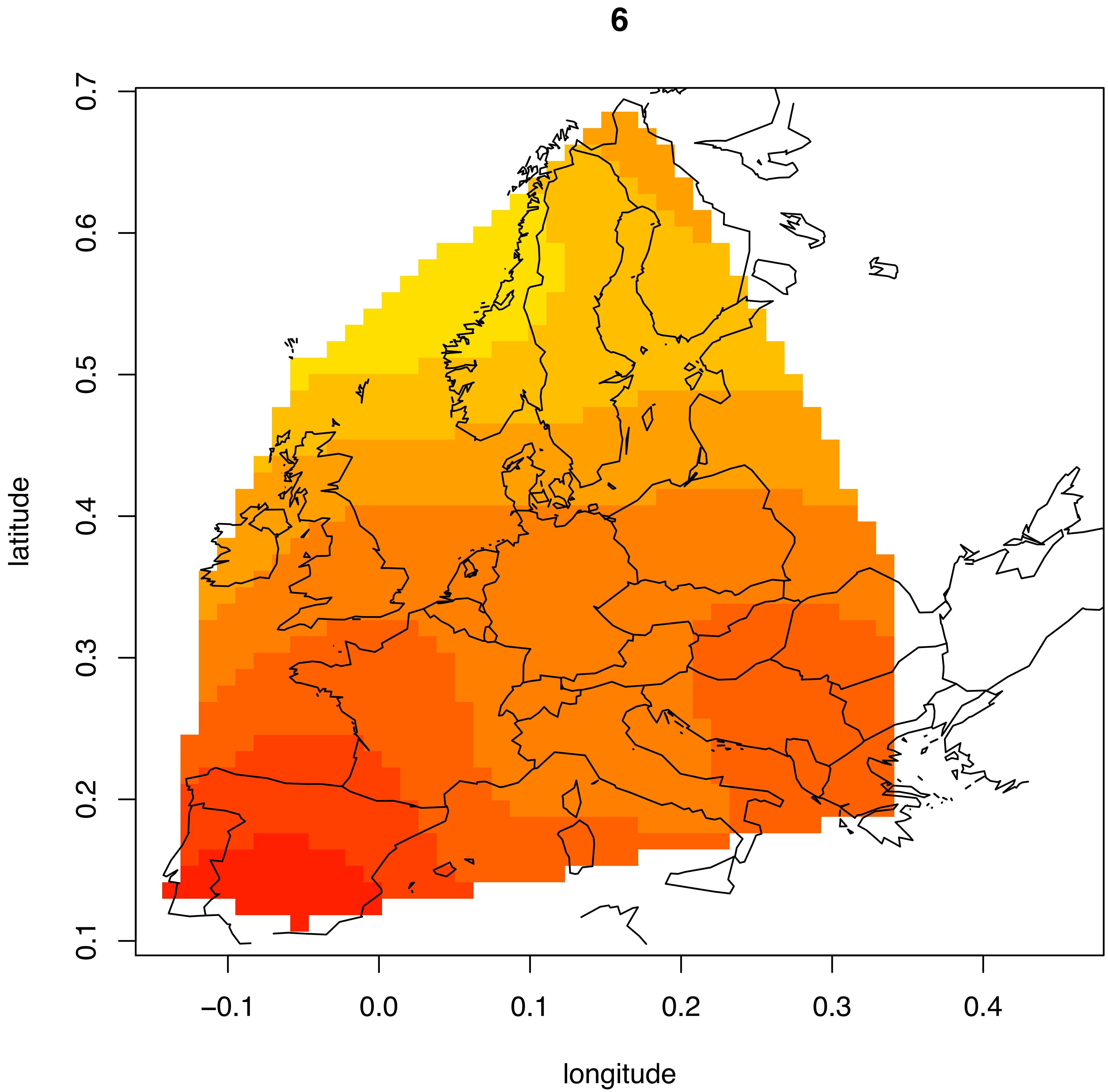
SO_2 pollution



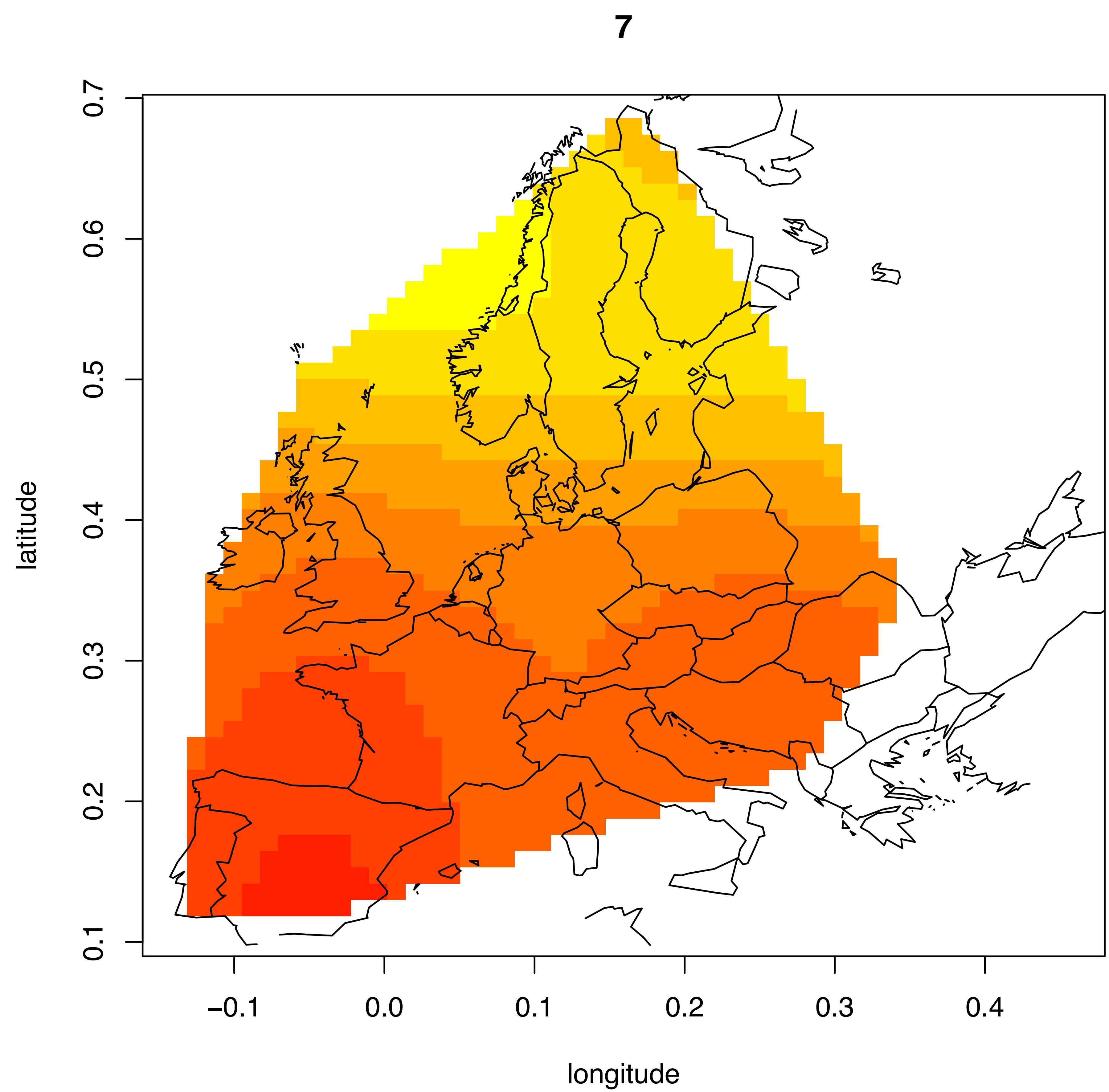
SO_2 pollution



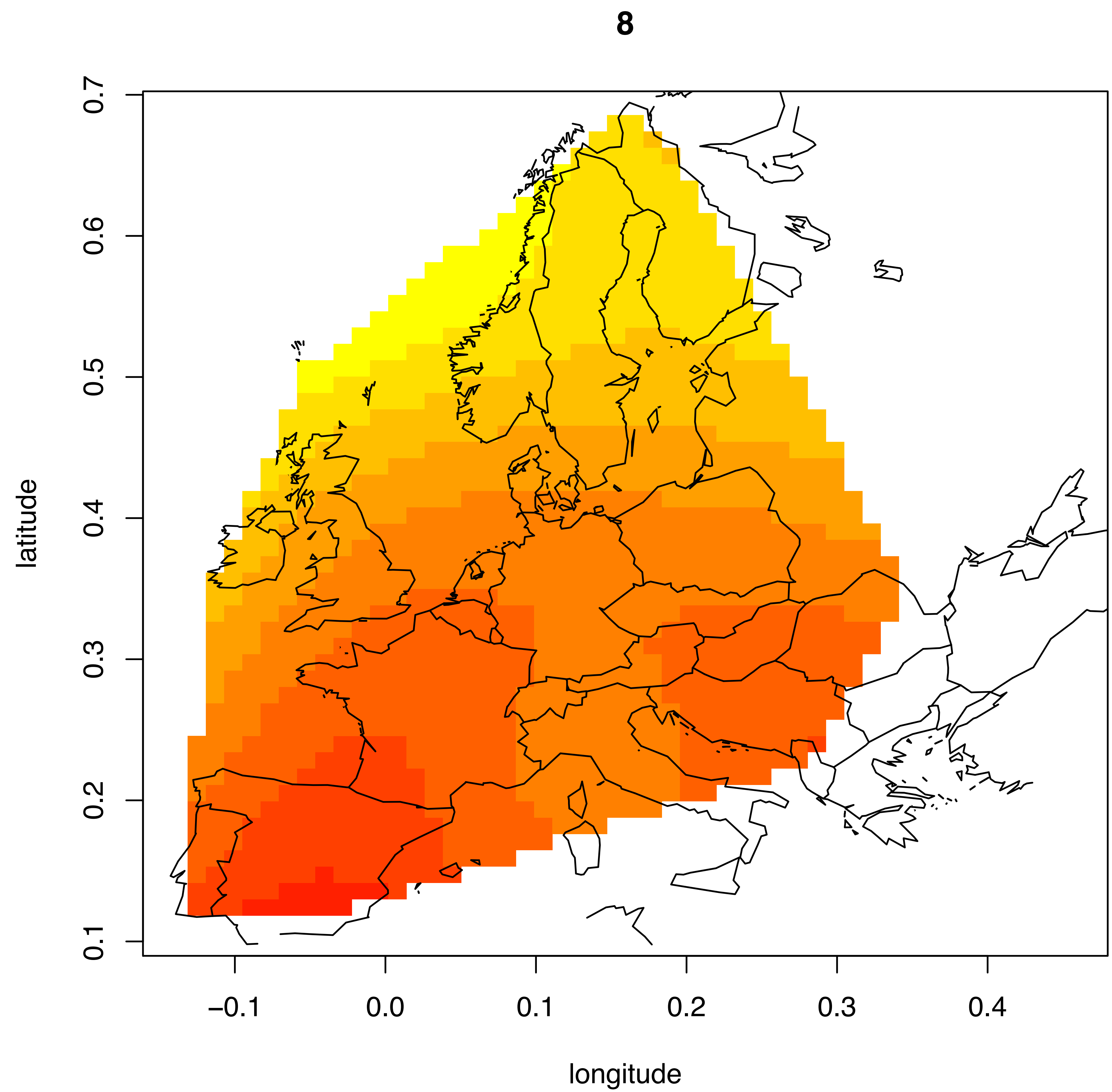
SO_2 pollution



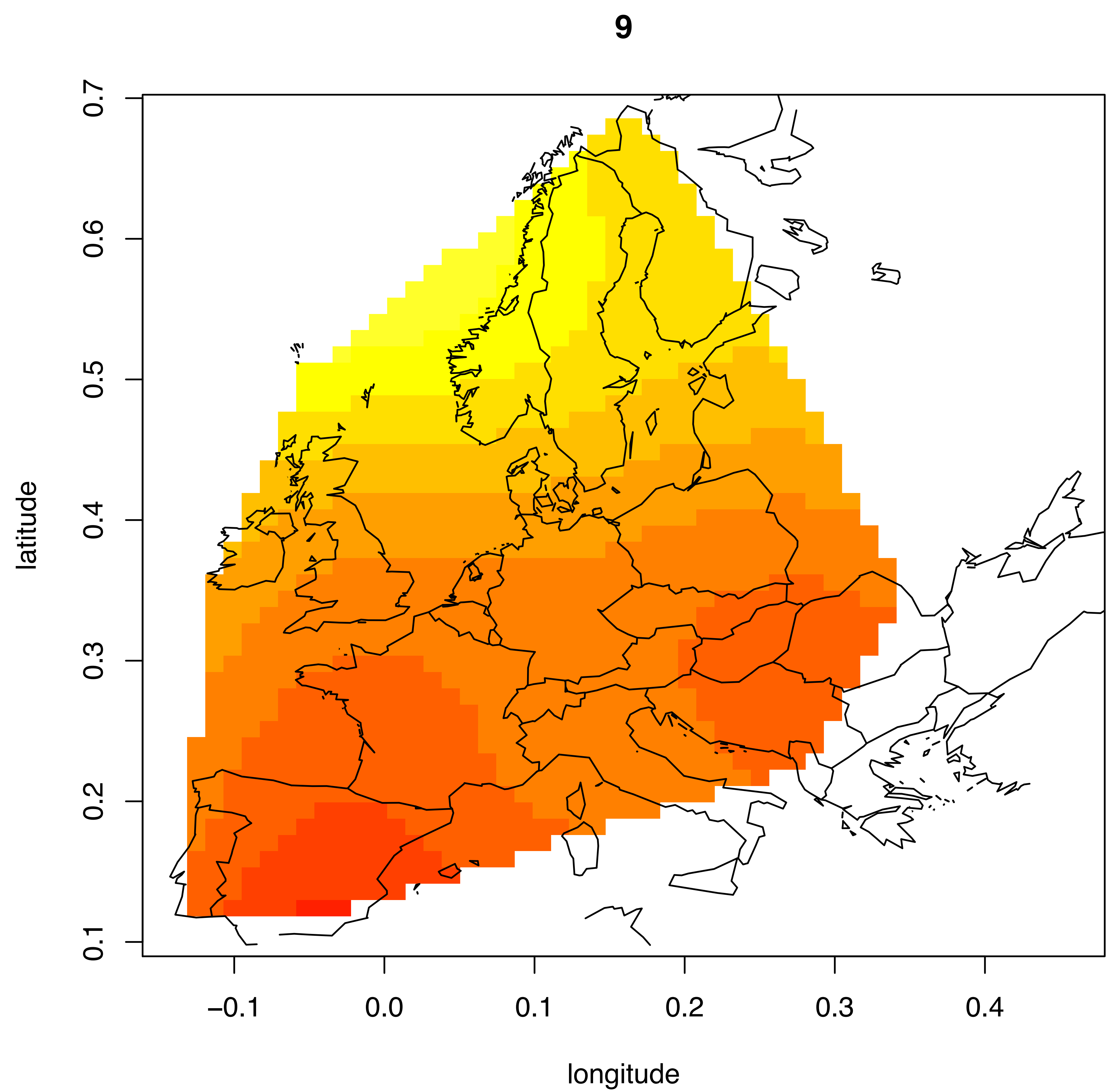
SO_2 pollution



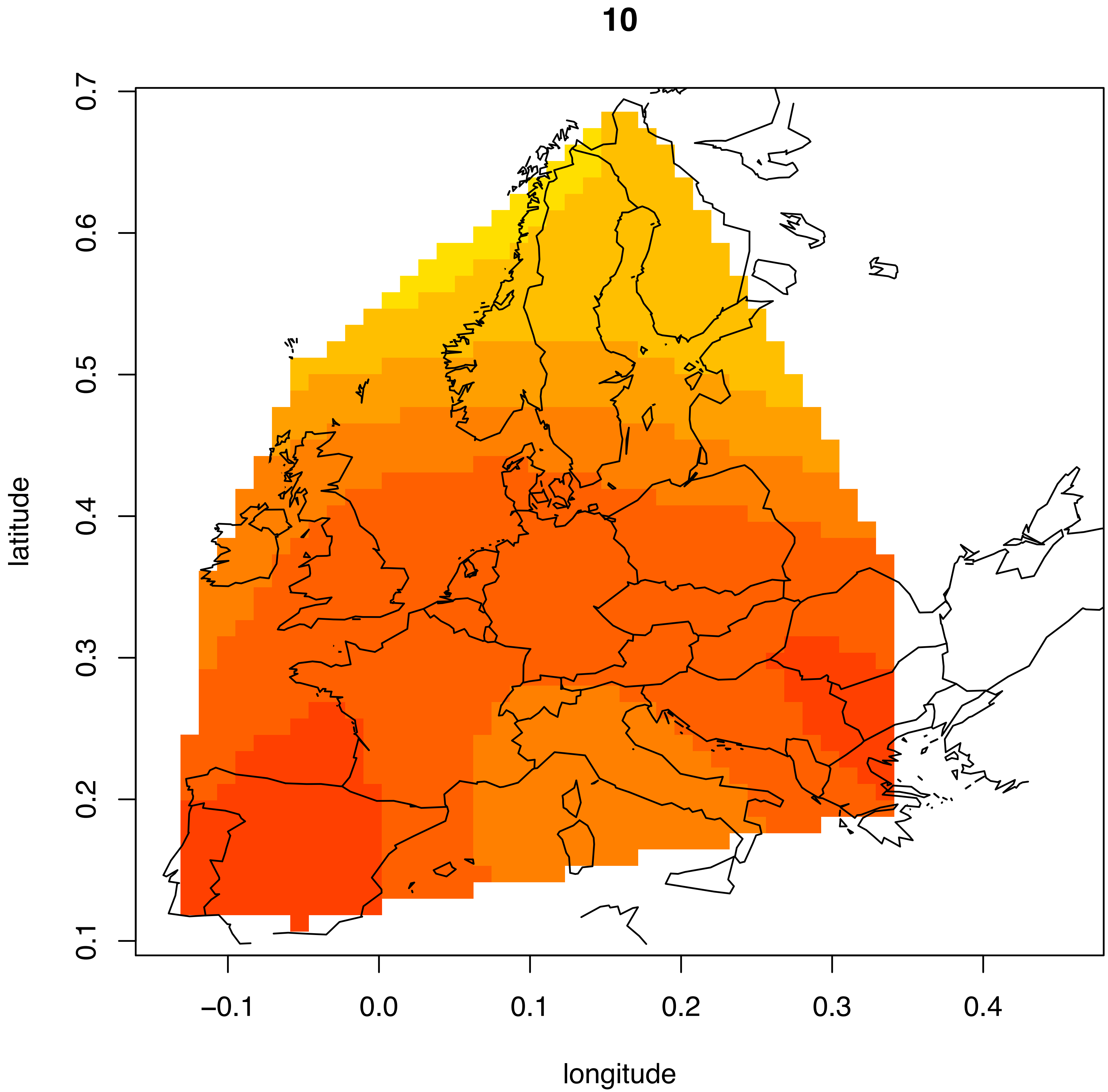
SO_2 pollution



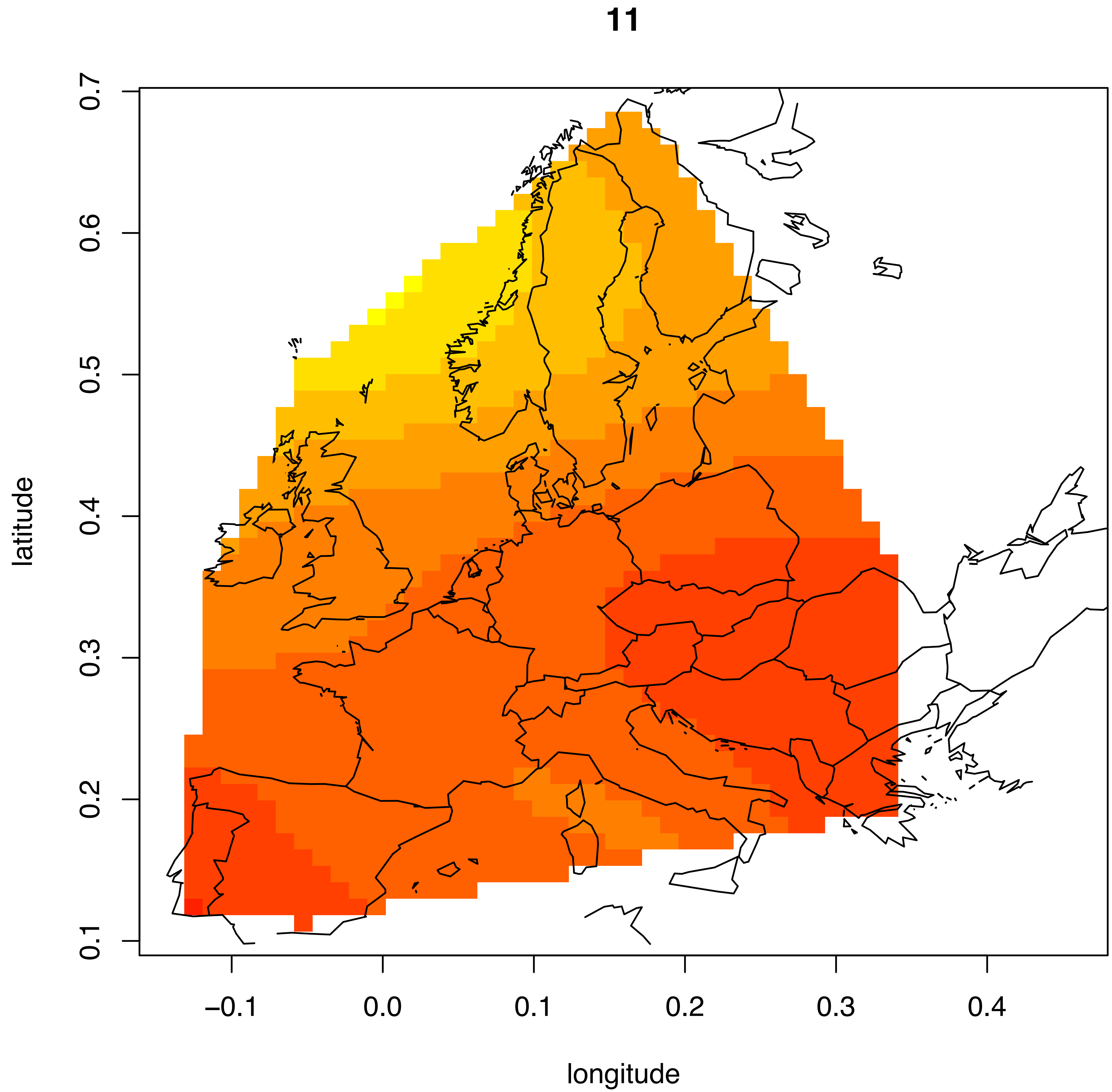
SO_2 pollution



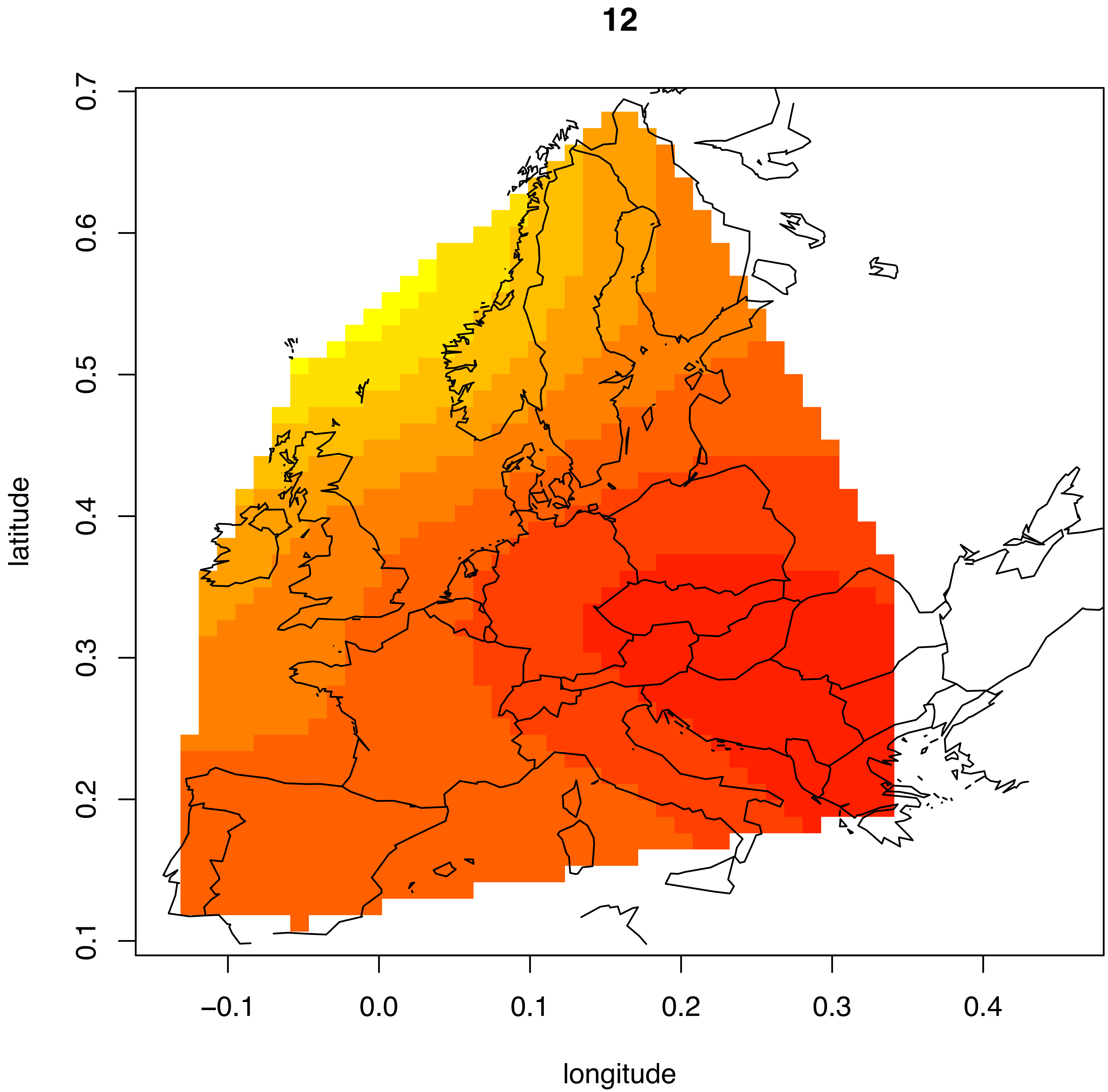
SO_2 pollution



SO_2 pollution



SO_2 pollution

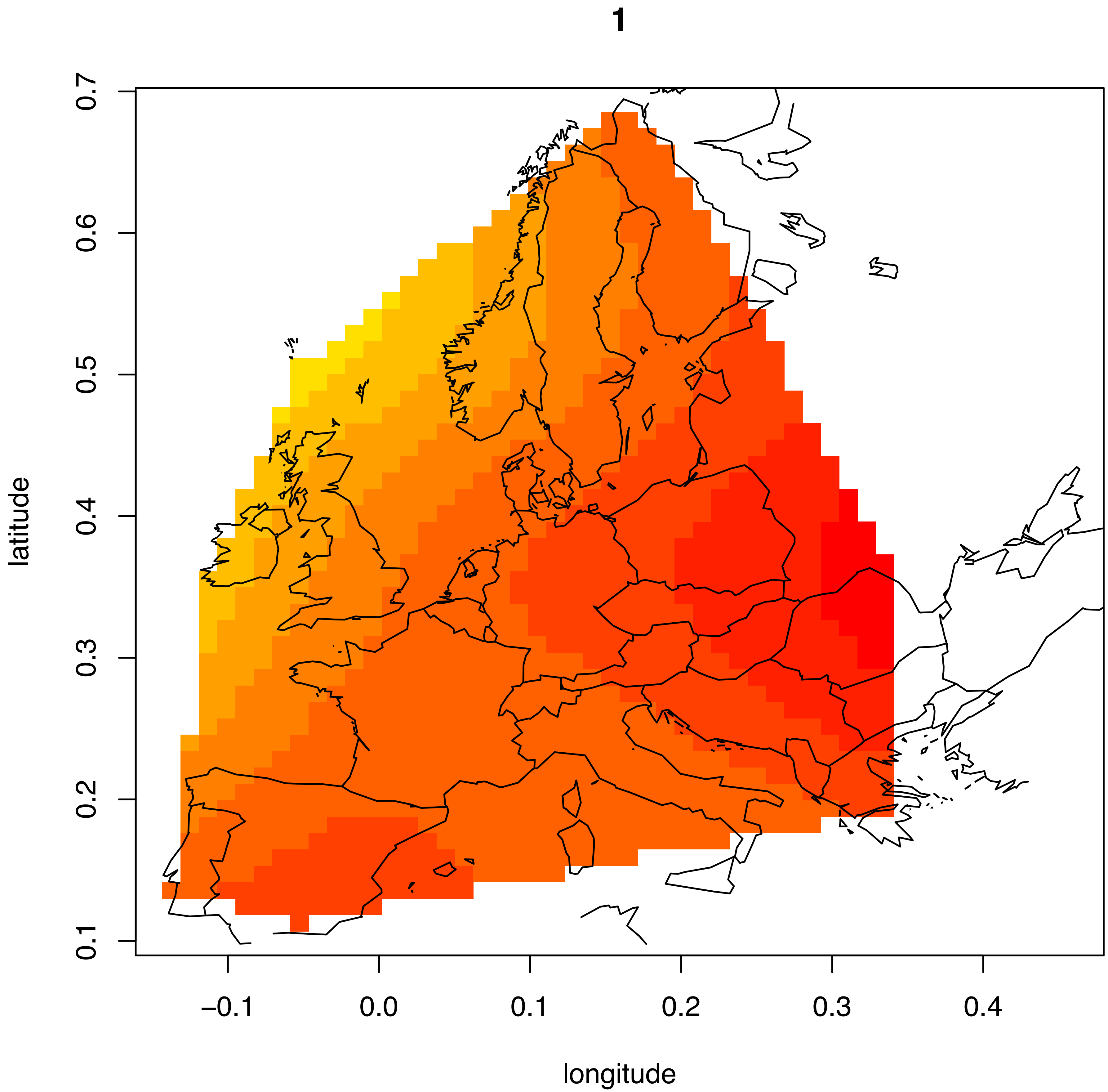


SO_2 pollution

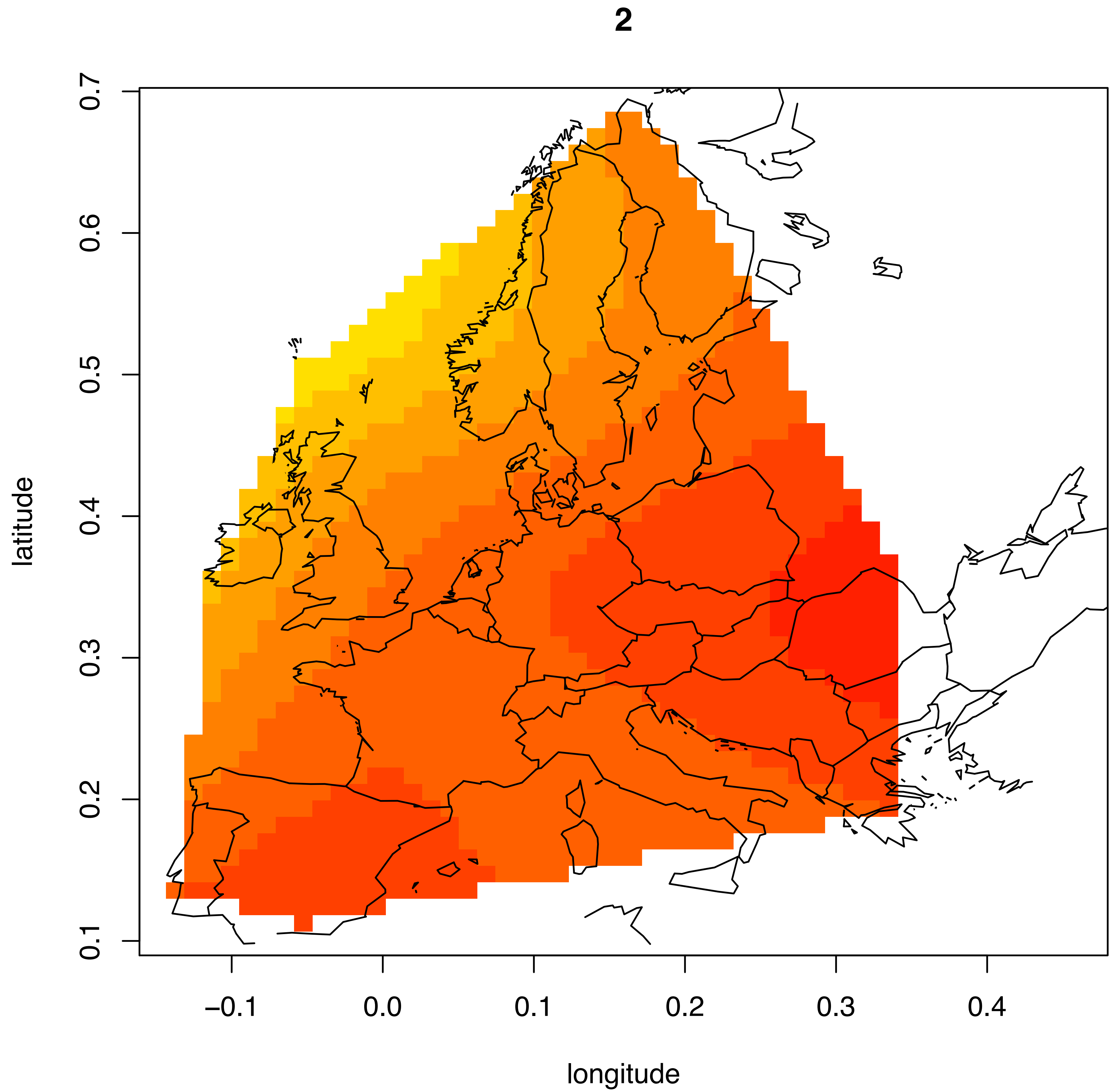
Three-dimensional smoothing:

- the data can be organised in a matrix form Y which is $s \times t$;
- three-dimensional smoothing can be applied as sequential operations
$$\hat{m} = S_s Y S_t^\top$$
- when a separable covariance structure is appropriate, efficient calculations are possible through the Kronecker product formulation
$$V = \Sigma \otimes \Gamma$$

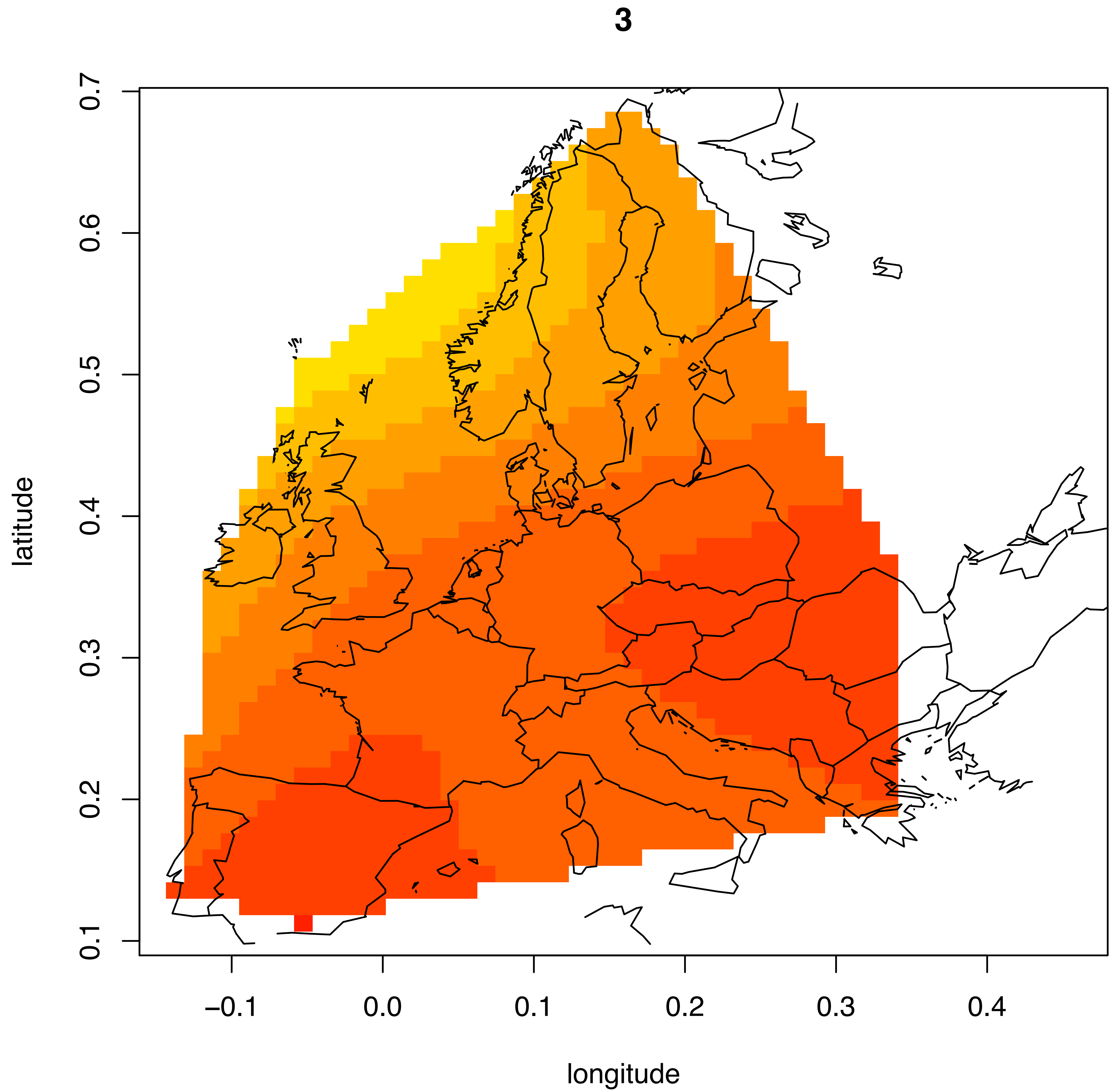
SO_2 pollution



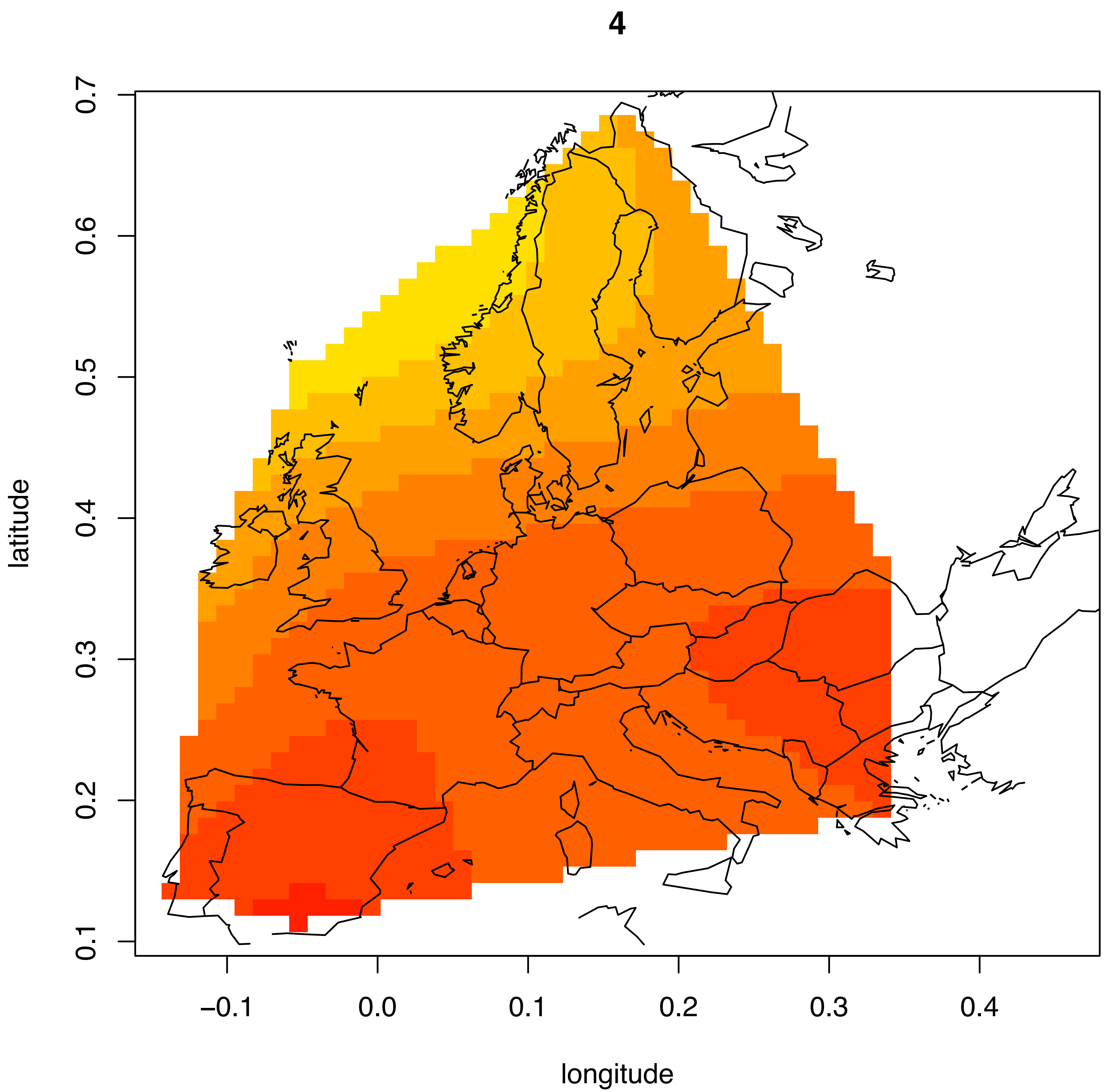
SO_2 pollution



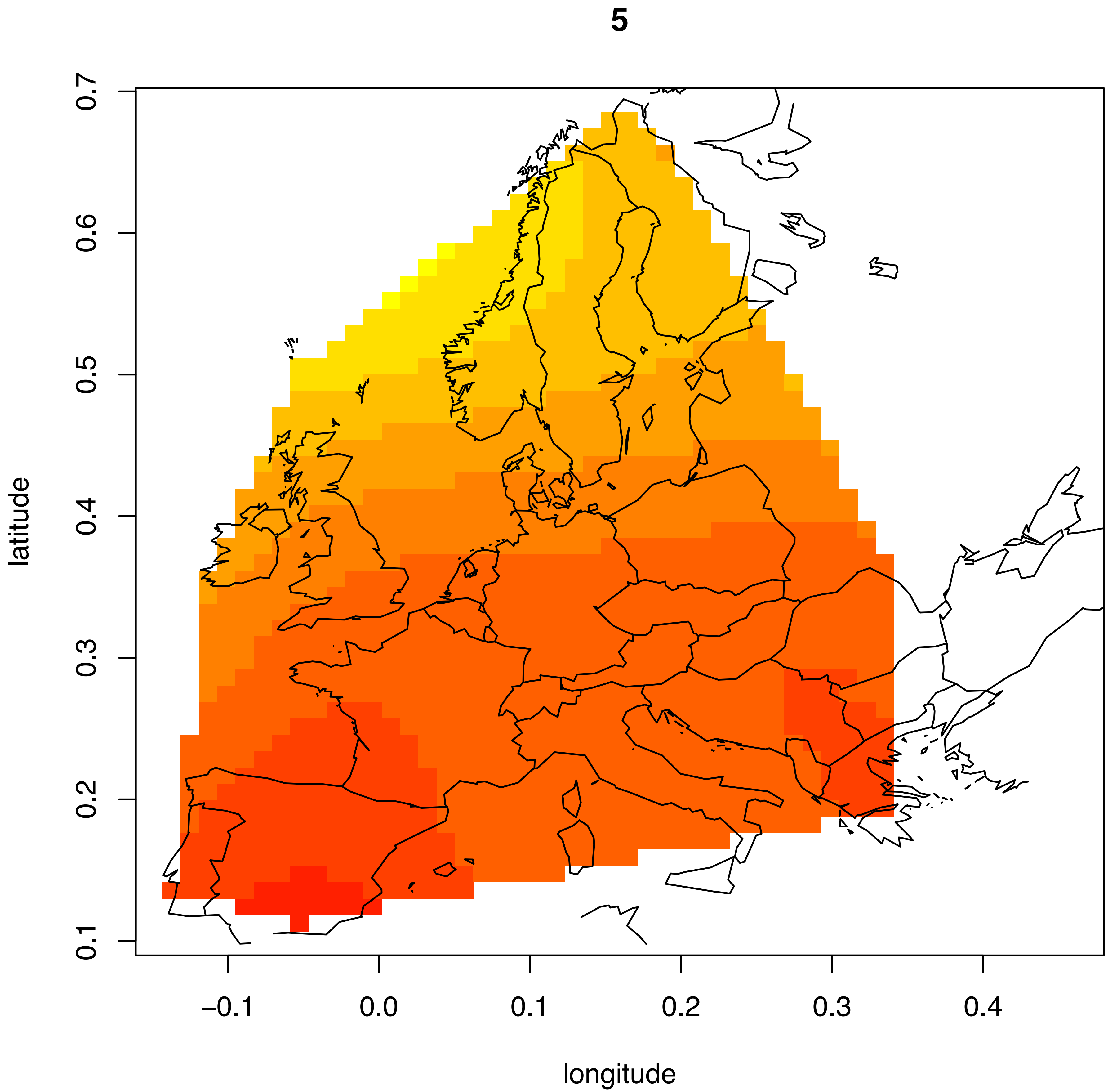
SO_2 pollution



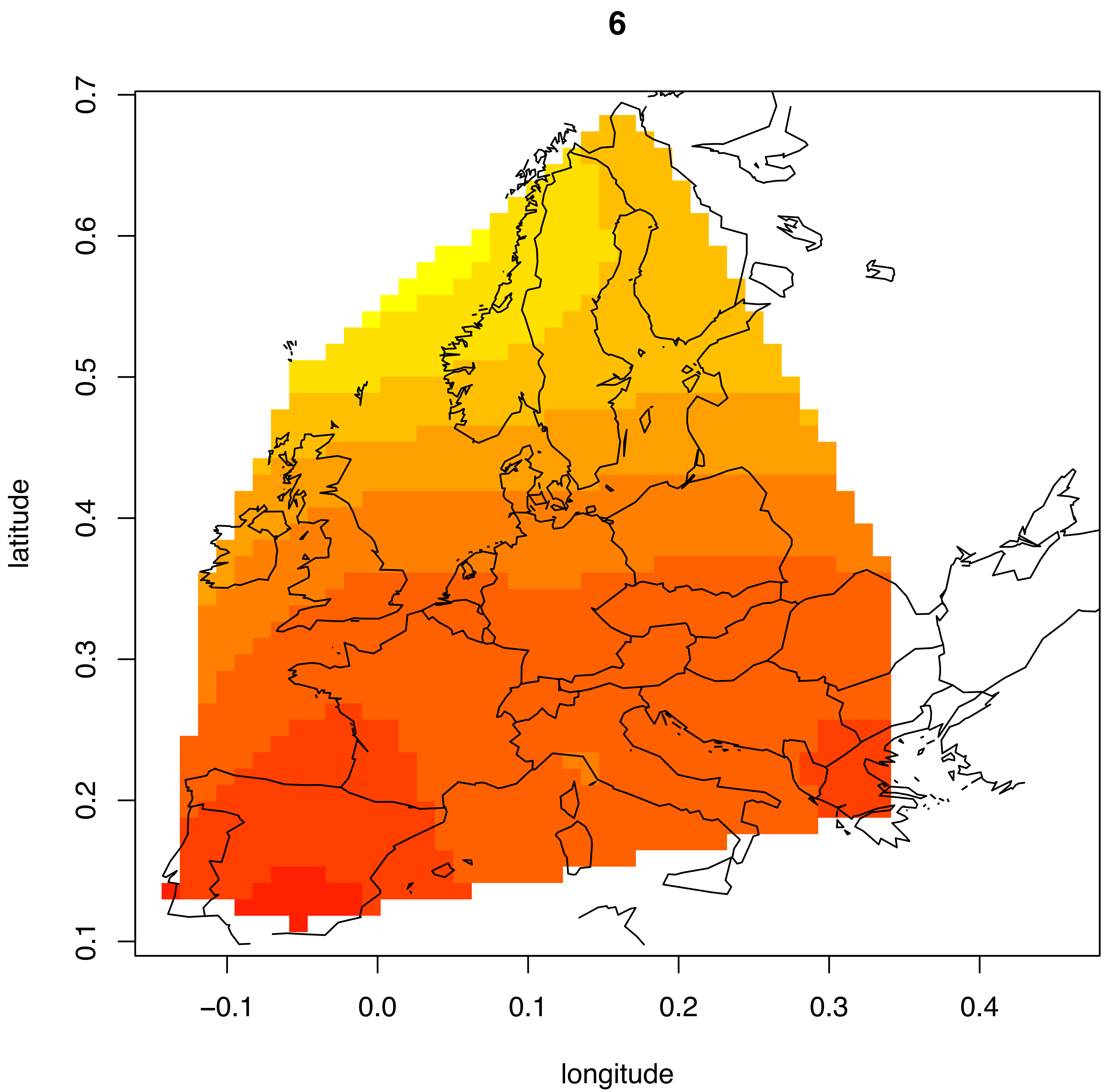
SO_2 pollution



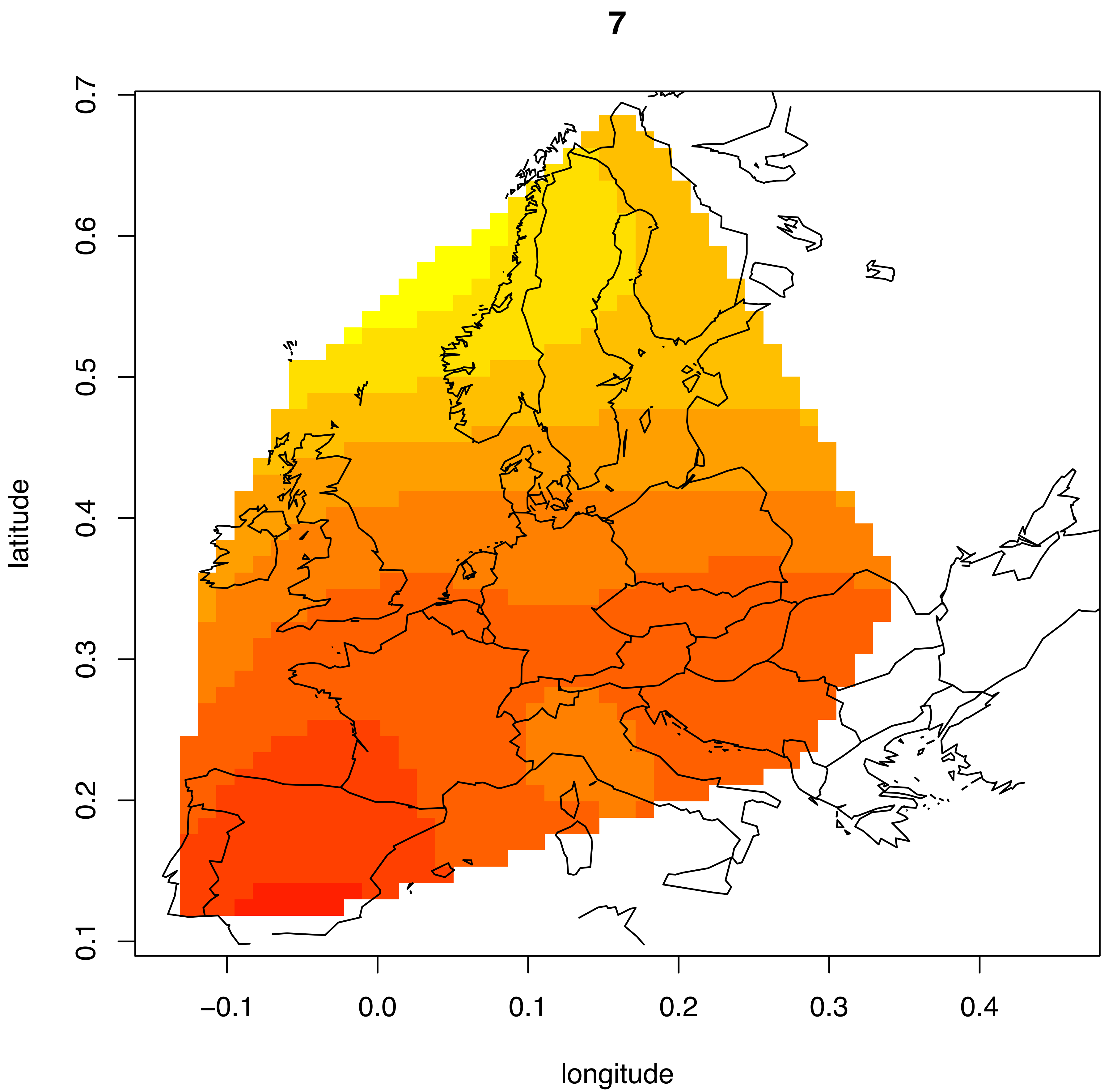
SO_2 pollution



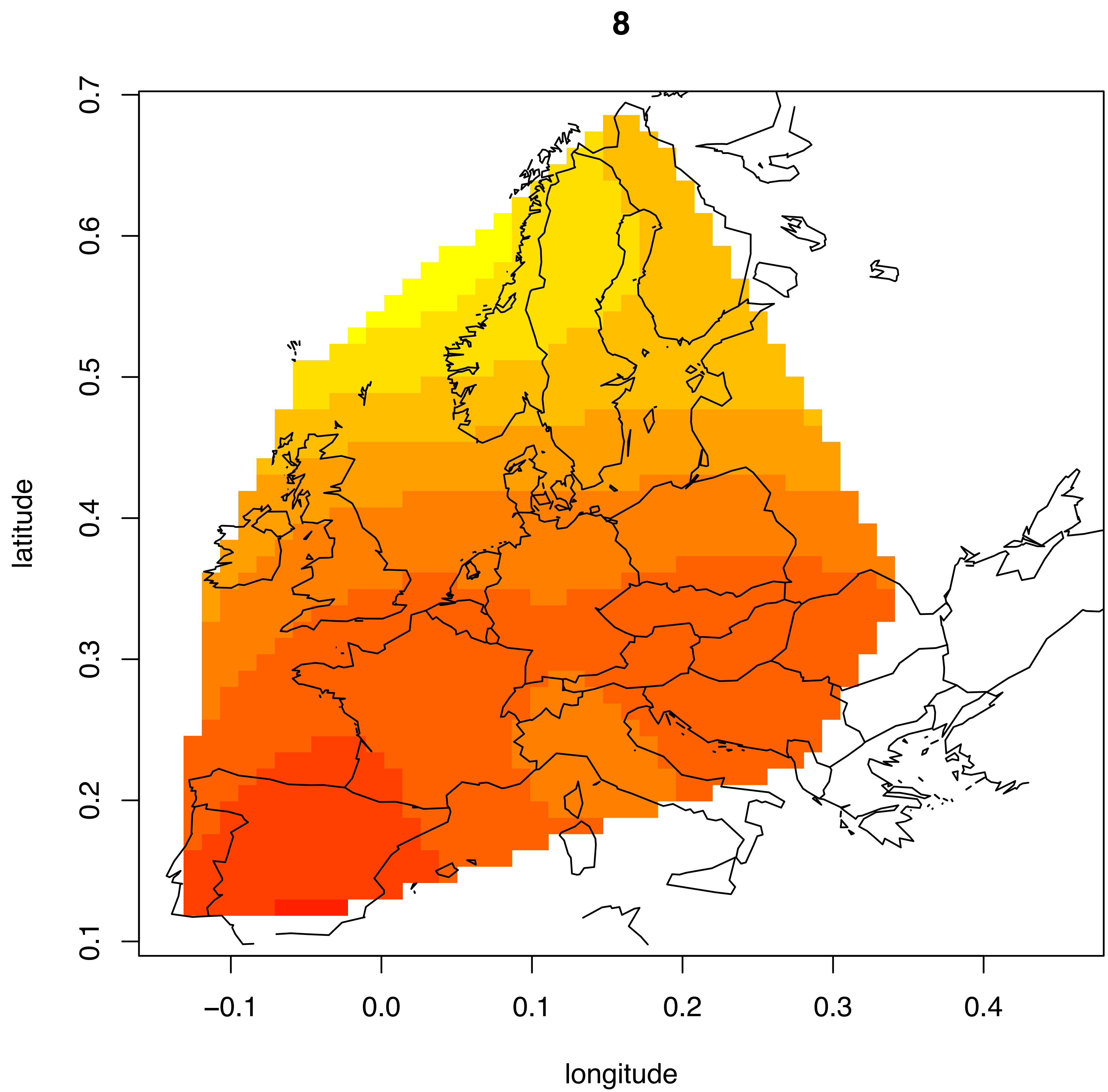
SO_2 pollution



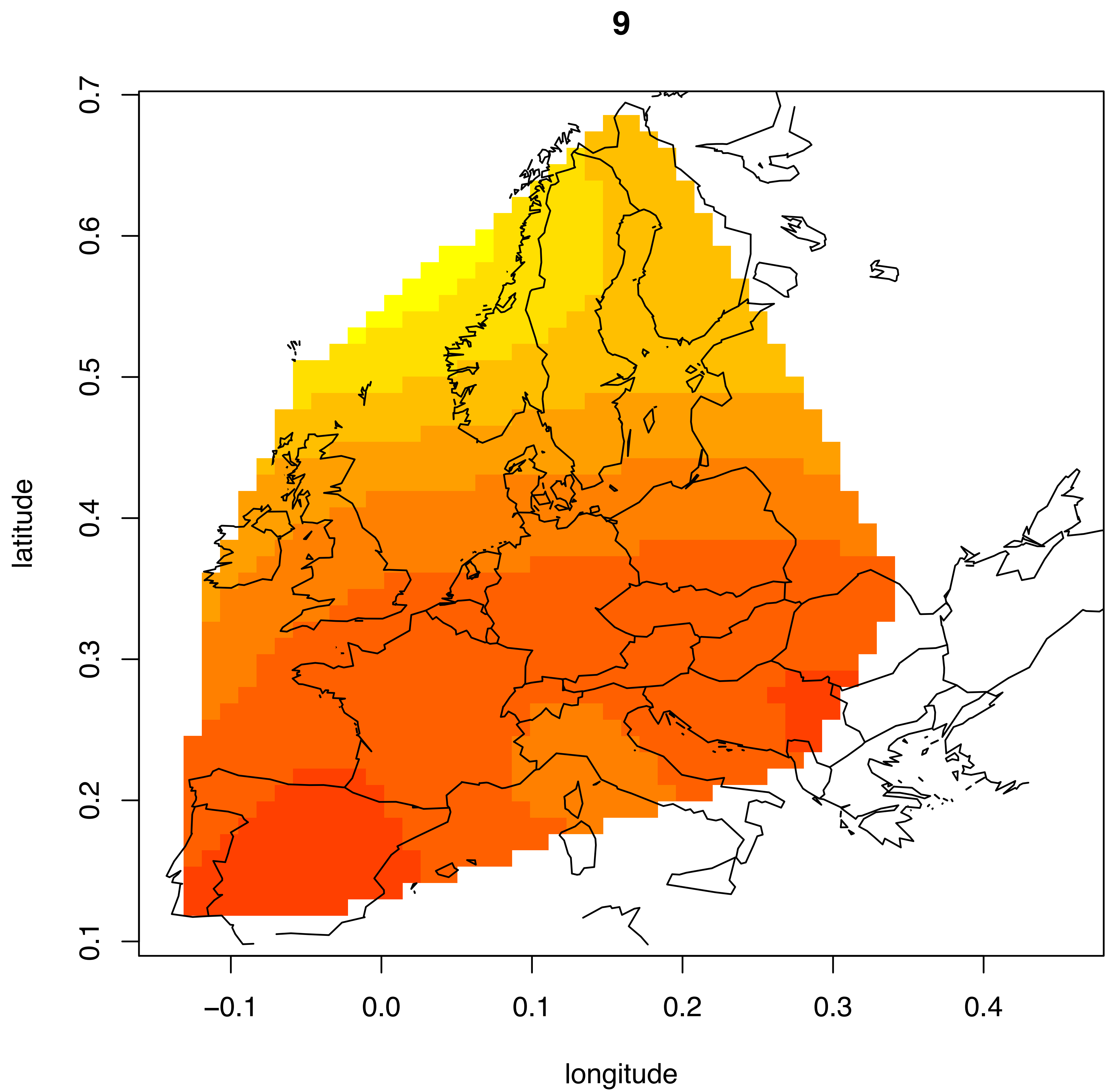
SO_2 pollution



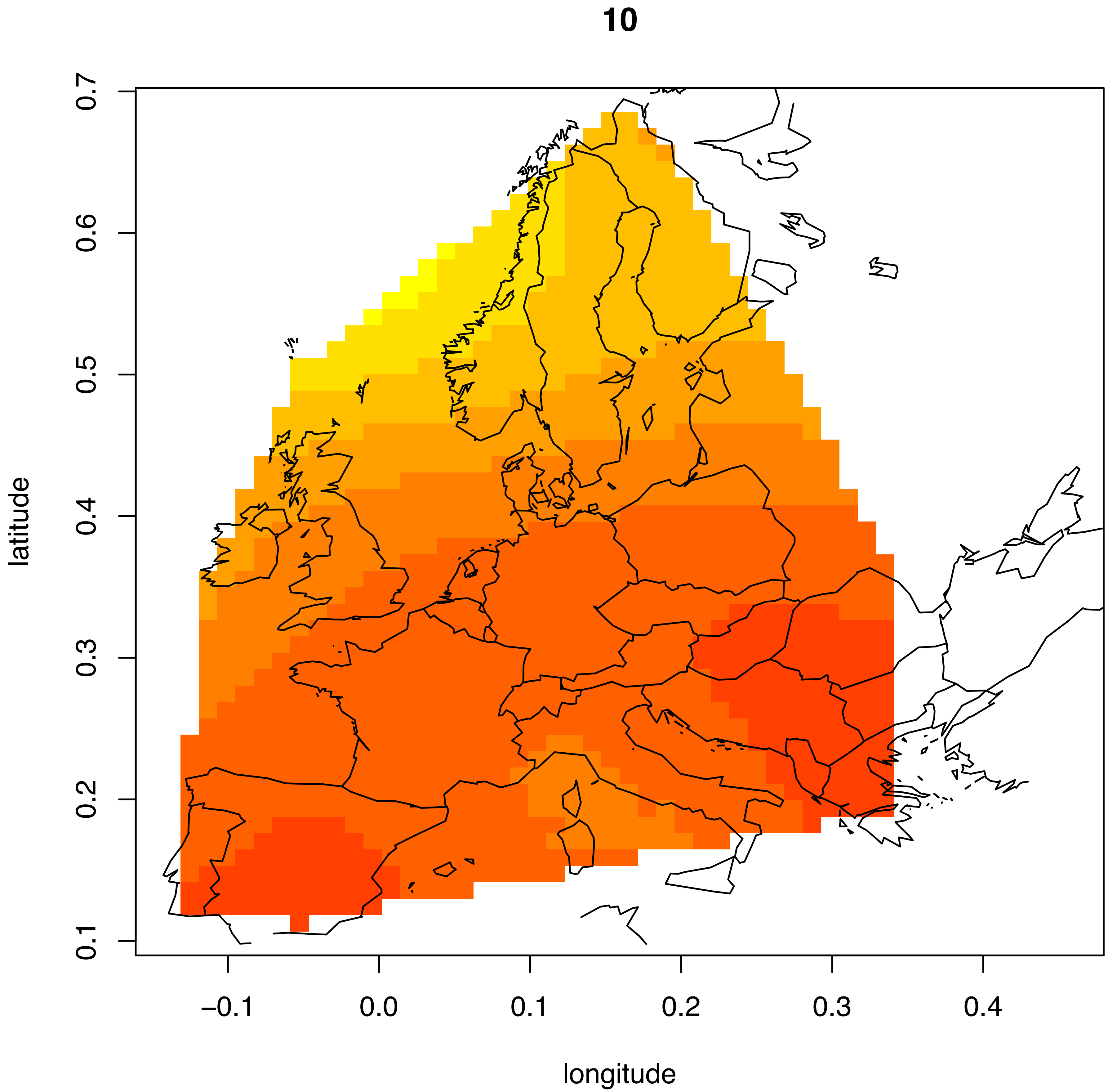
SO_2 pollution



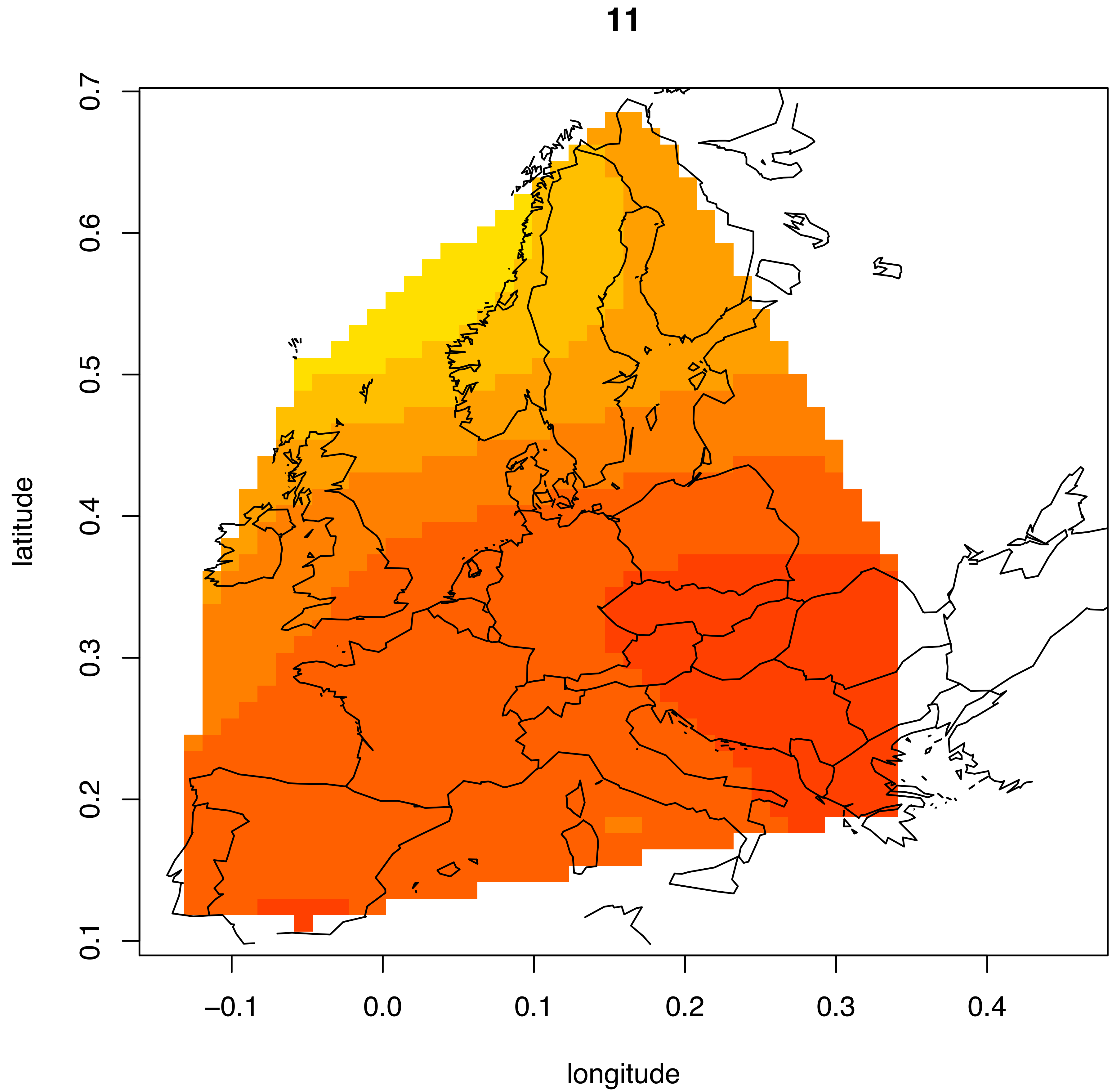
SO_2 pollution



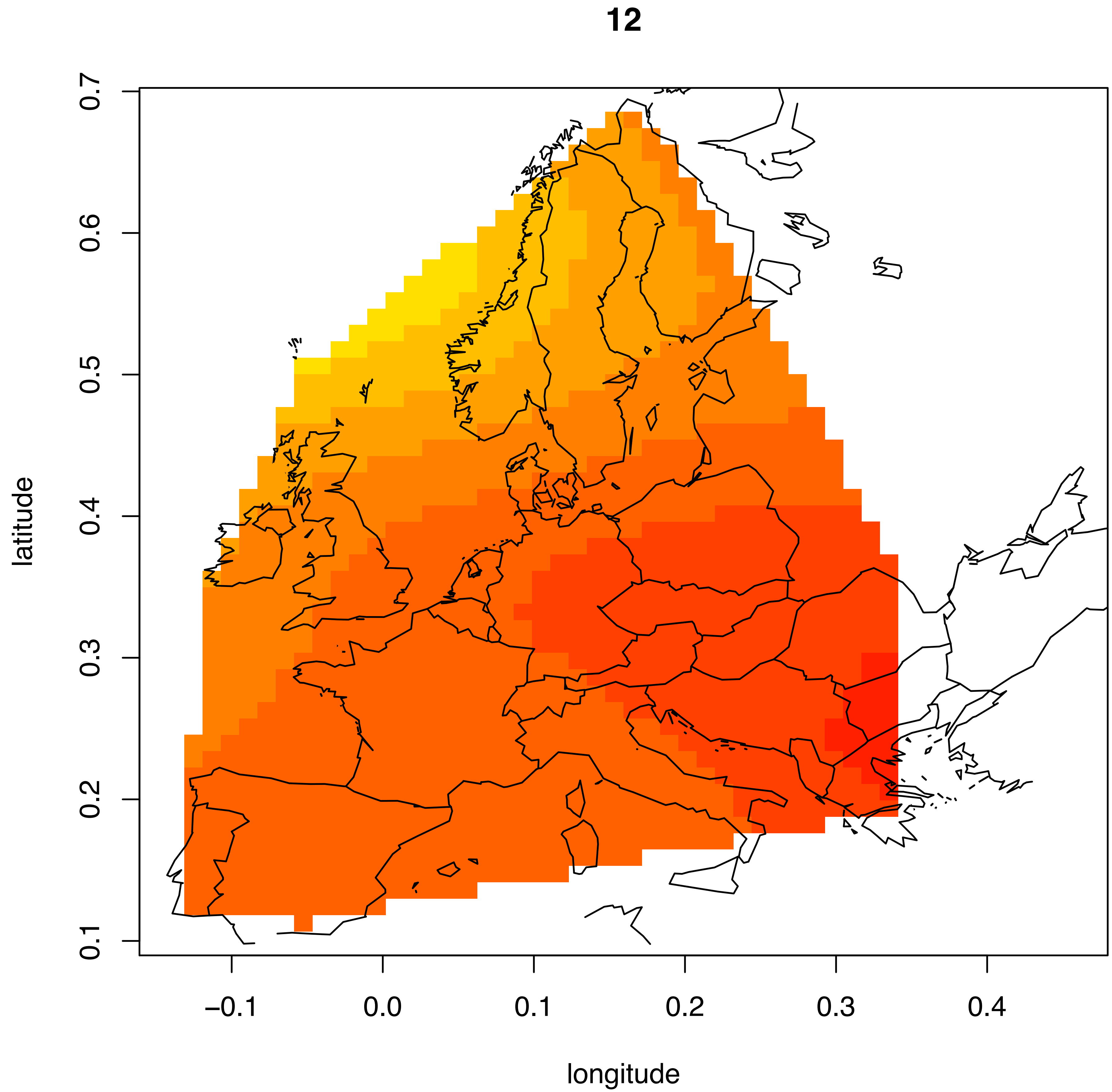
SO_2 pollution



SO_2 pollution



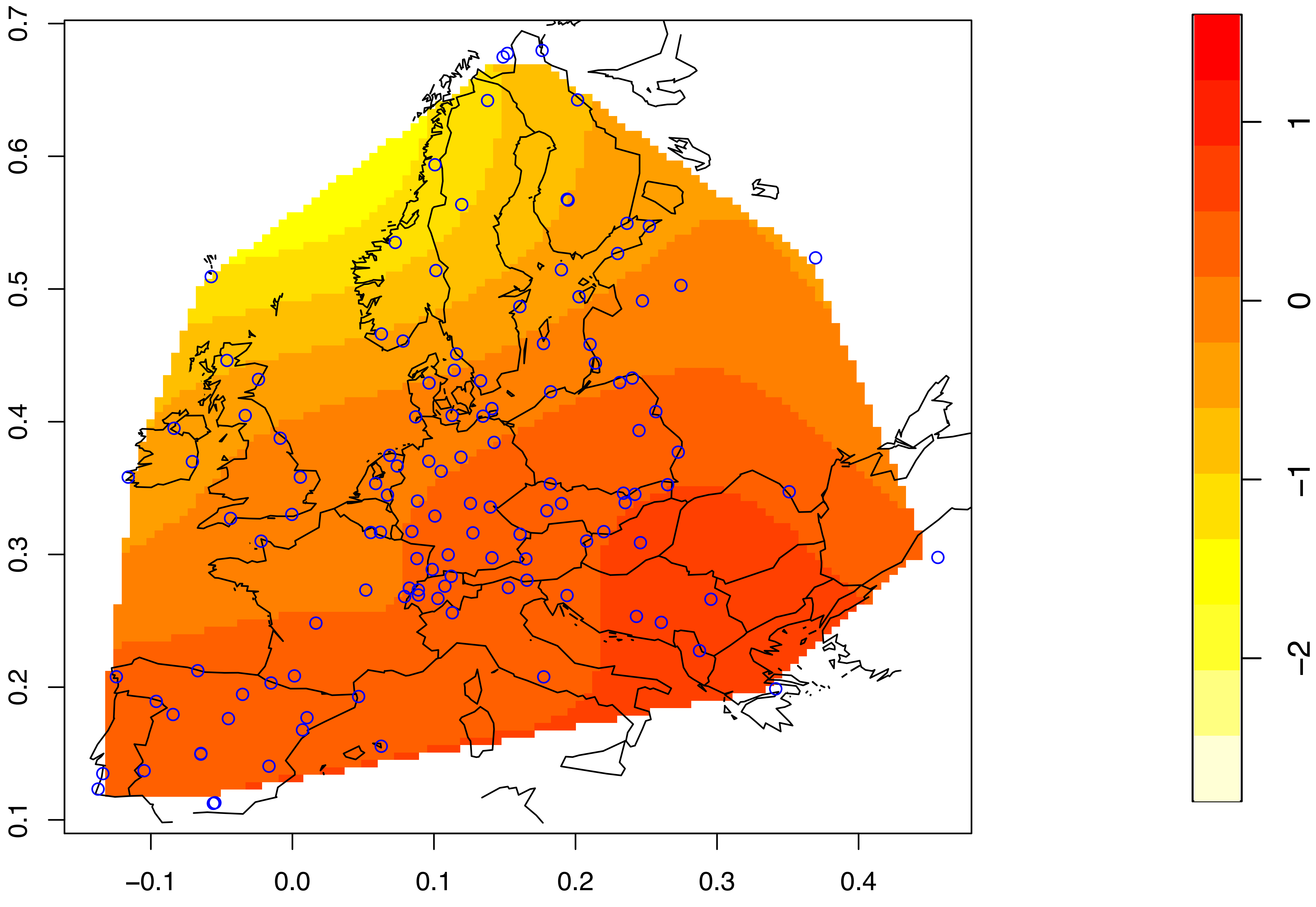
SO_2 pollution



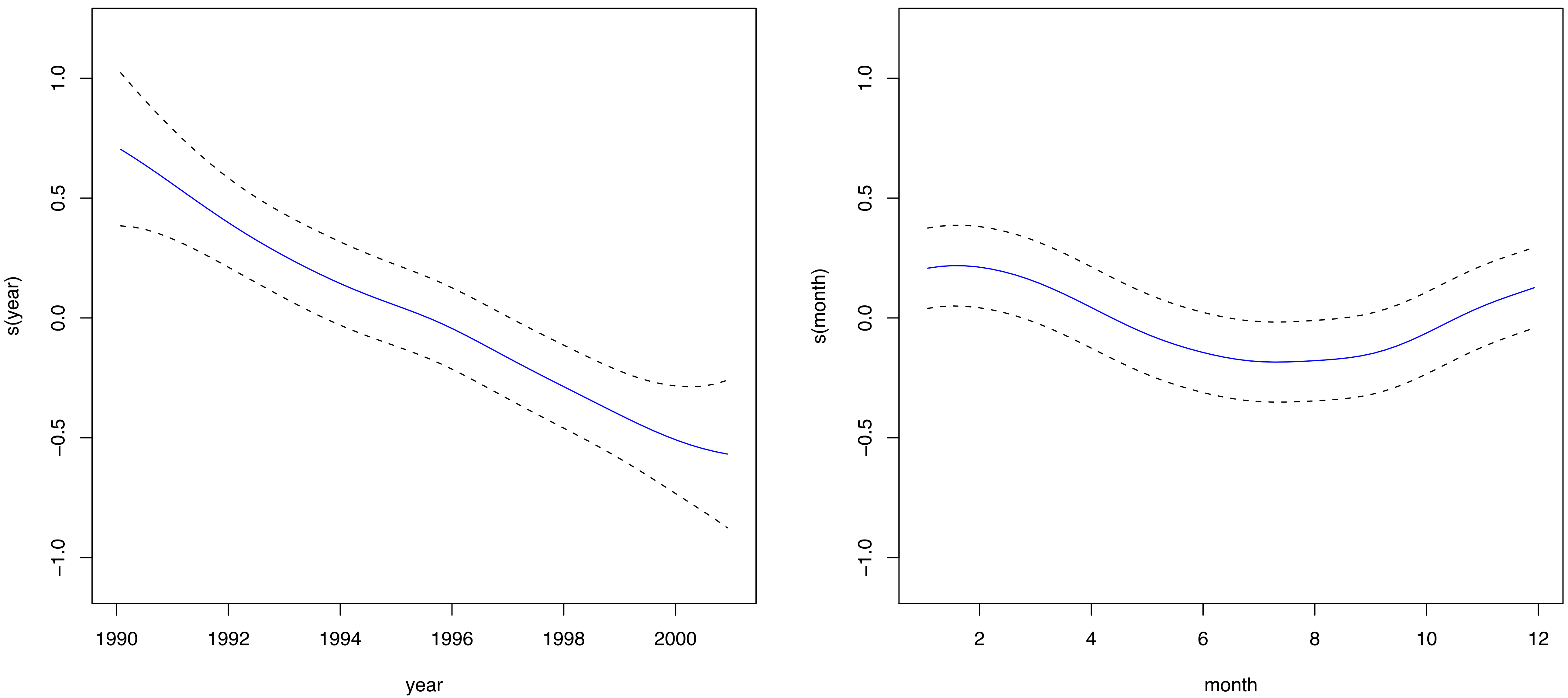
SO_2 pollution - interaction model

$$
\begin{aligned}
y = \quad & \mu \quad + m_s(x_1, x_2) + m_t(t) + m_z(z) + \\
& m_s(x_1, x_2) : m_t(t) + m_s(x_1, x_2) : m_z(z) + m_t(t) : m_z(z) + \varepsilon
\end{aligned}
$$

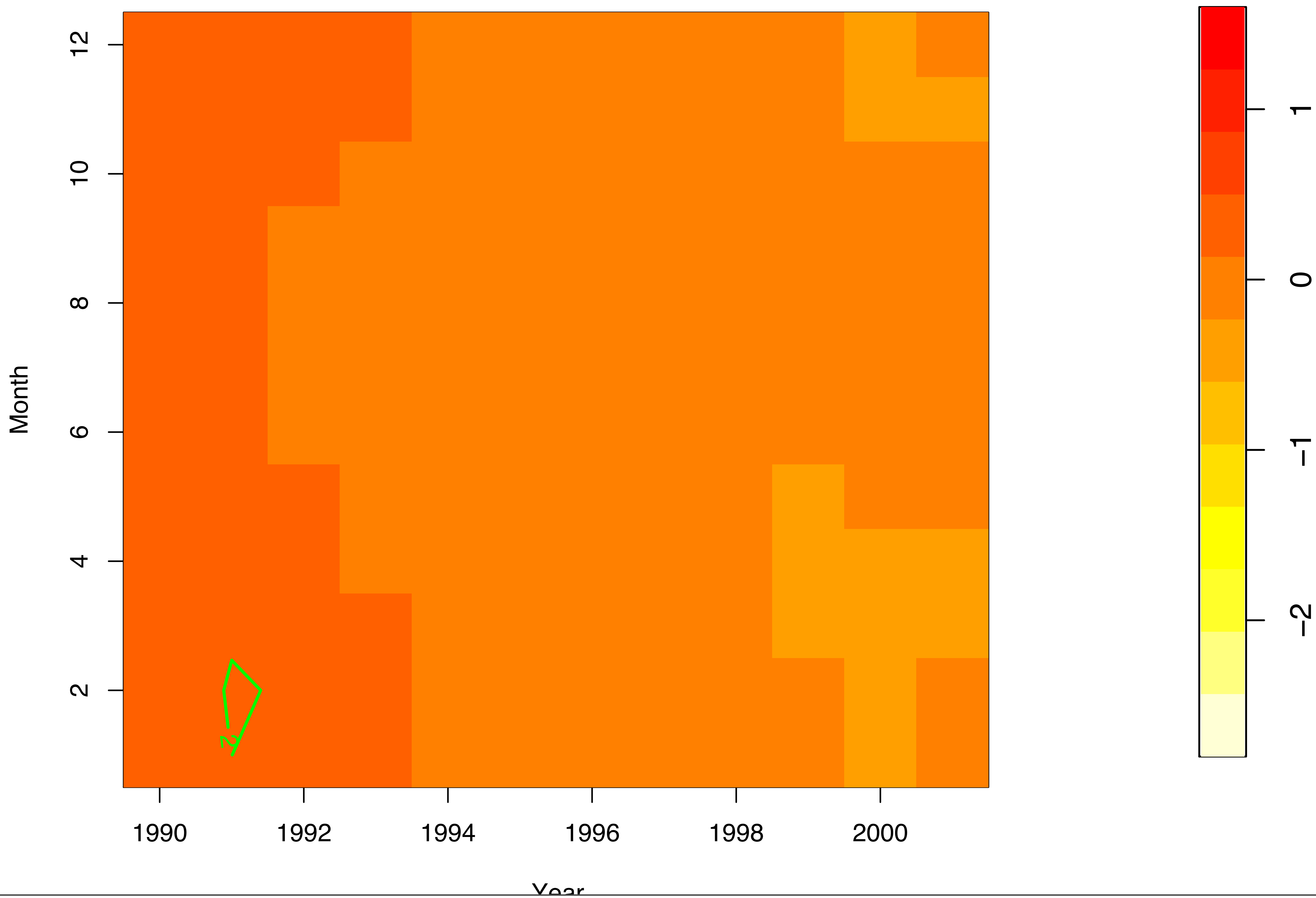
SO_2 pollution - additive model



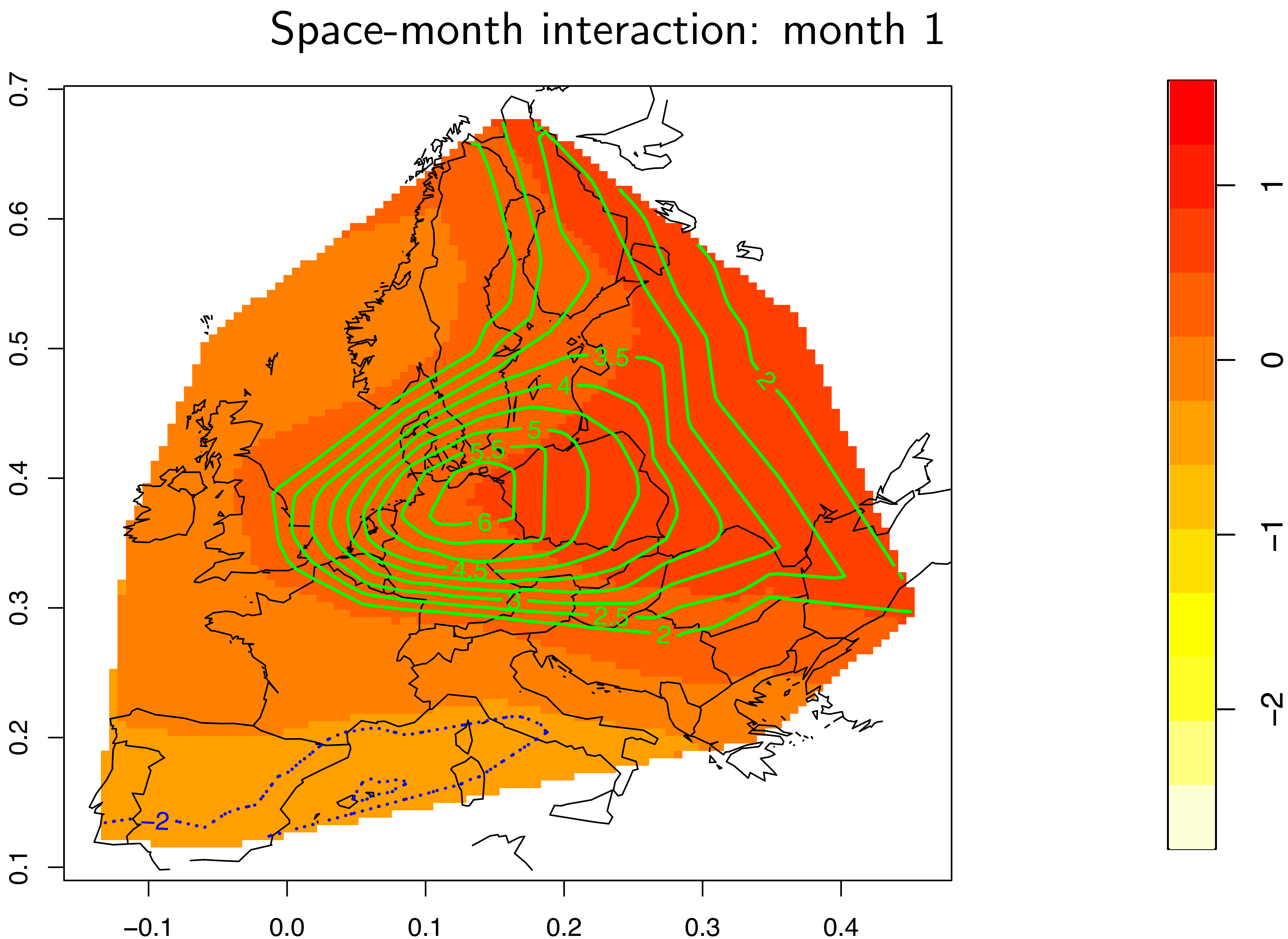
SO_2 pollution - additive model



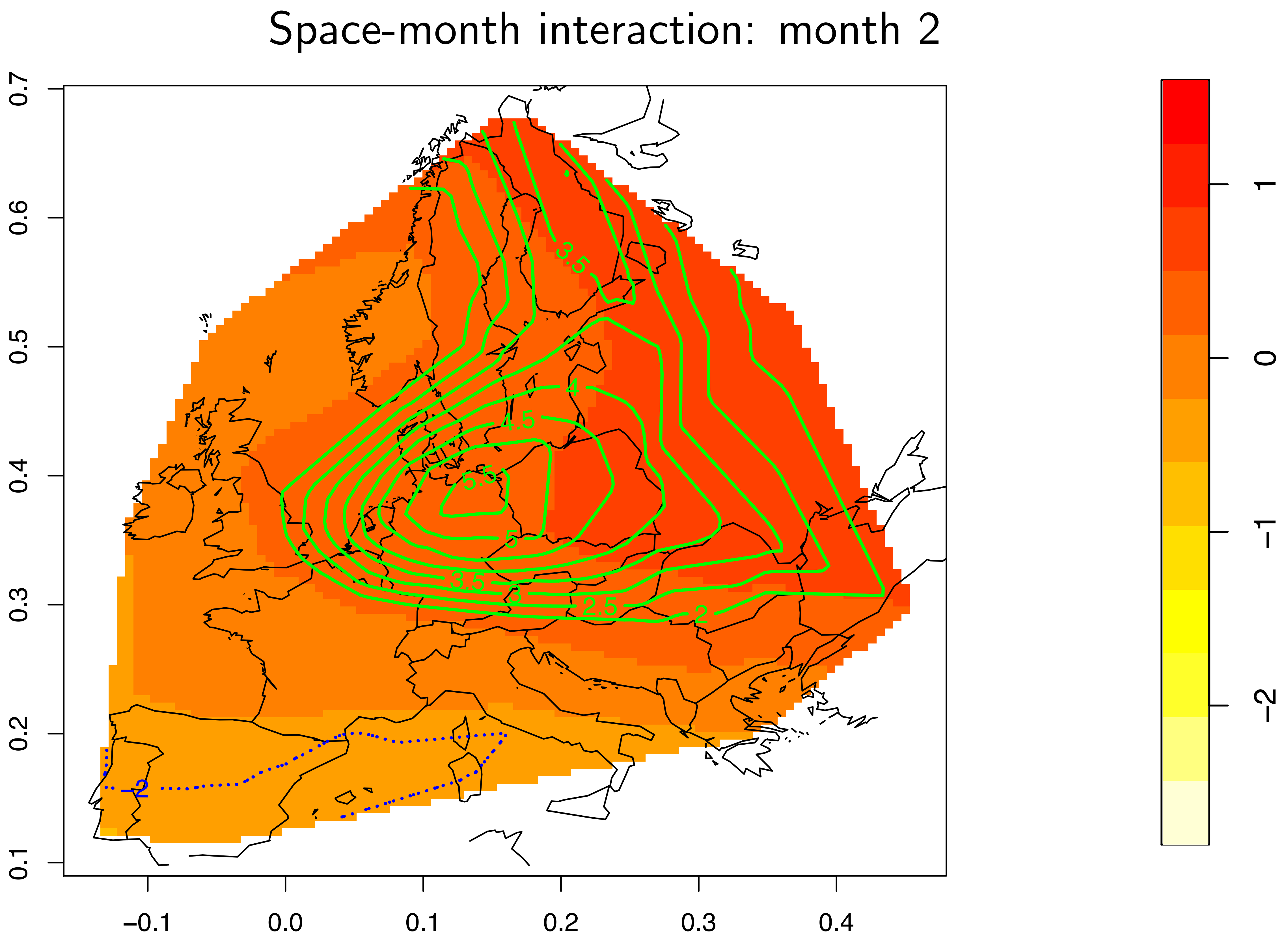
SO_2 pollution



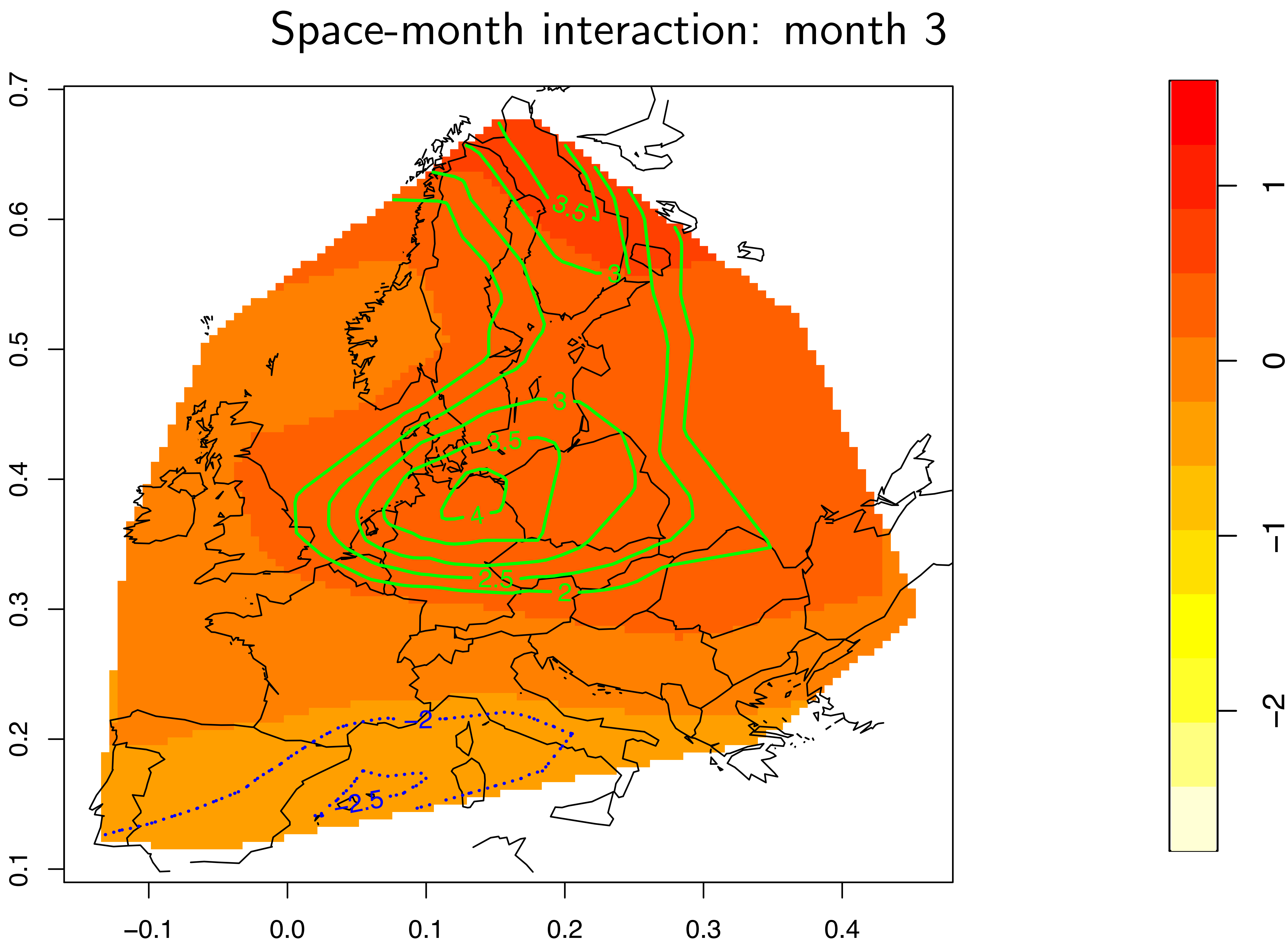
SO_2 pollution



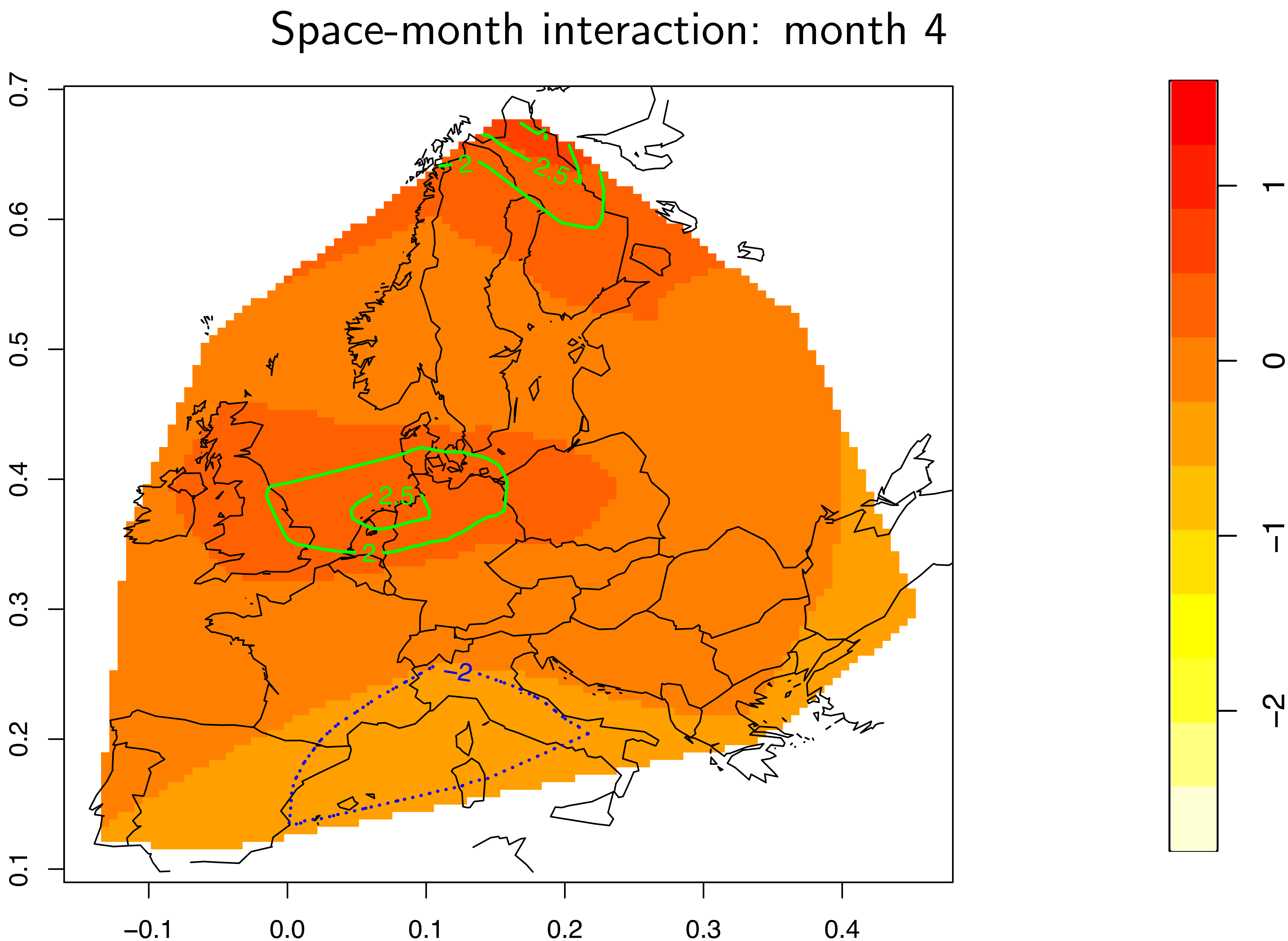
SO_2 pollution



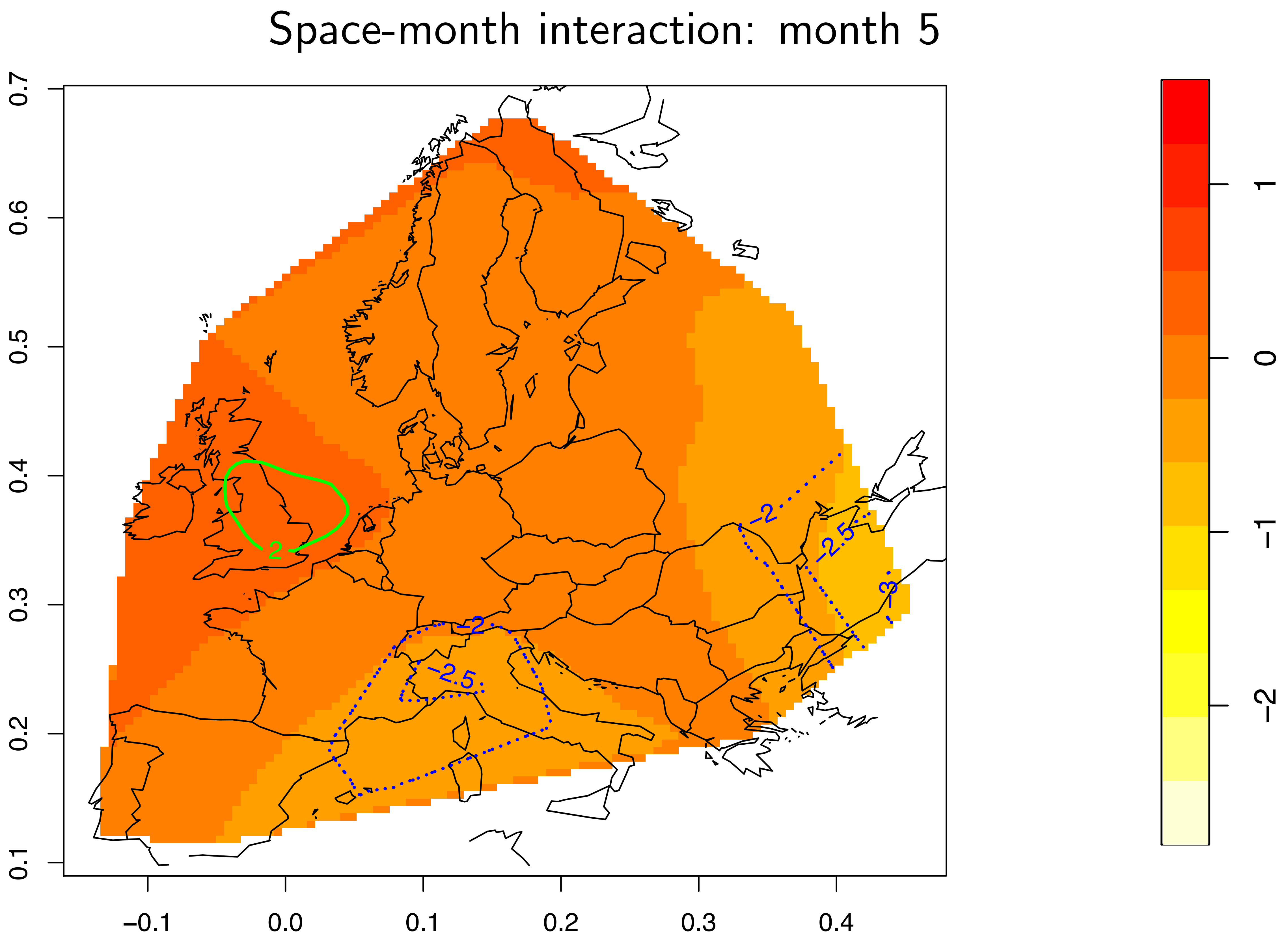
SO_2 pollution



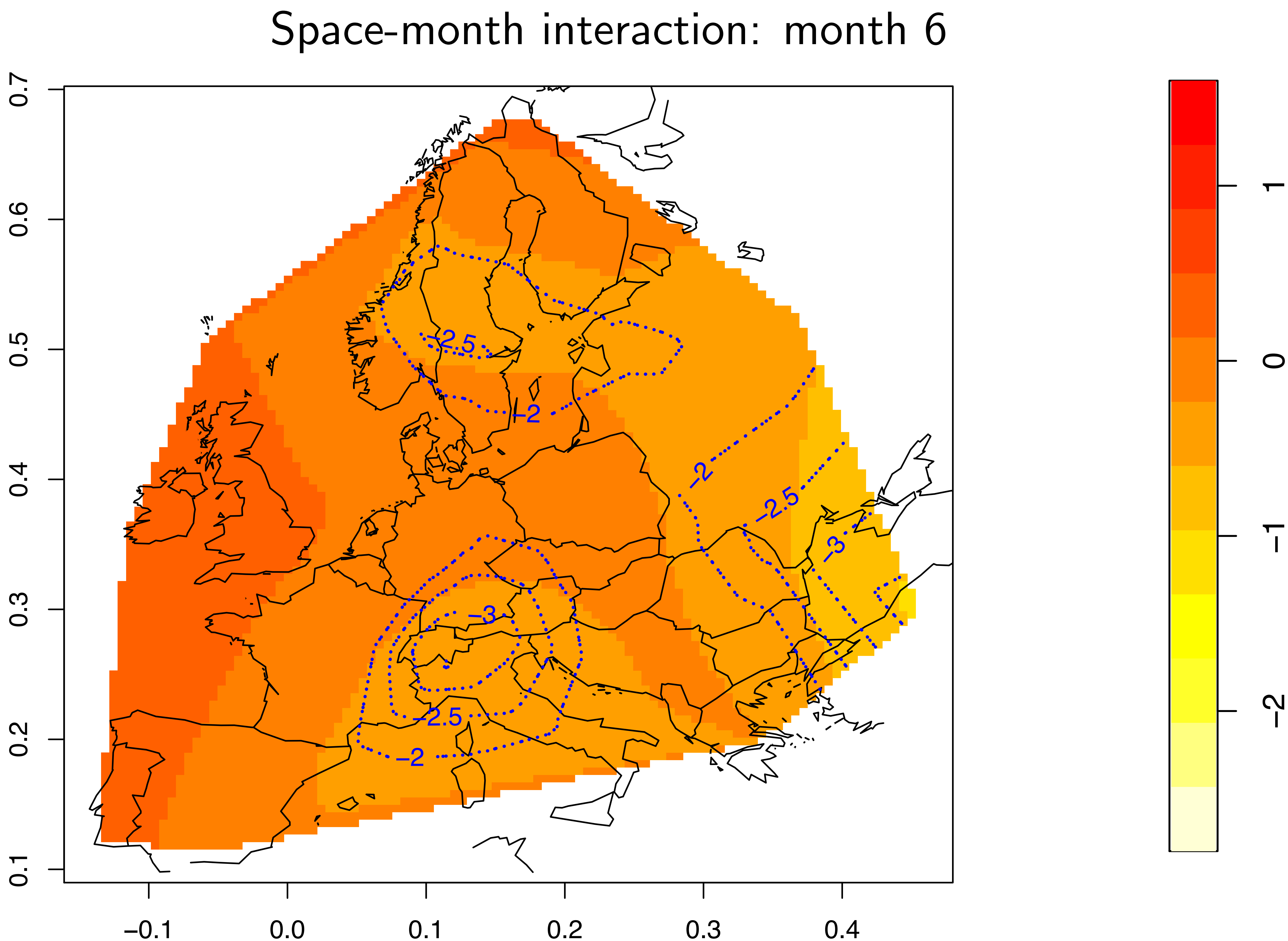
SO_2 pollution



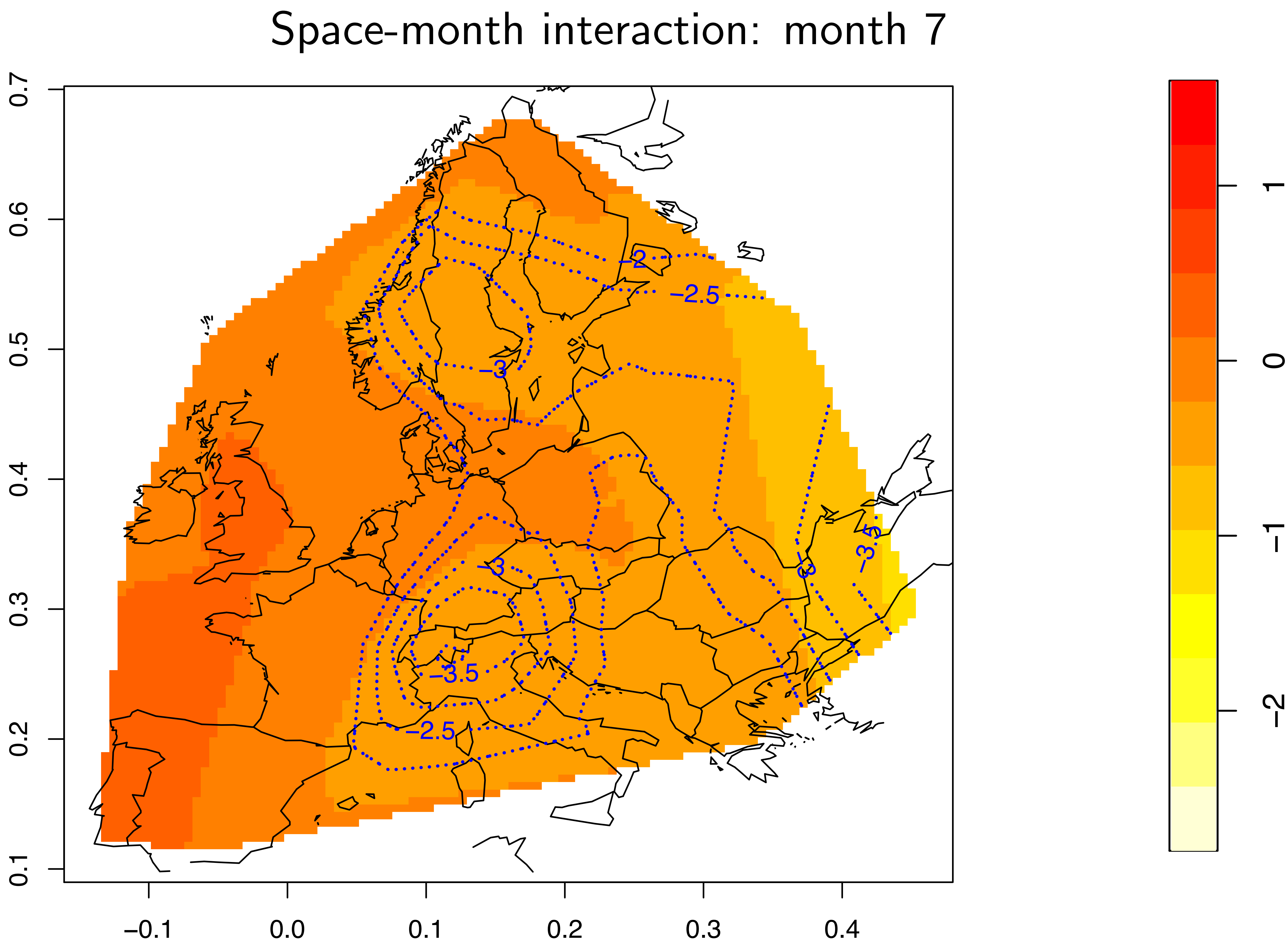
SO_2 pollution



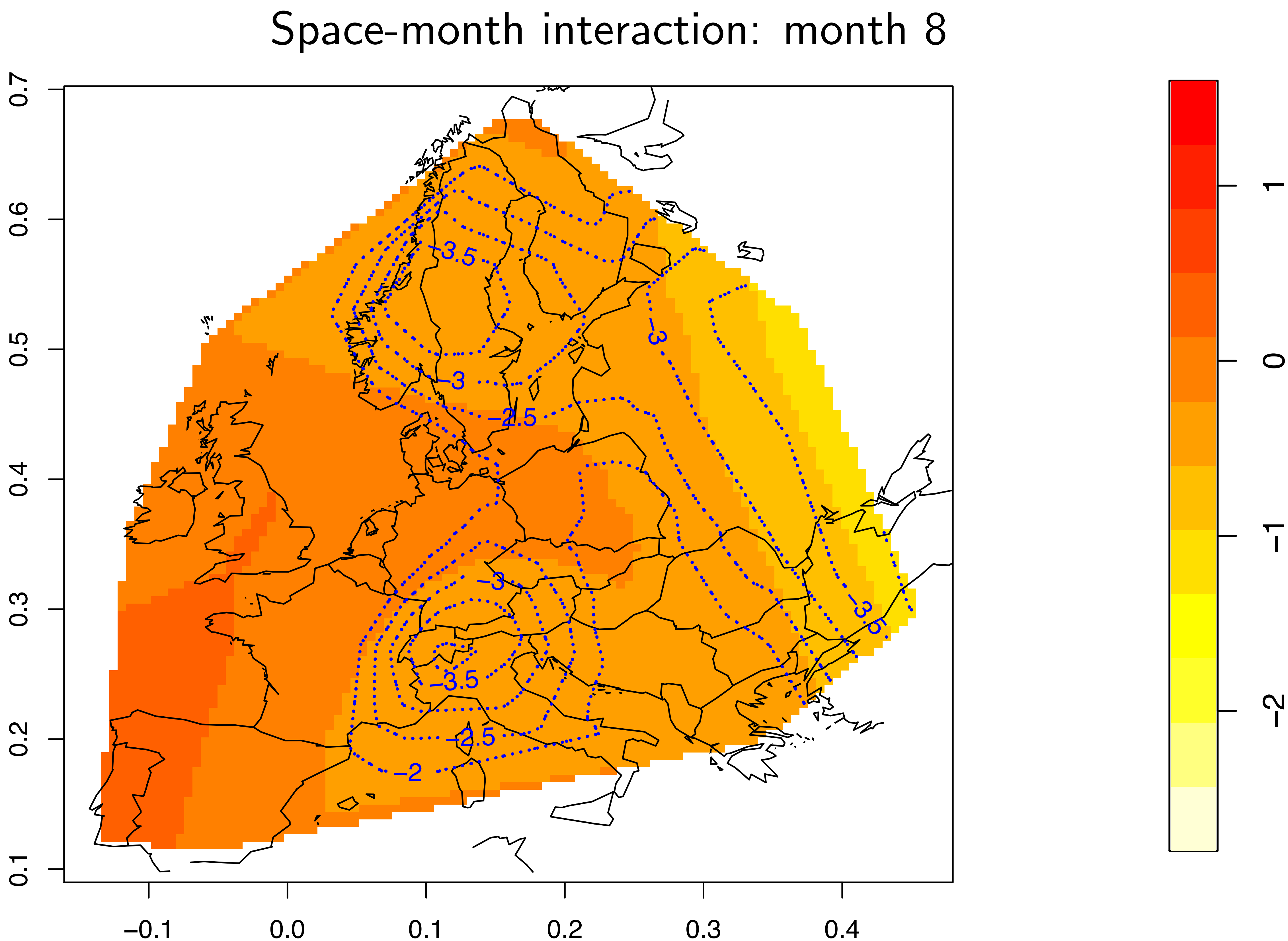
SO_2 pollution



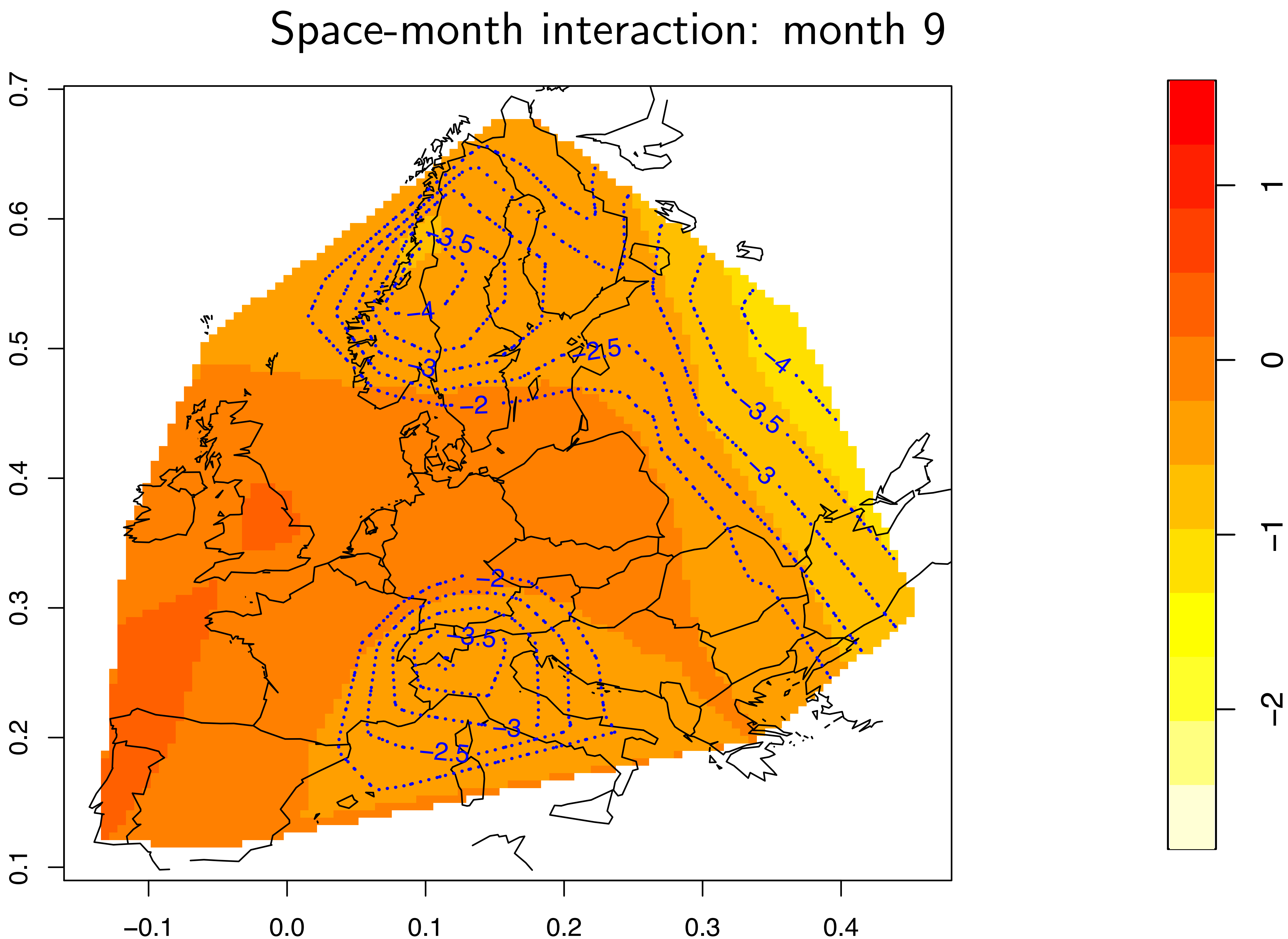
SO_2 pollution



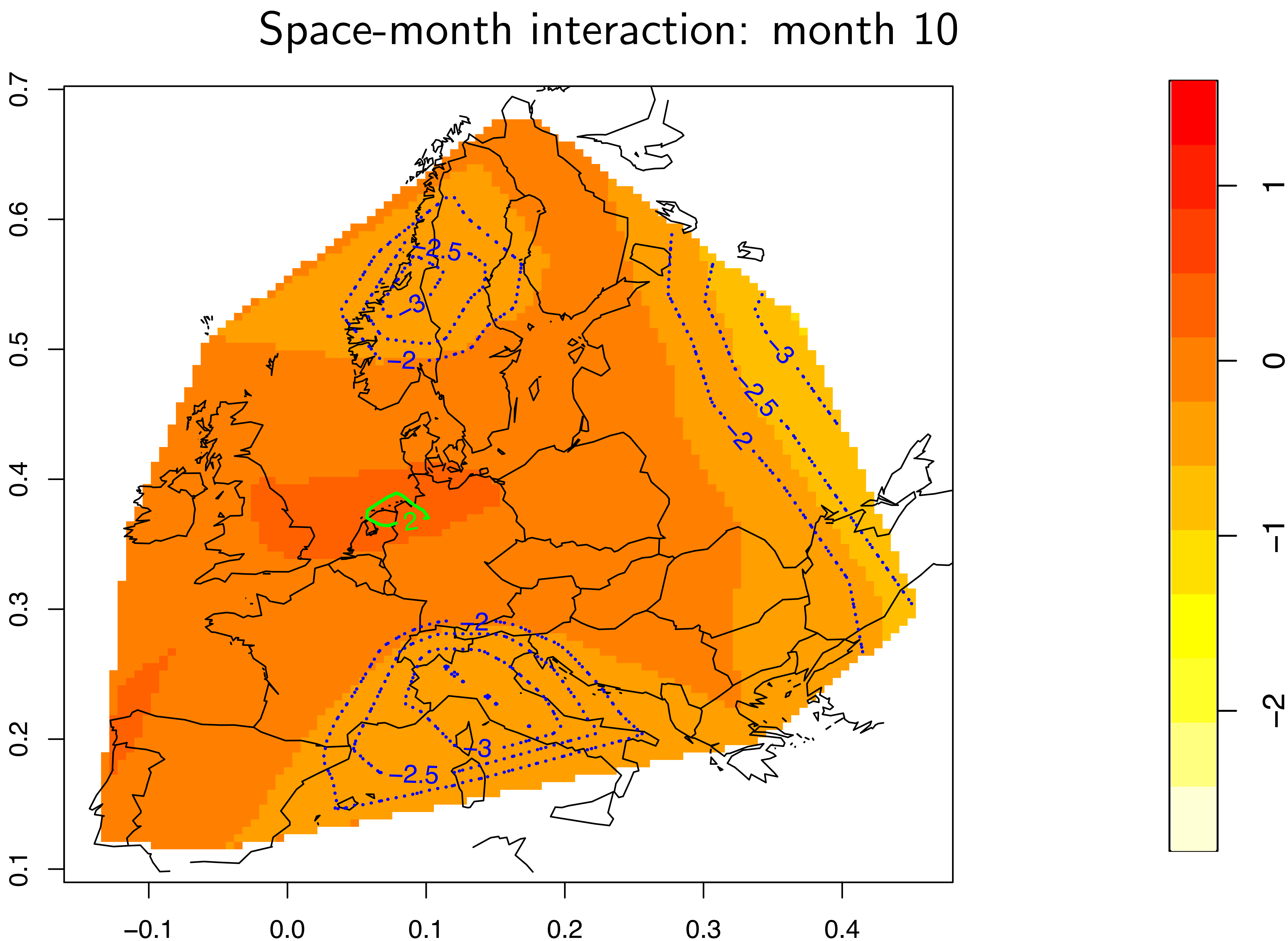
SO_2 pollution



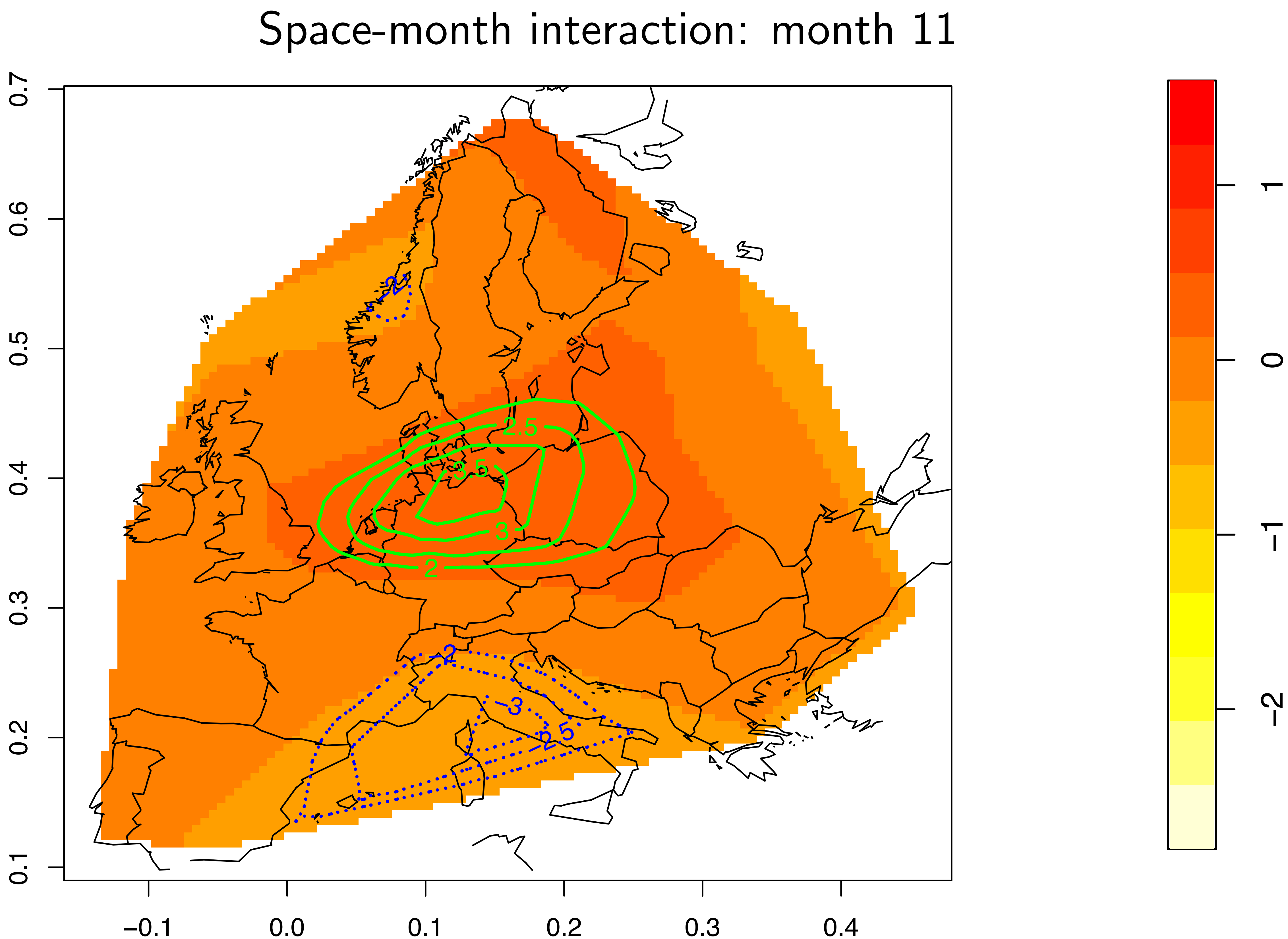
SO_2 pollution



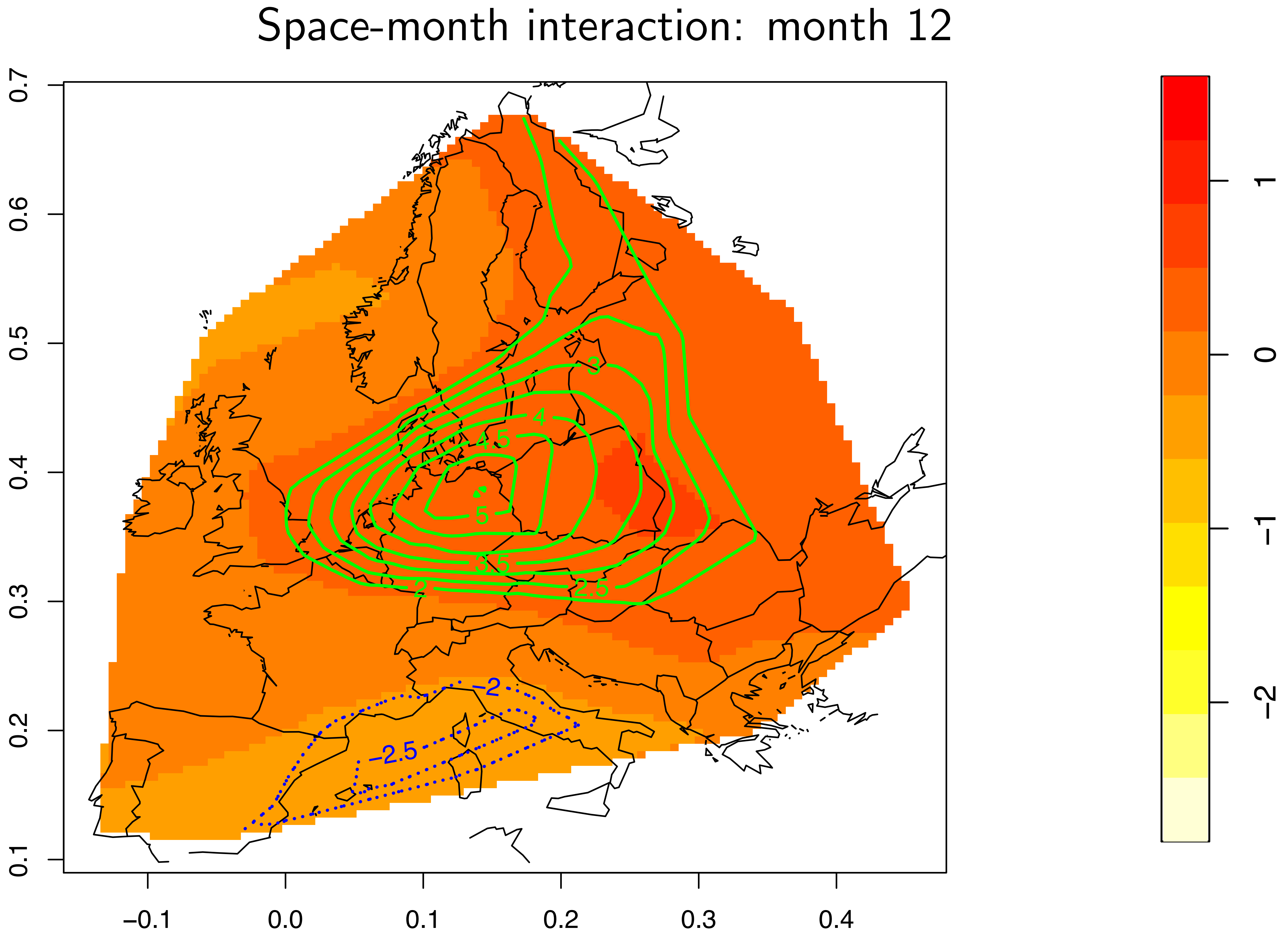
SO_2 pollution



SO_2 pollution

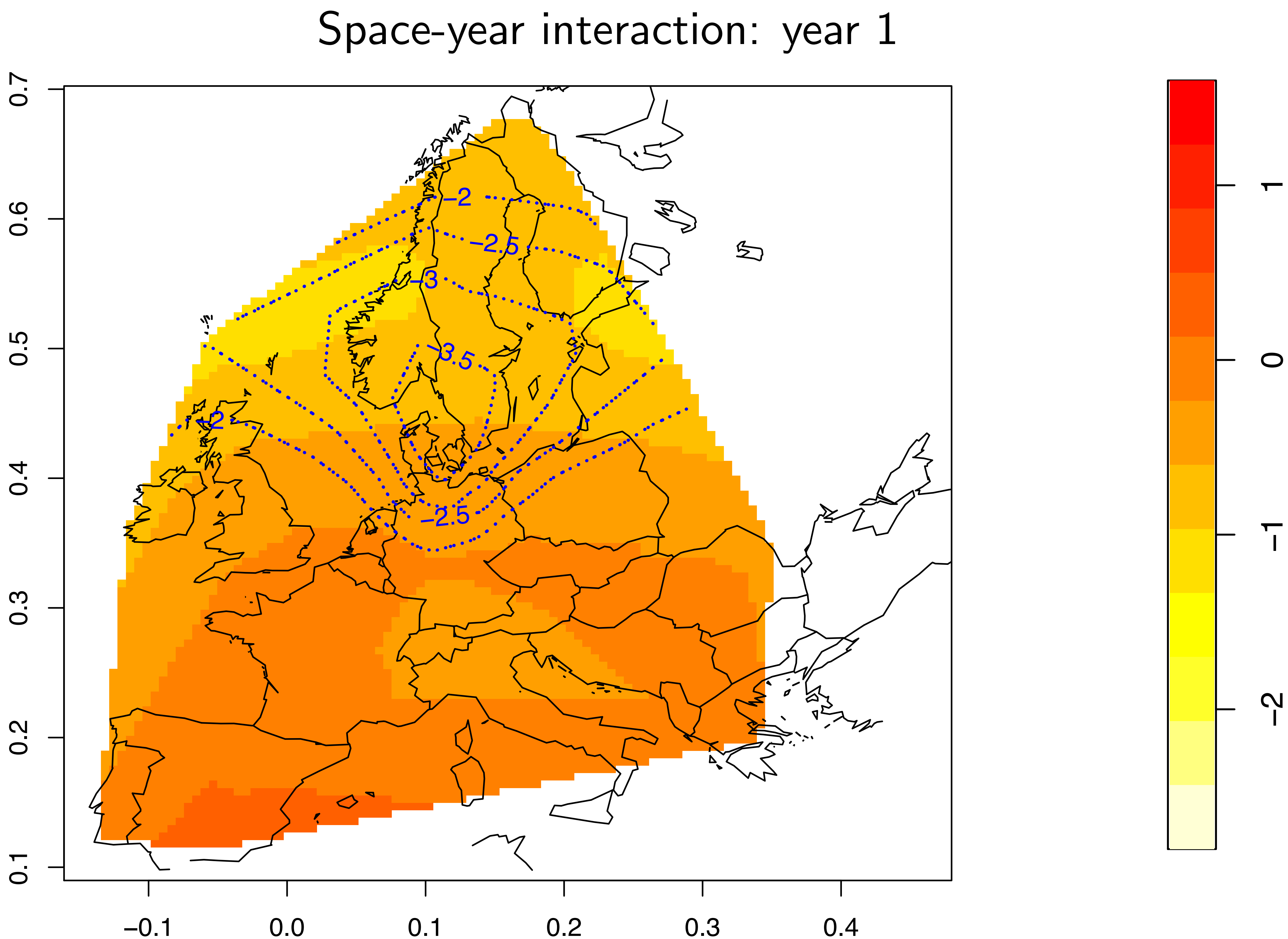


SO_2 pollution

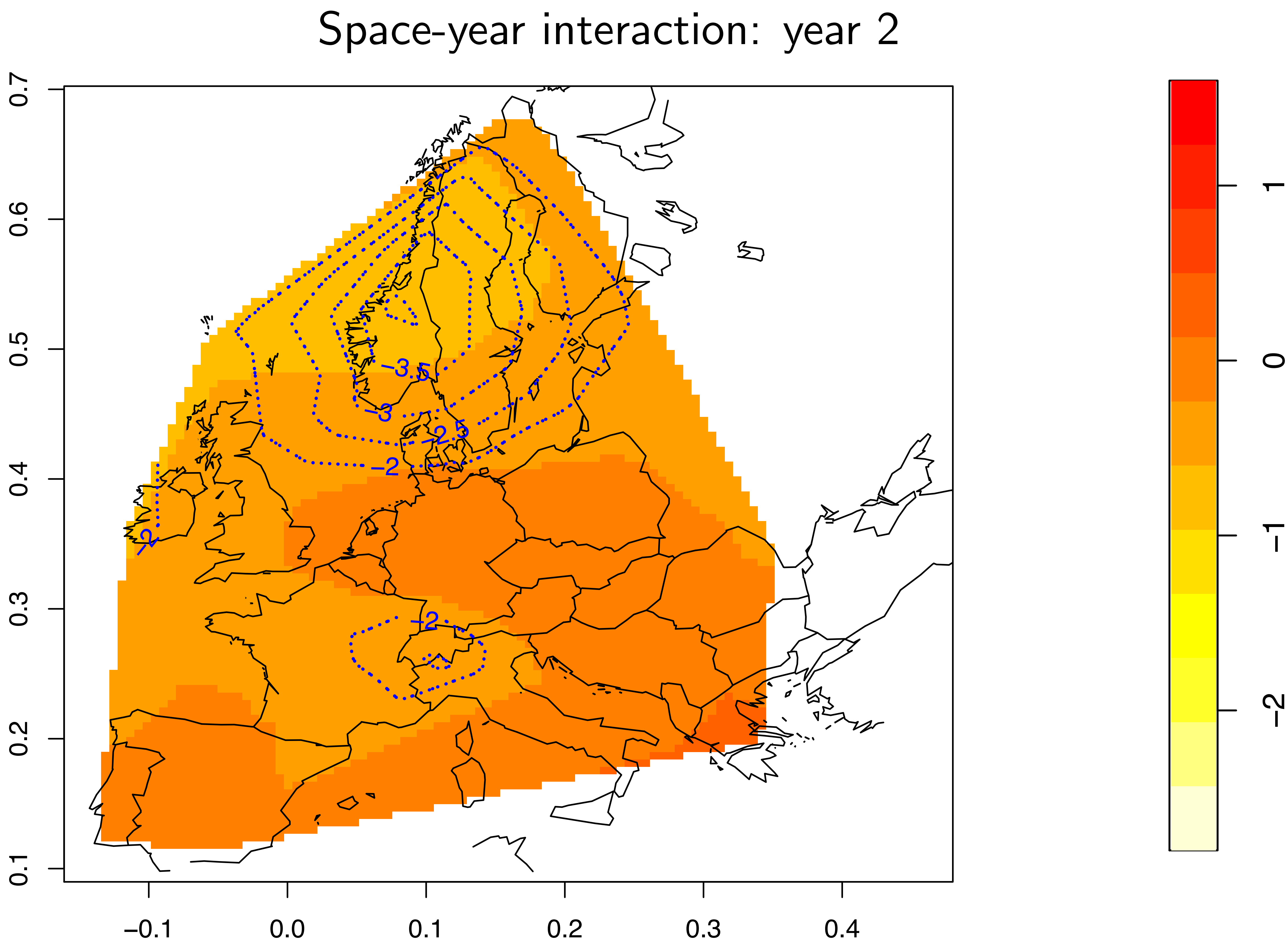


SO_2 pollution

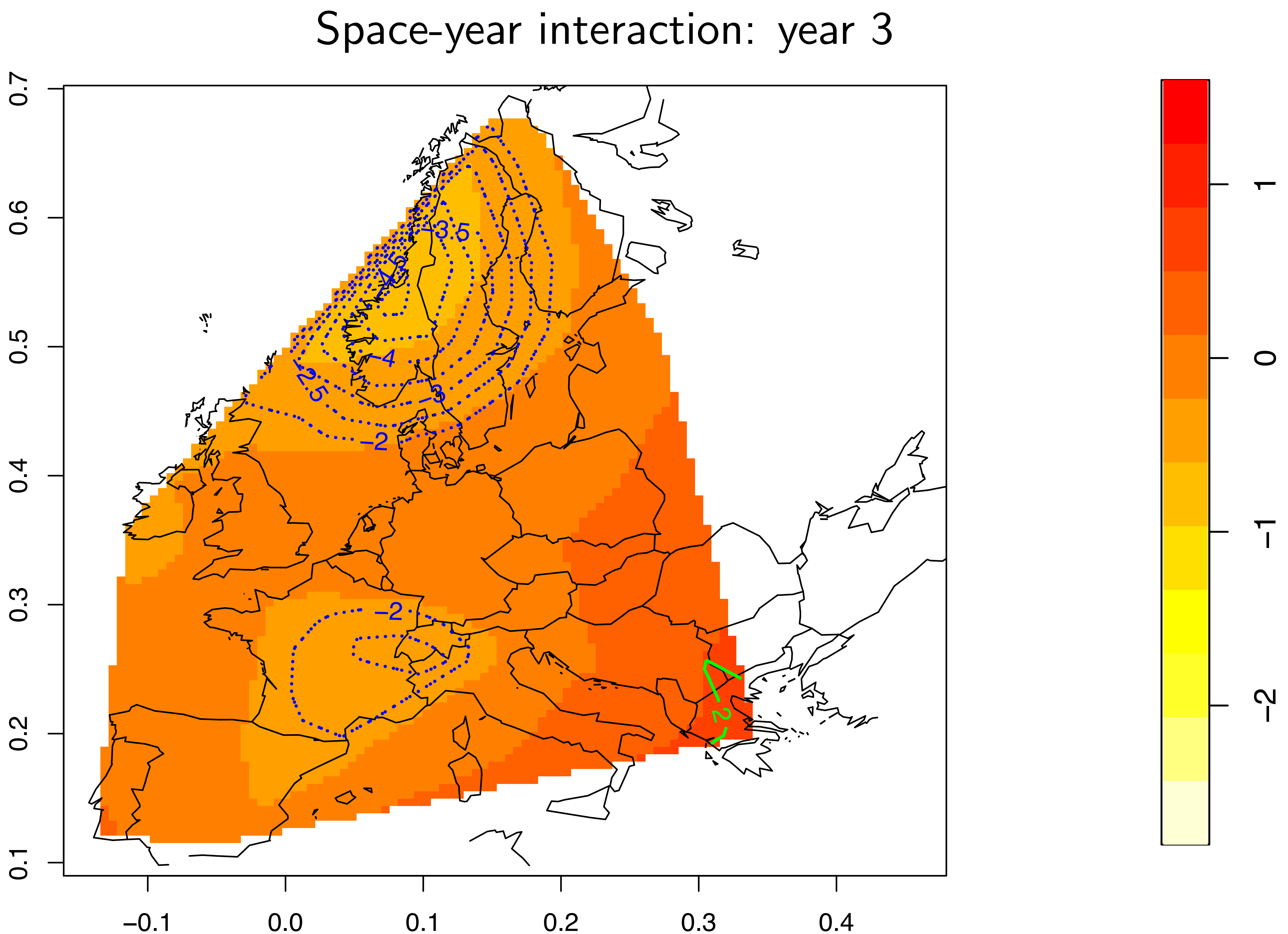
SO_2 pollution



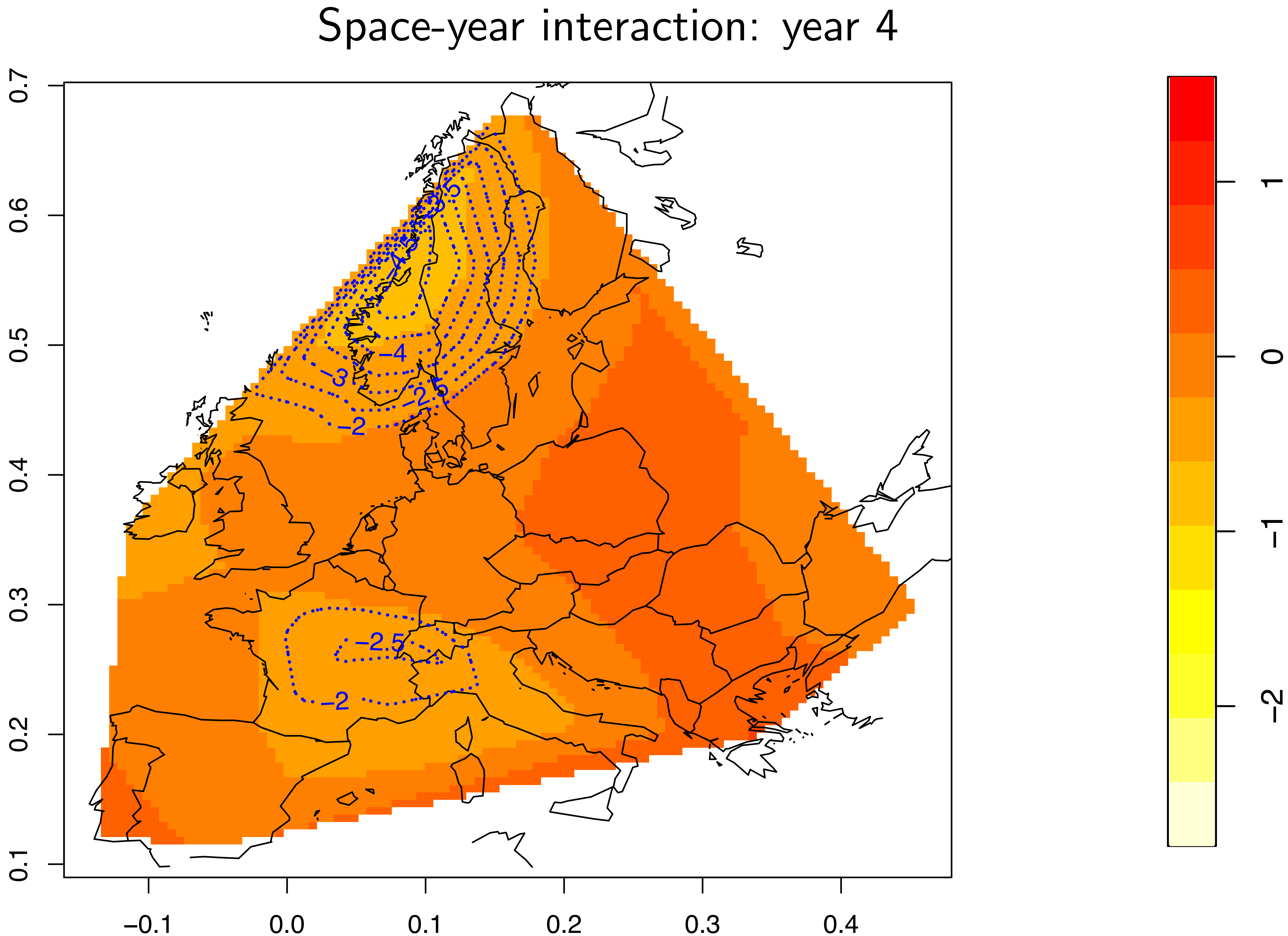
SO_2 pollution



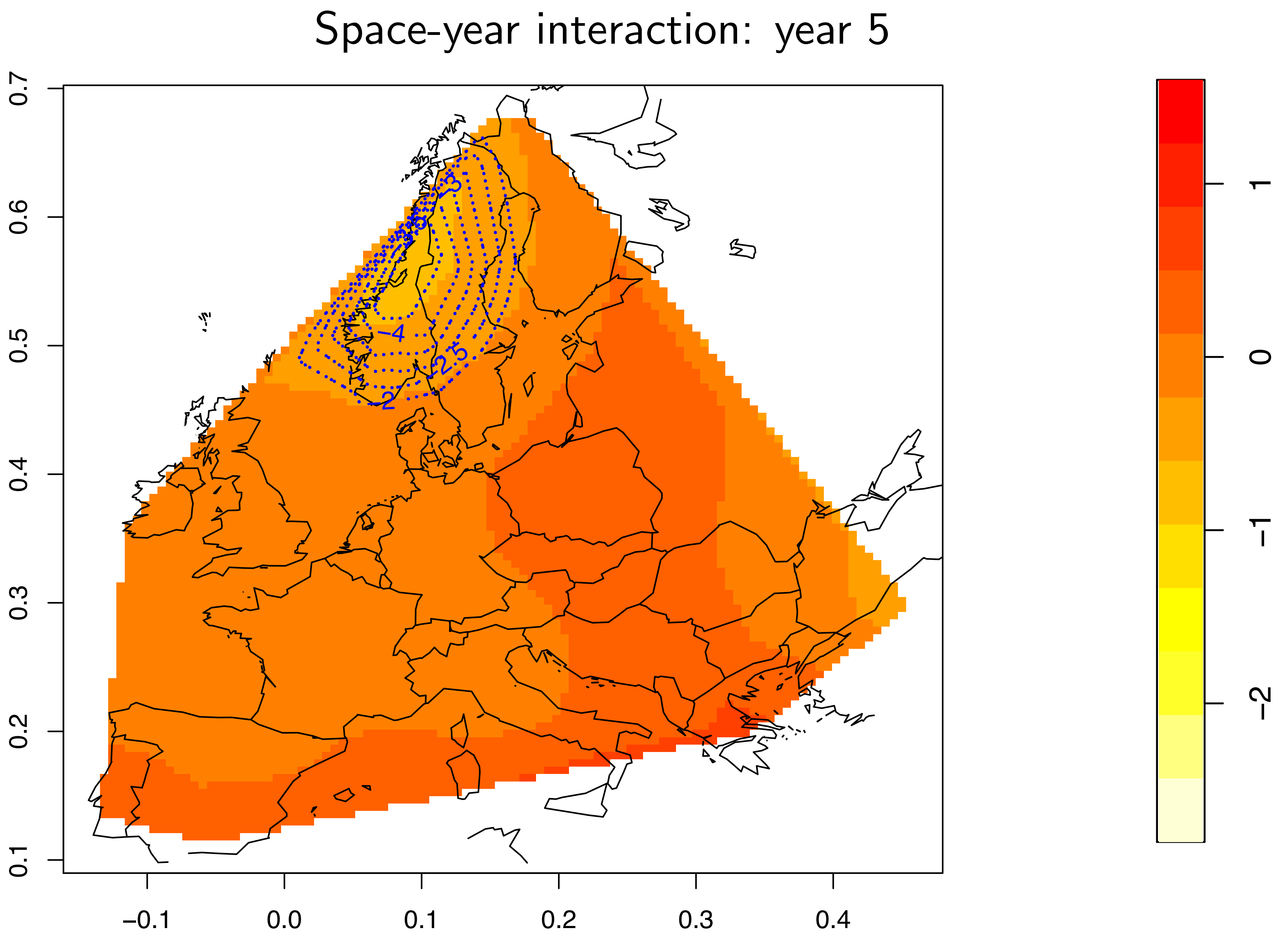
SO_2 pollution



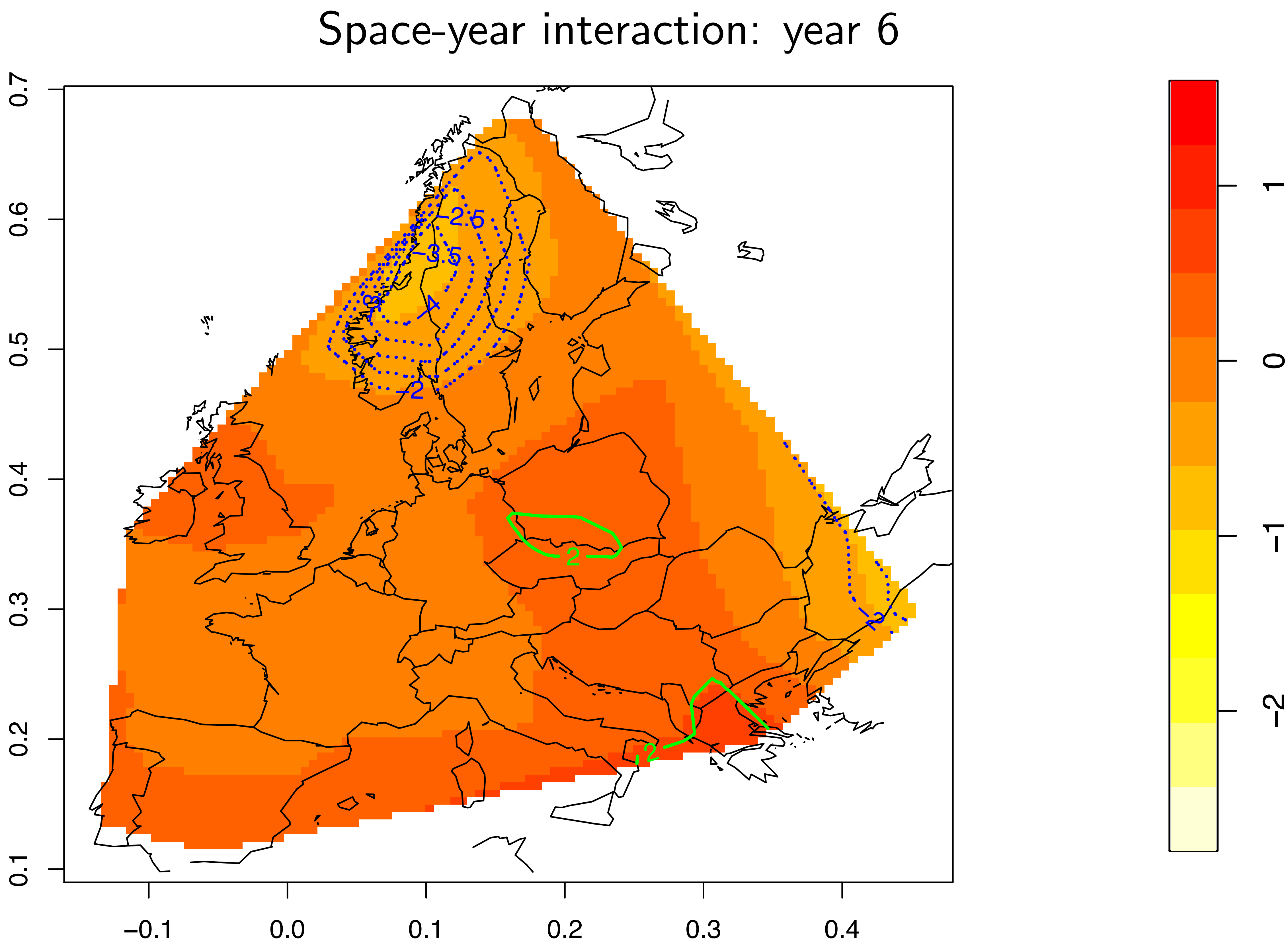
SO_2 pollution



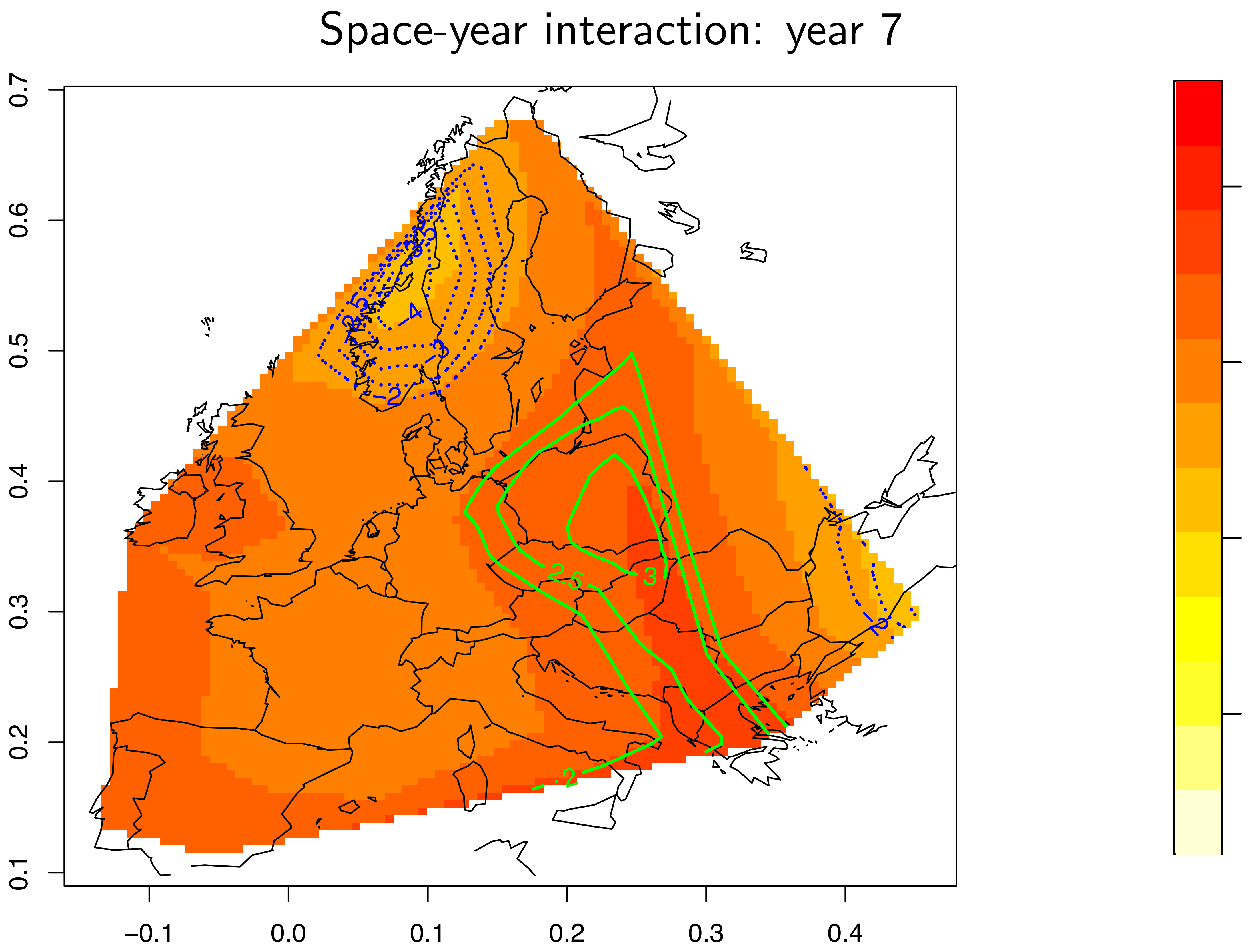
SO_2 pollution



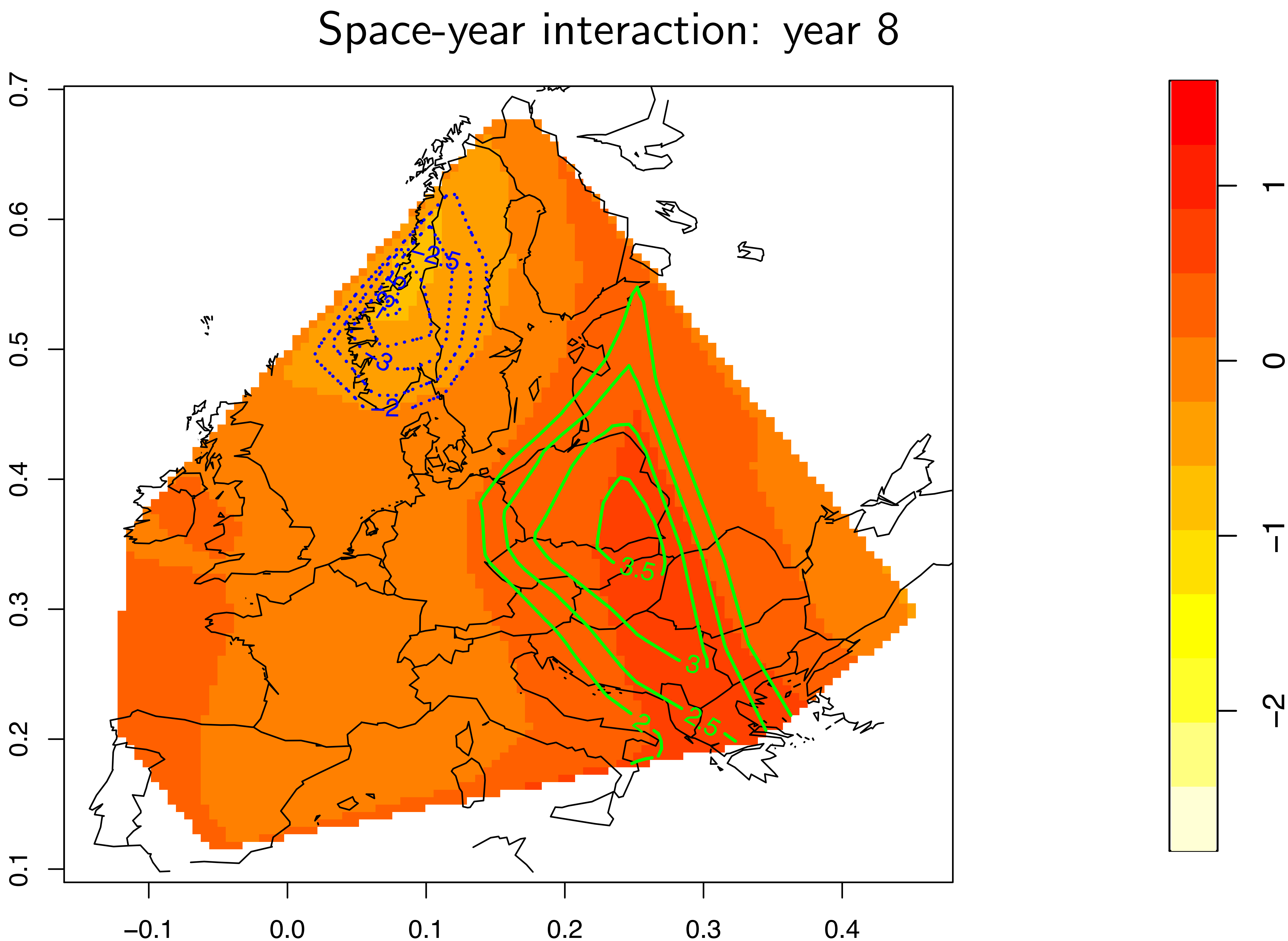
SO_2 pollution



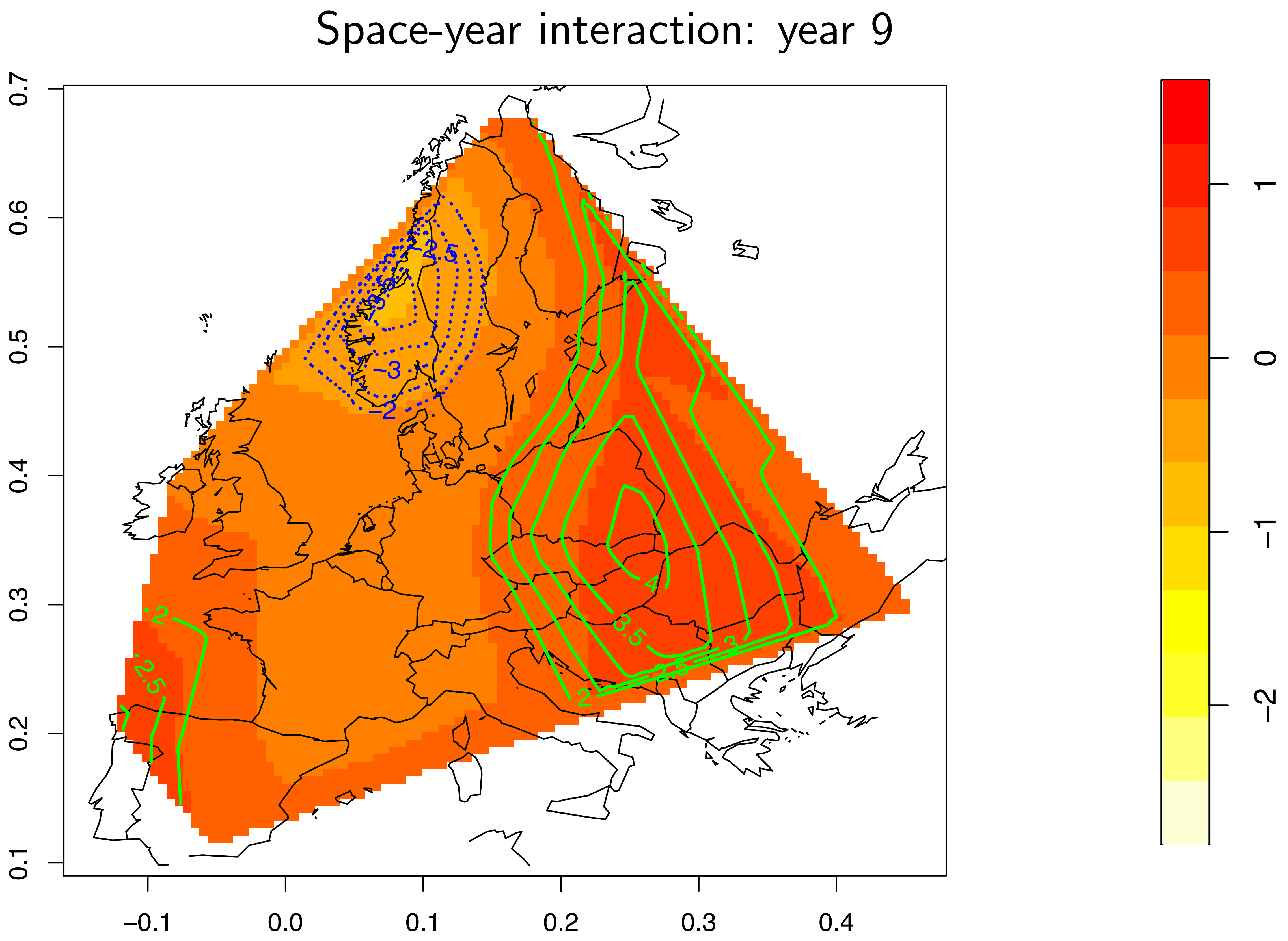
SO_2 pollution



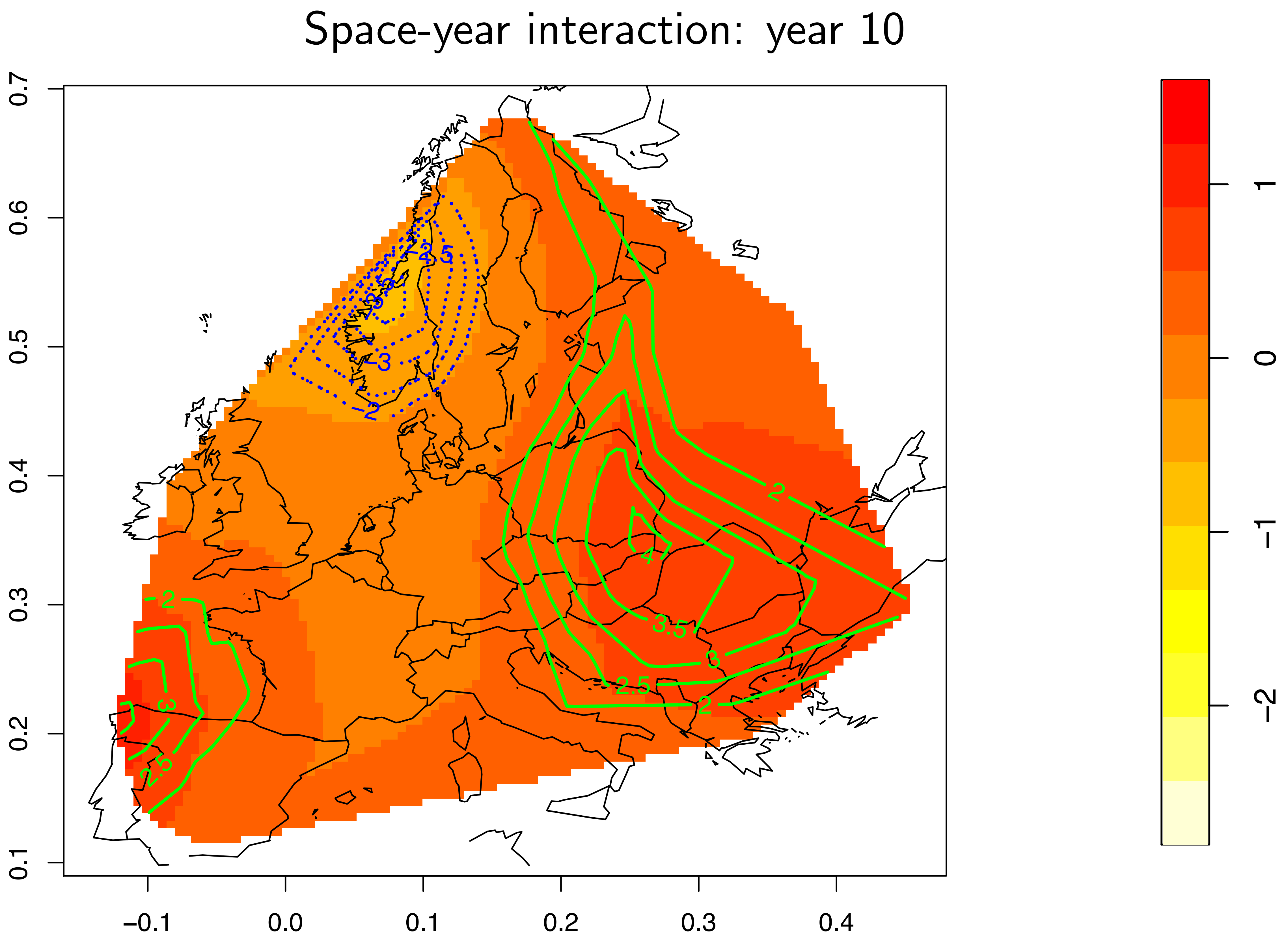
SO_2 pollution



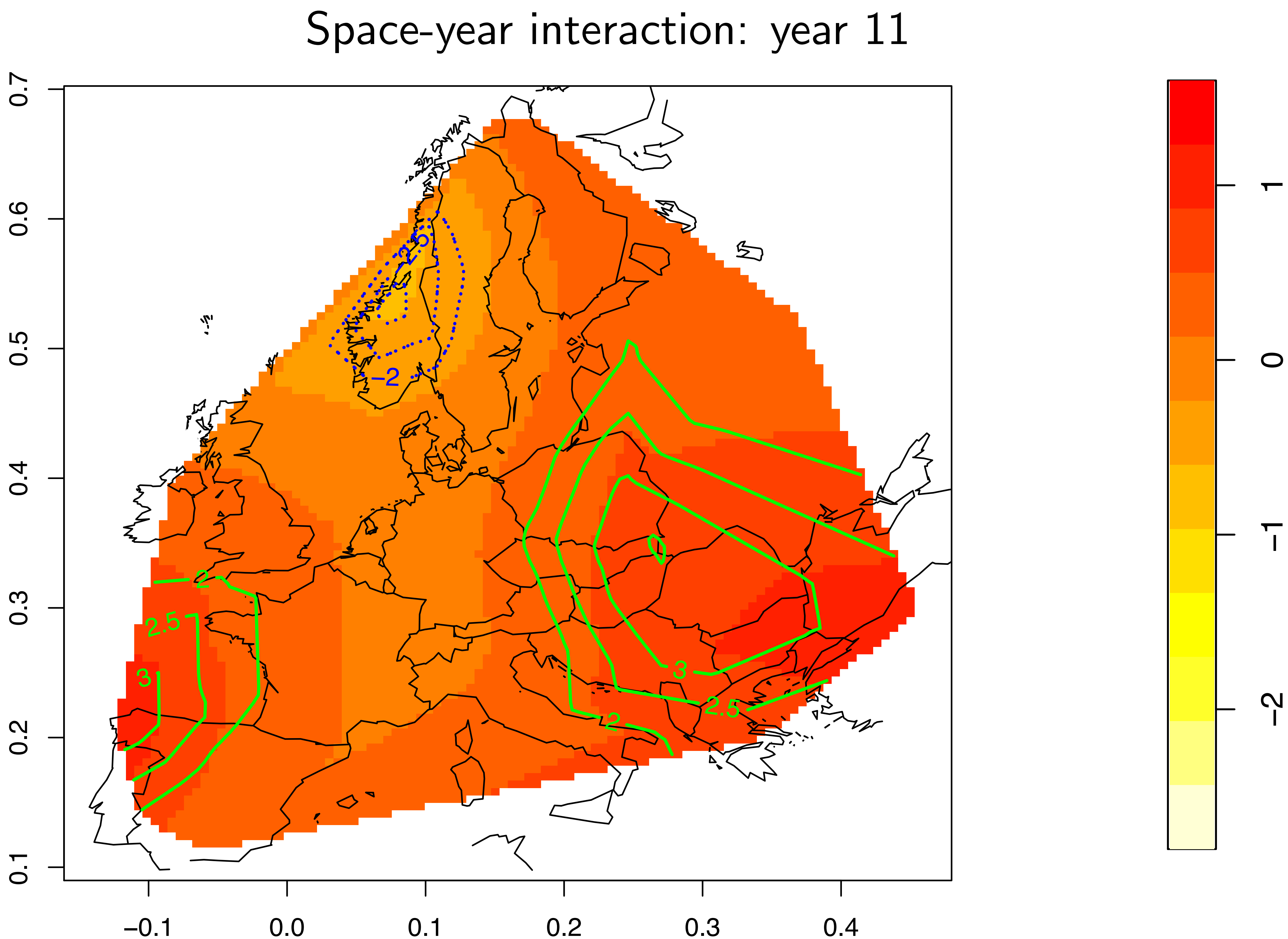
SO_2 pollution



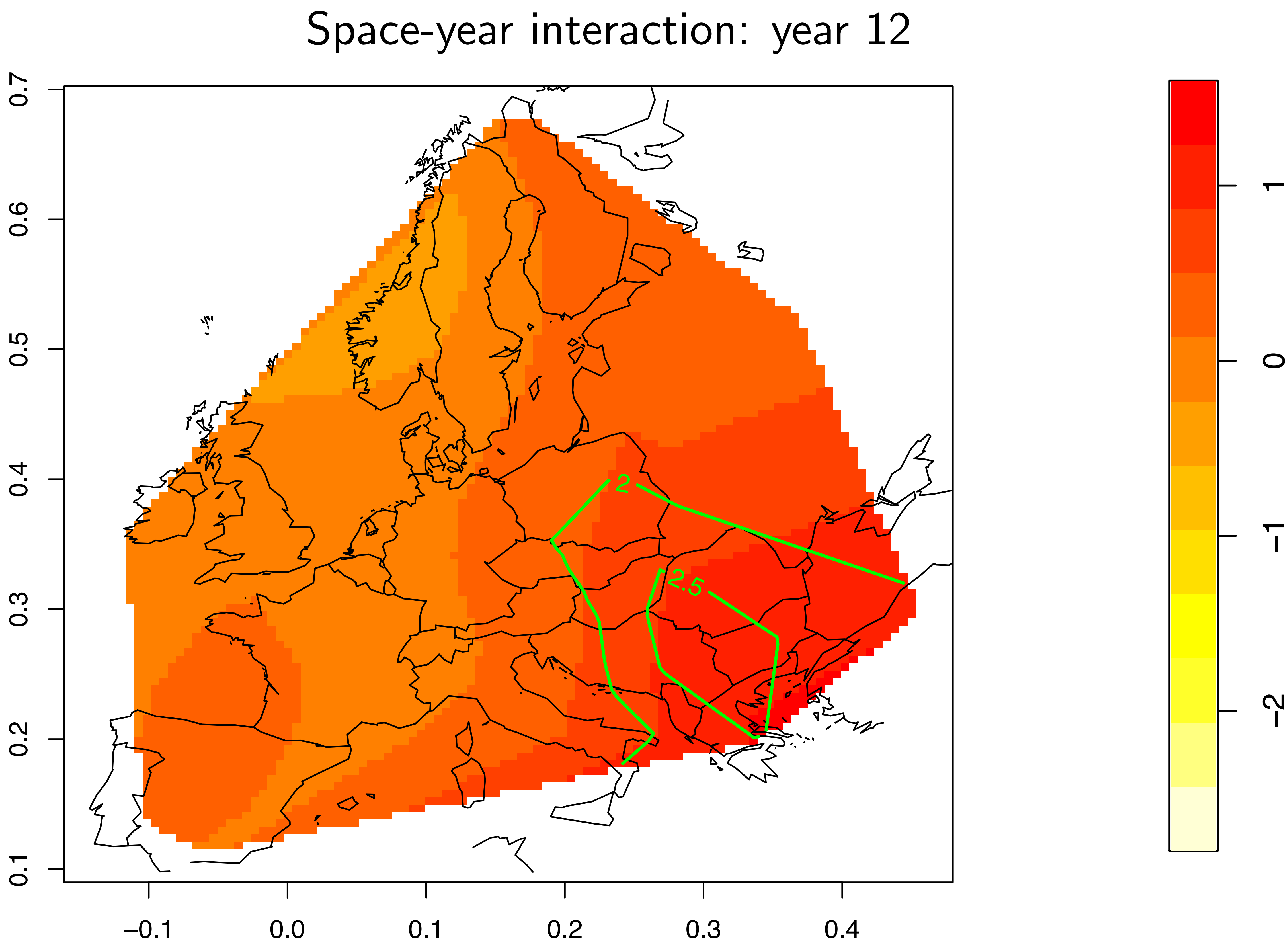
SO_2 pollution



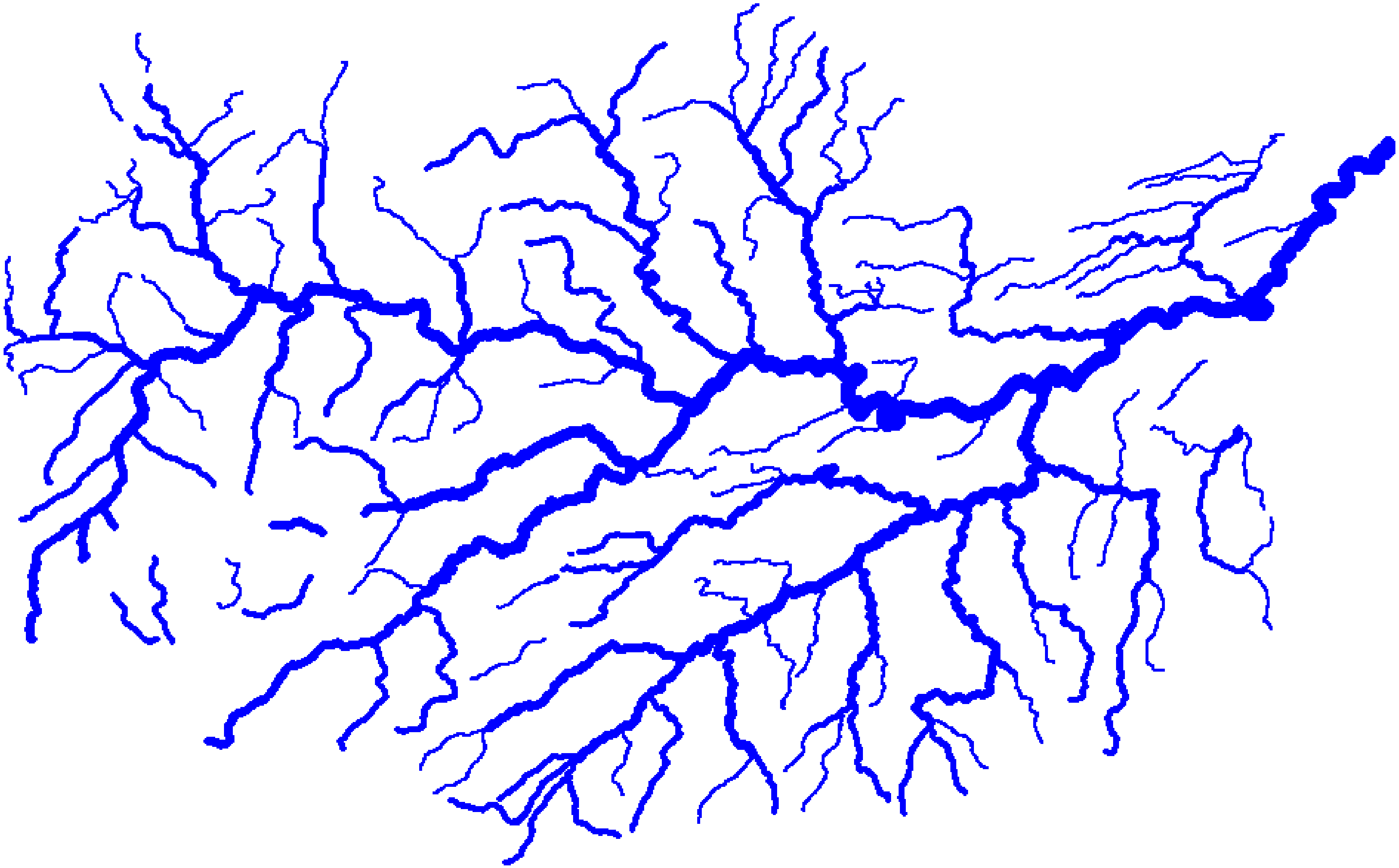
SO_2 pollution



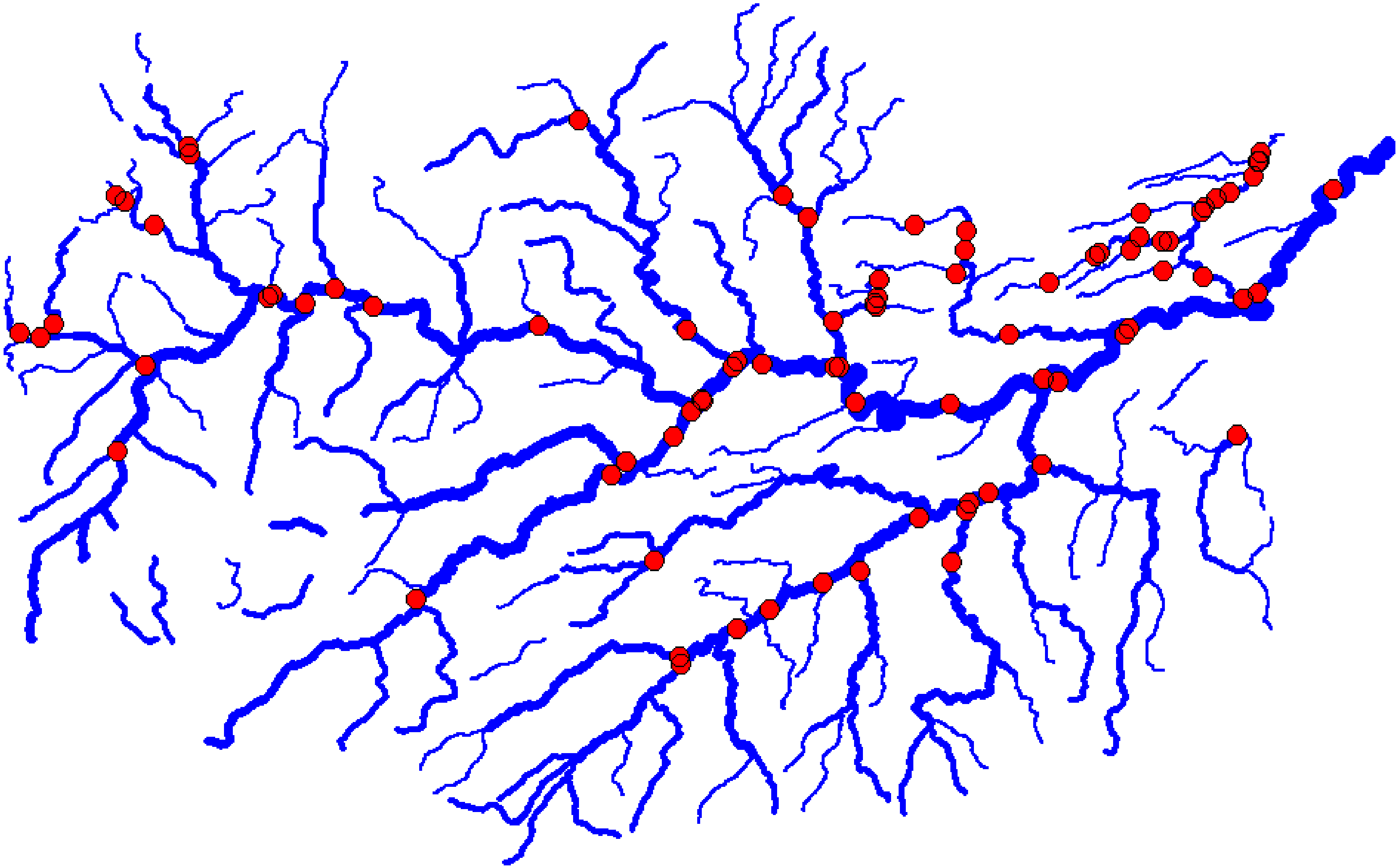
SO_2 pollution



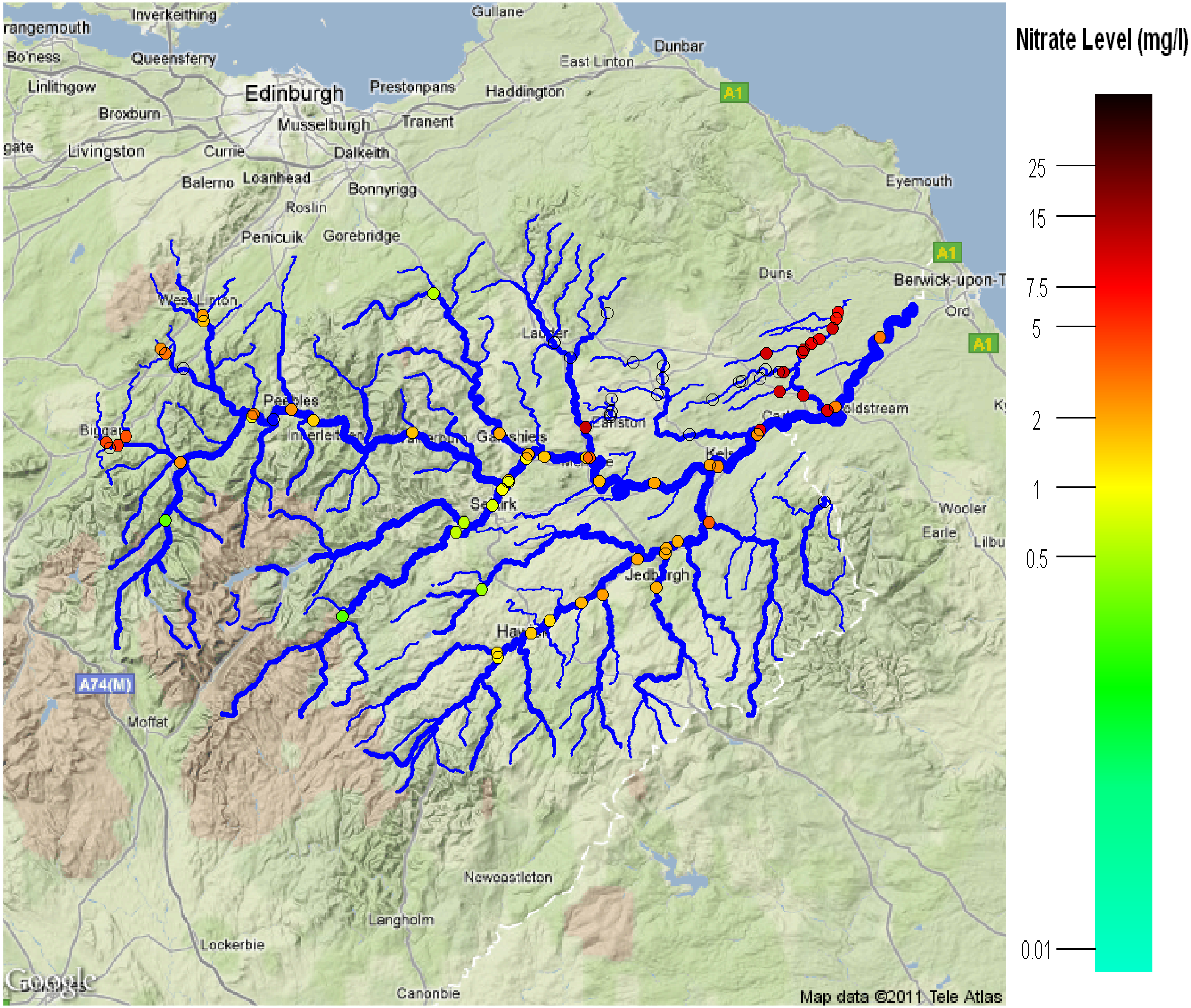
The River Tweed



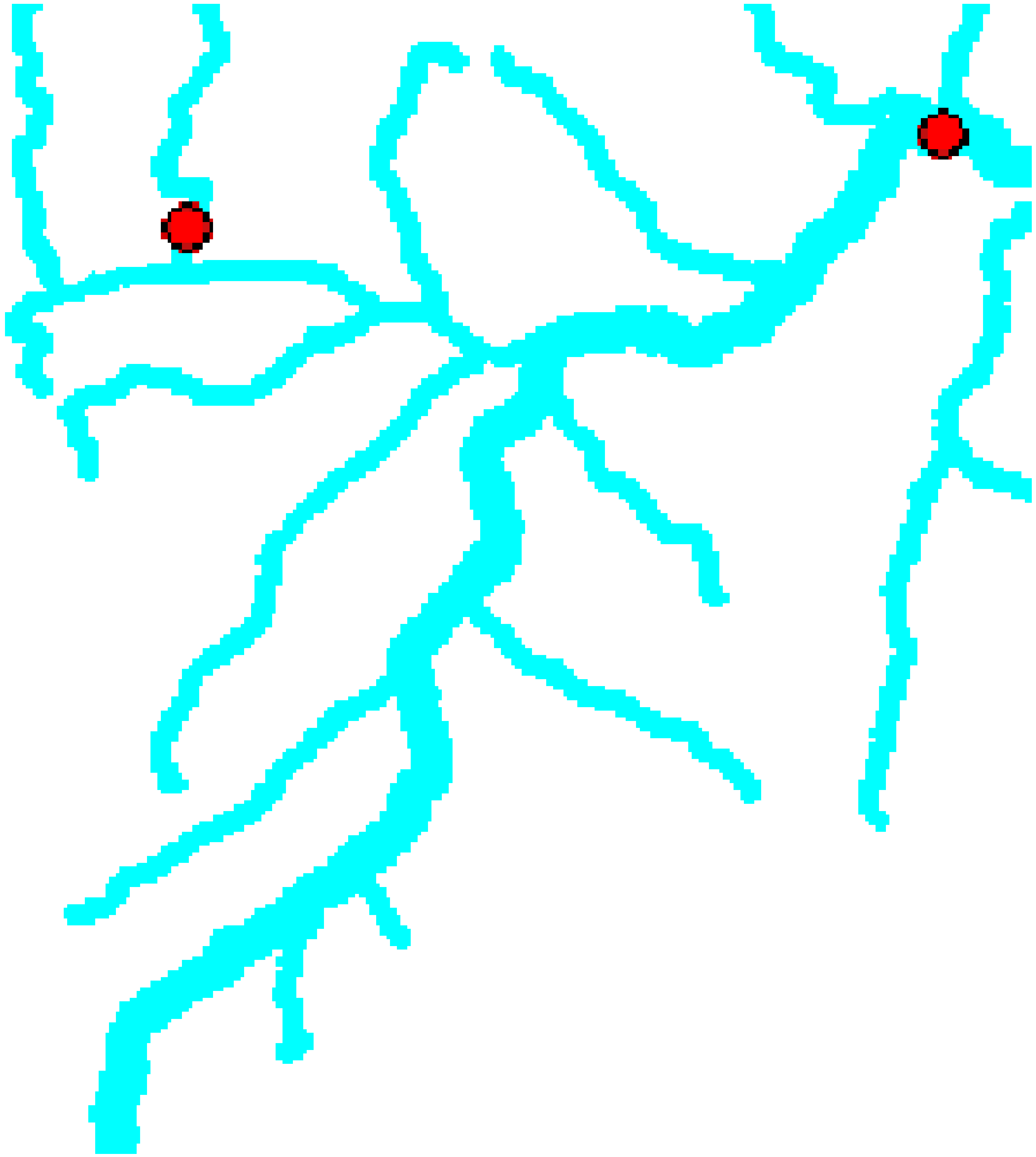
The River Tweed



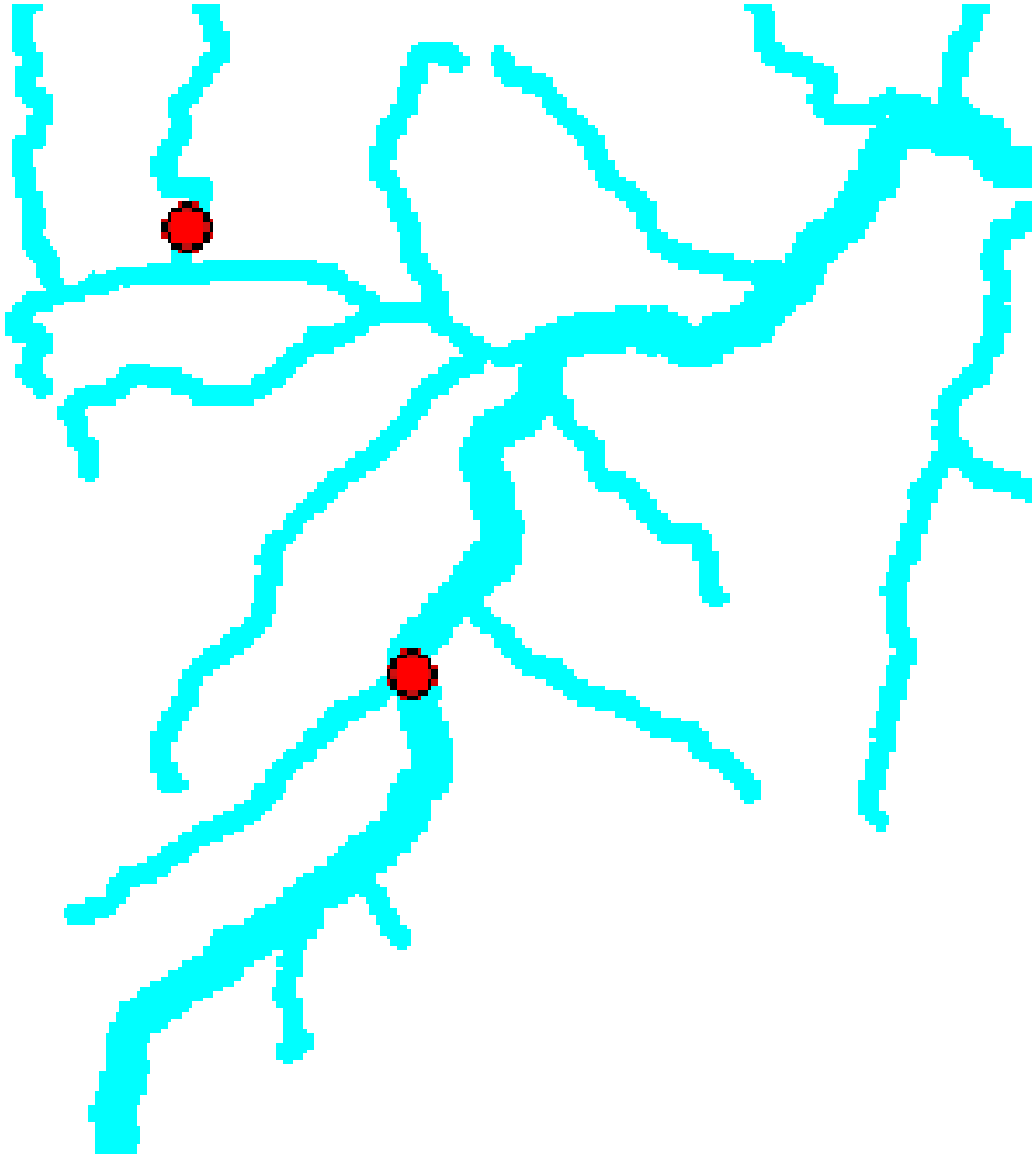
The River Tweed



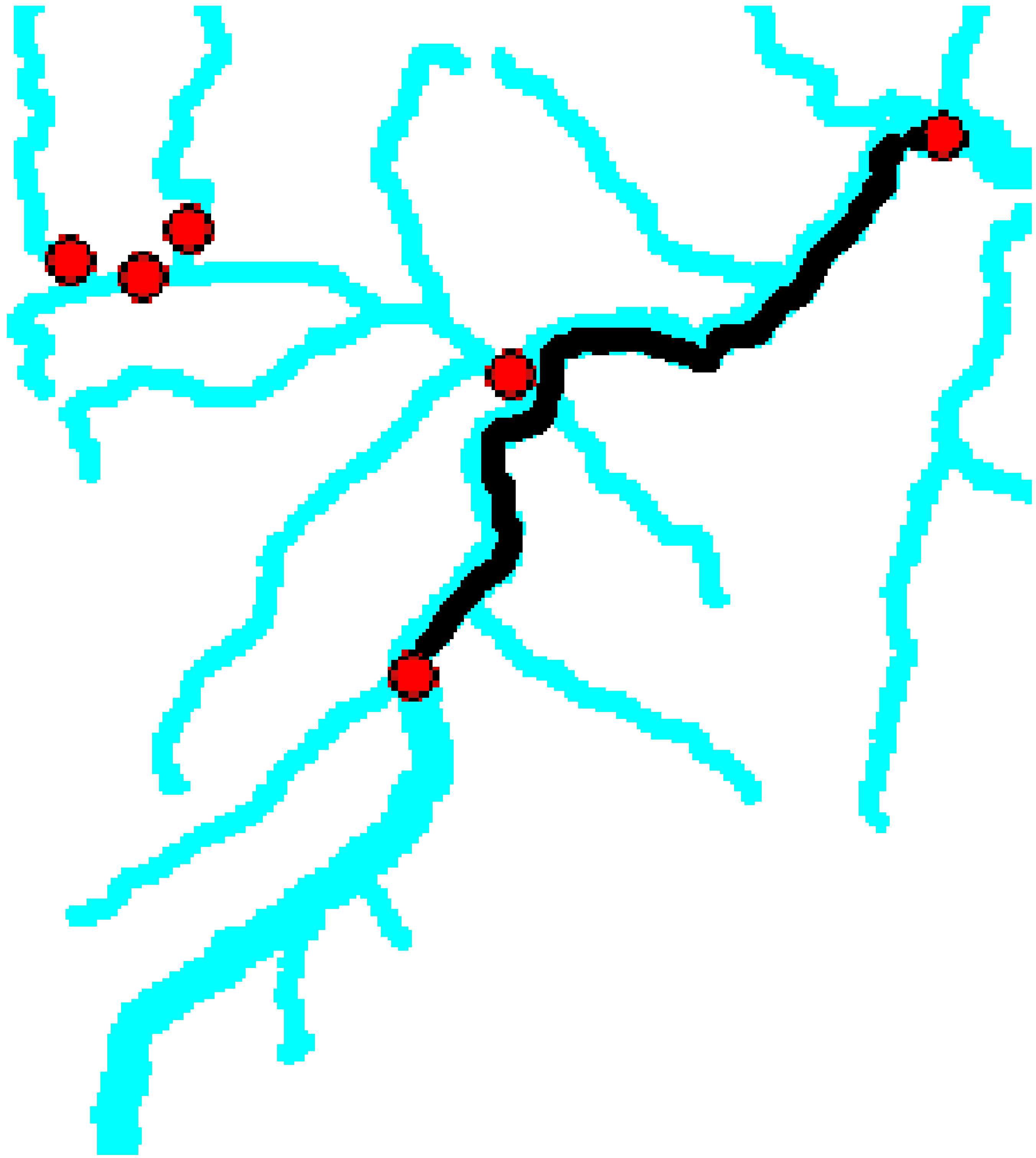
Flow connected



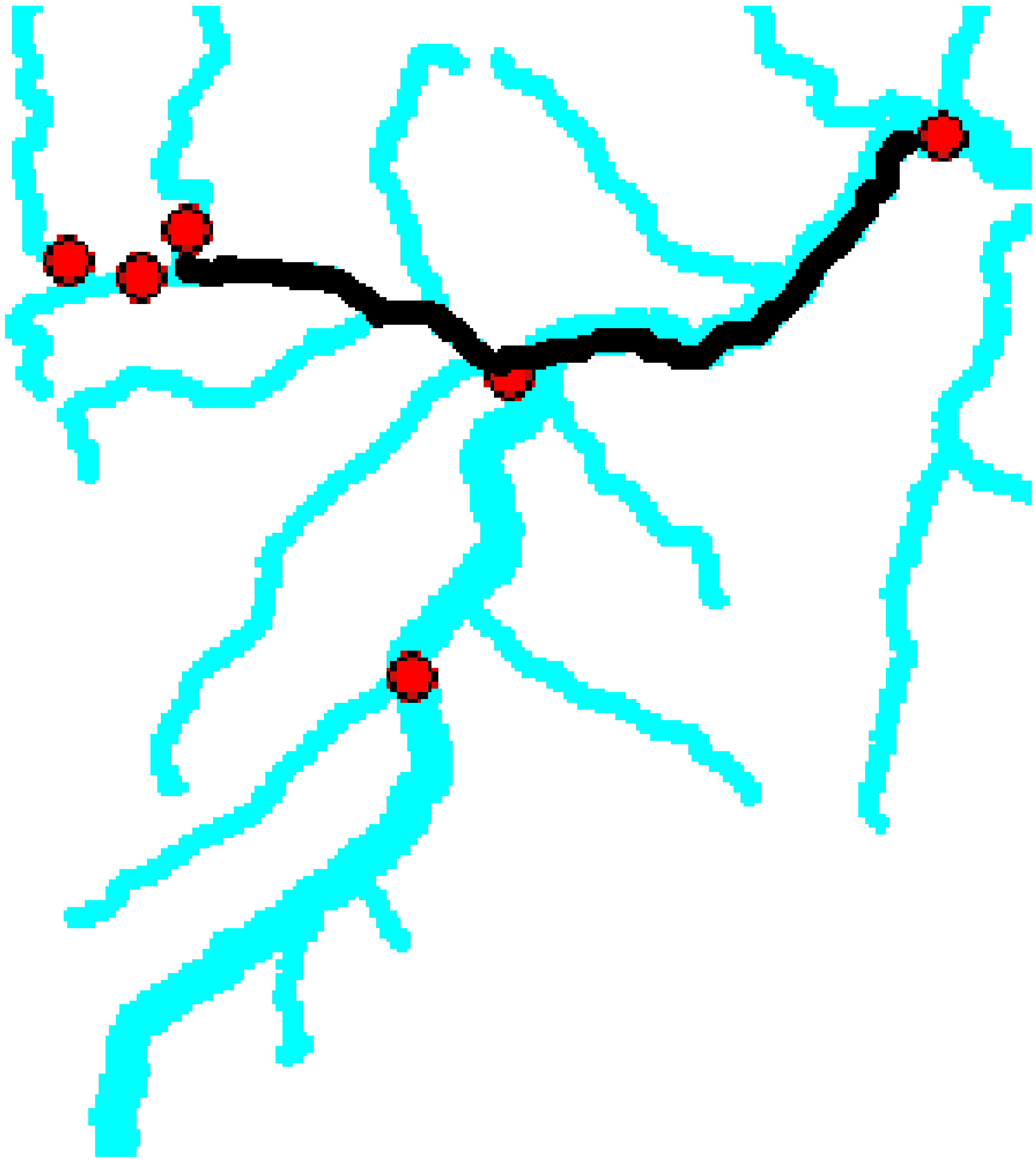
Not flow connected



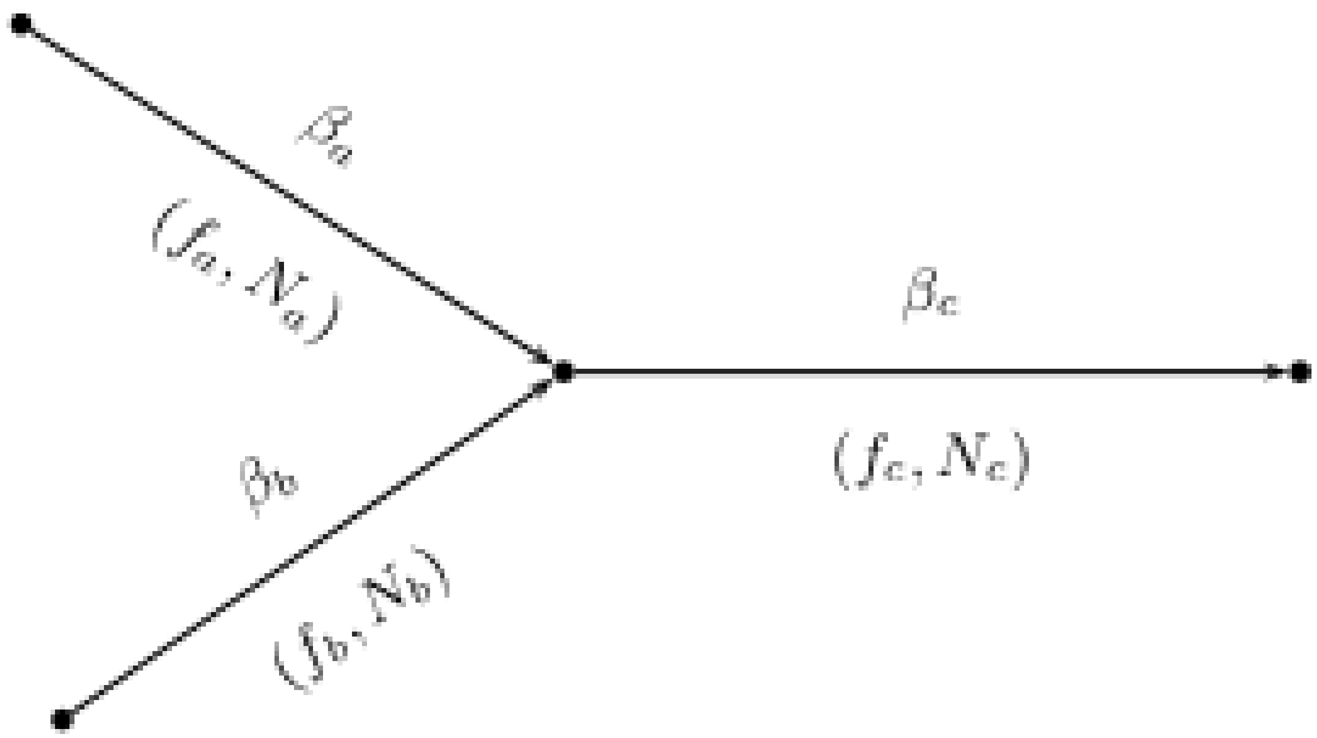
River distance



River distance



Confluences



The ver Hoef model

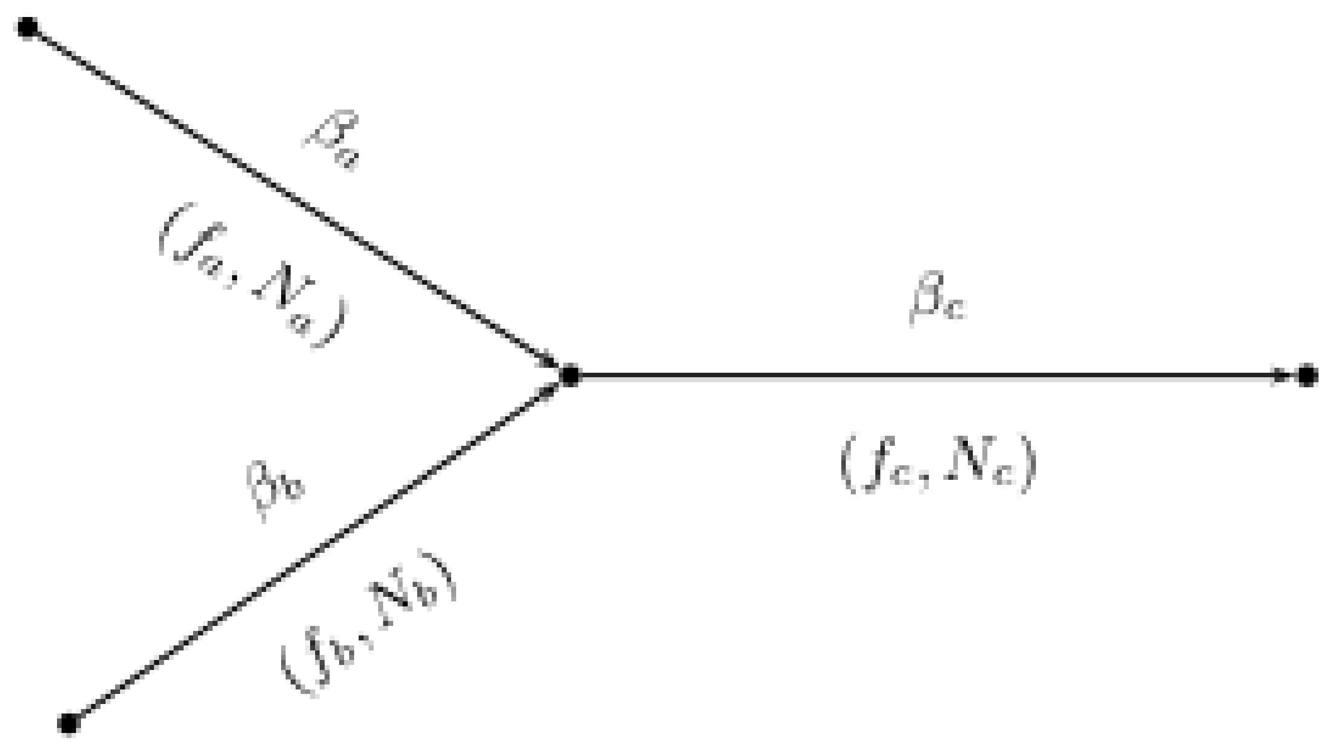
Covariance function

$$C(s_i, t_j) = \begin{cases} 0 & \text{if } s \text{ and } t \text{ are not flow connected;} \\ c_0 + c_1 & \text{if } s = t; \\ w \; c_1 \exp\left(-\frac{d_{s,t}}{c_2}\right) & \text{otherwise.} \end{cases}$$

where

- $d_{s,t}$ denotes the river distance between stations s and t;
- c_0, c_1, c_2 denote the nugget, partial sill and range parameters;
- $w = \prod_{k \in B_{s,t}} \sqrt{\omega_k}$
- $B_{s,t}$ denotes the set of all water stretches between s and t;
- ω_k denotes the proportion of flow contributed by water stretch k to its subsequent confluence.

Confluences



The $\sqrt{\omega}$ ensures that the variance in all water stretches is constant.

$$\text{var}\{\sqrt{\omega_a}N_a + \sqrt{\omega_b}N_b\} = (\omega_a + \omega_b)\sigma^2 = \sigma^2,$$

where σ^2 denotes the variance of each measurement.

Flexible regression - weight functions

Data $\{(y_i, x_i), i = 1, \ldots, n\}$ can be modelled by a flexible regression

$$y_i = m(x_i) + \varepsilon_i,$$

where m denotes a smooth function and the ε_i denote error terms.

Flexible regression - weight functions

Data $\{(y_i, x_i), i = 1, \ldots, n\}$ can be modelled by a flexible regression

$$y_i = m(x_i) + \varepsilon_i,$$

where m denotes a smooth function and the ε_i denote error terms.

A very simple form of estimator is

$$\hat{m}(x) = \sum_i v_i y_i,$$

where the weights v_i decrease with distance from x.

Flexible regression - weight functions

Data $\{(y_i, x_i), i = 1, \ldots, n\}$ can be modelled by a flexible regression

$$y_i = m(x_i) + \varepsilon_i,$$

where m denotes a smooth function and the ε_i denote error terms.

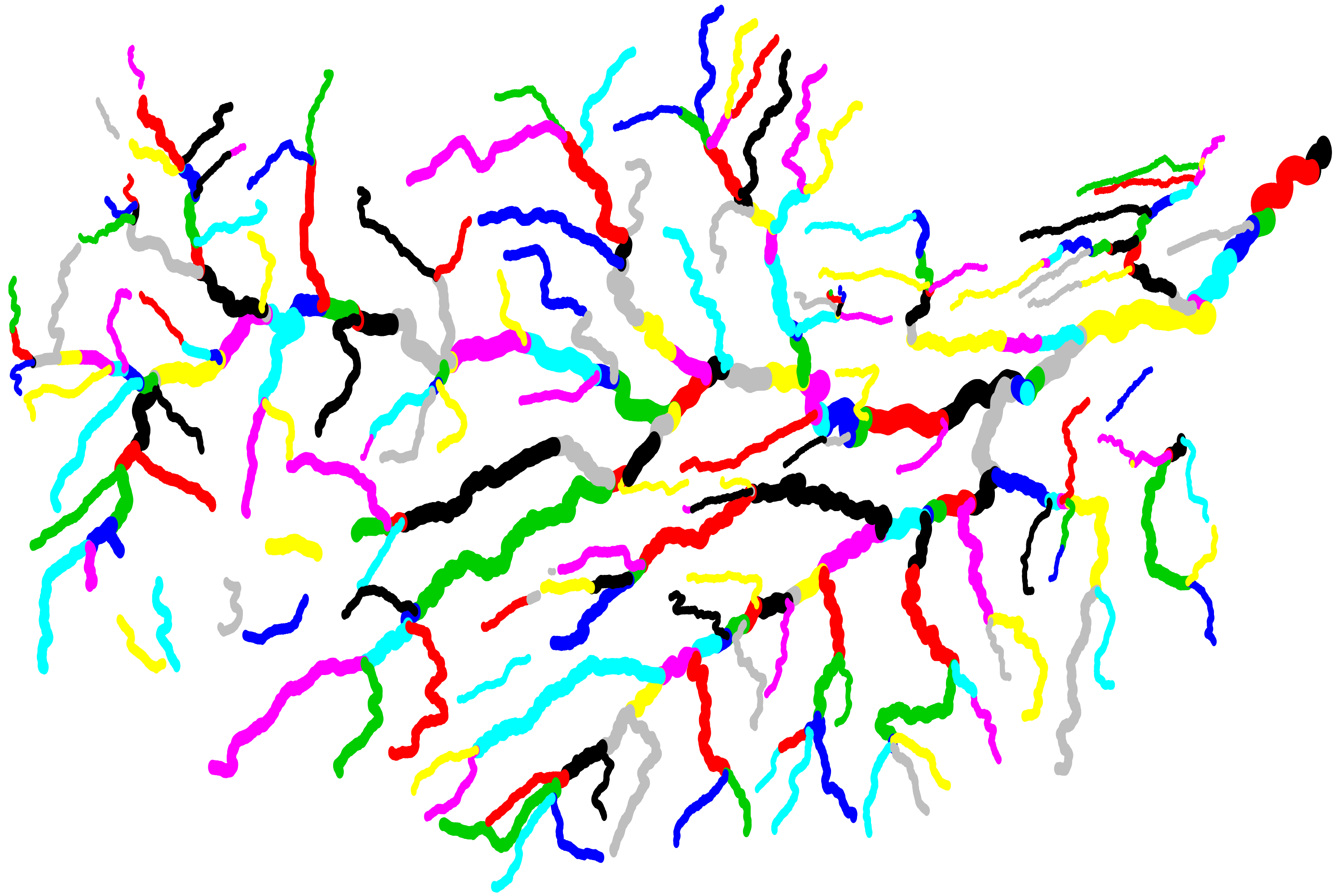
A very simple form of estimator is

$$\hat{m}(x) = \sum_i v_i y_i,$$

where the weights v_i decrease with distance from x.

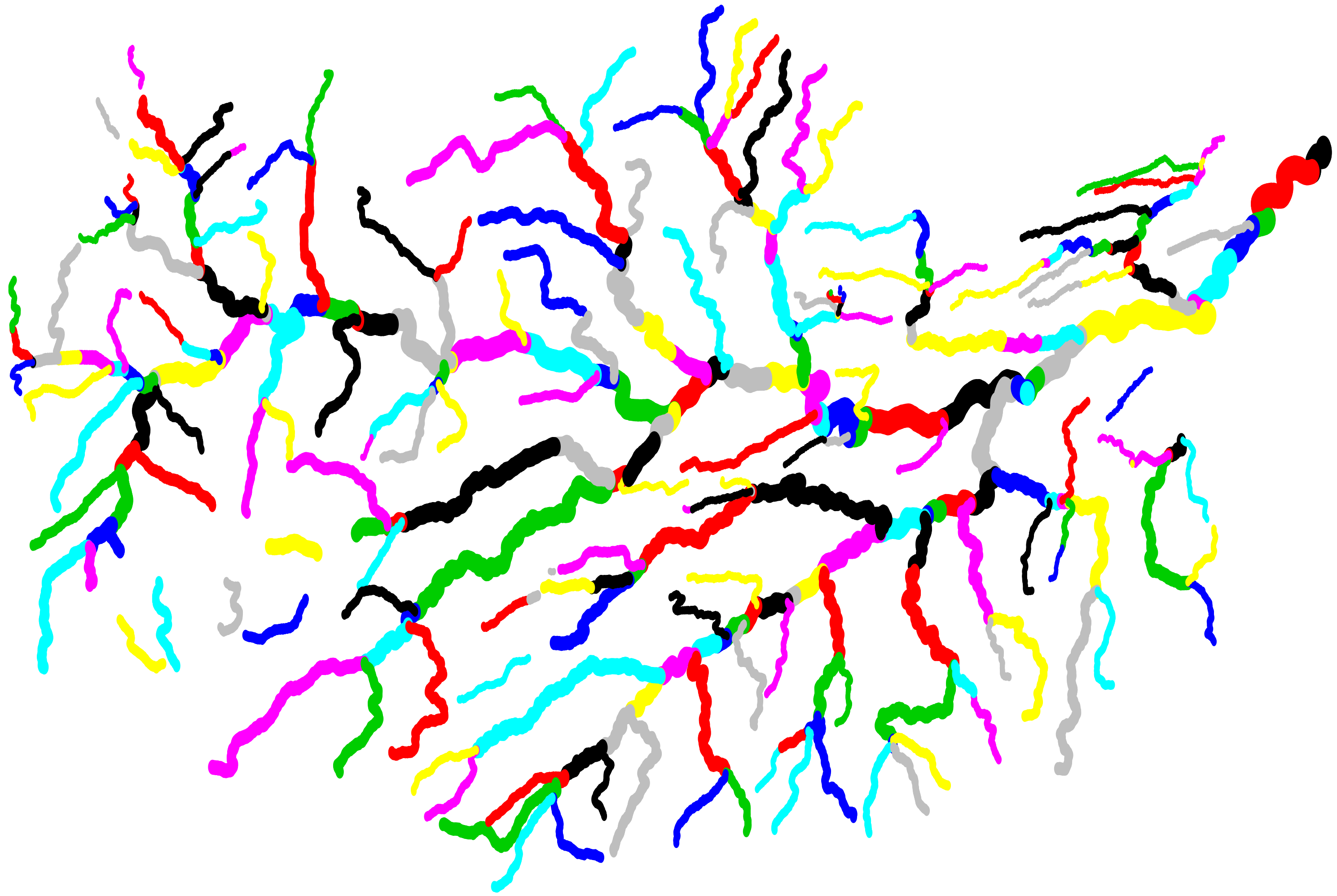
To adapt to a river network, we can use river distance and also incorporate the flow weights $w = \prod_{k \in B_{s,t}} \sqrt{\omega_k}$ into the weighting scheme.

Stream segments



An alternative approach to the estimation of smooth functions is to consider the network as a collection of small stream segments.

Stream segments

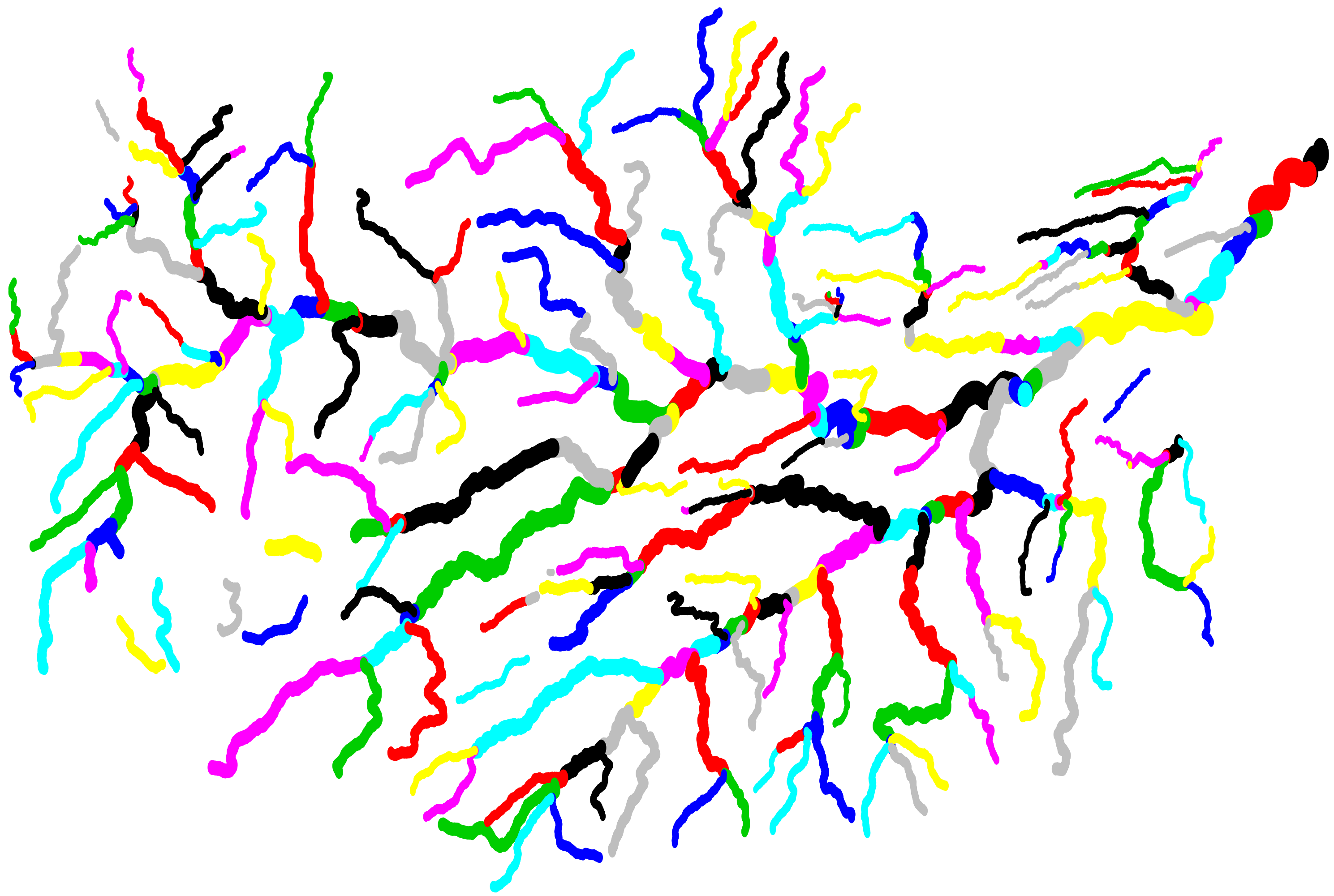


An alternative approach to the estimation of smooth functions is to consider the network as a collection of small stream segments.

An estimator is then available as

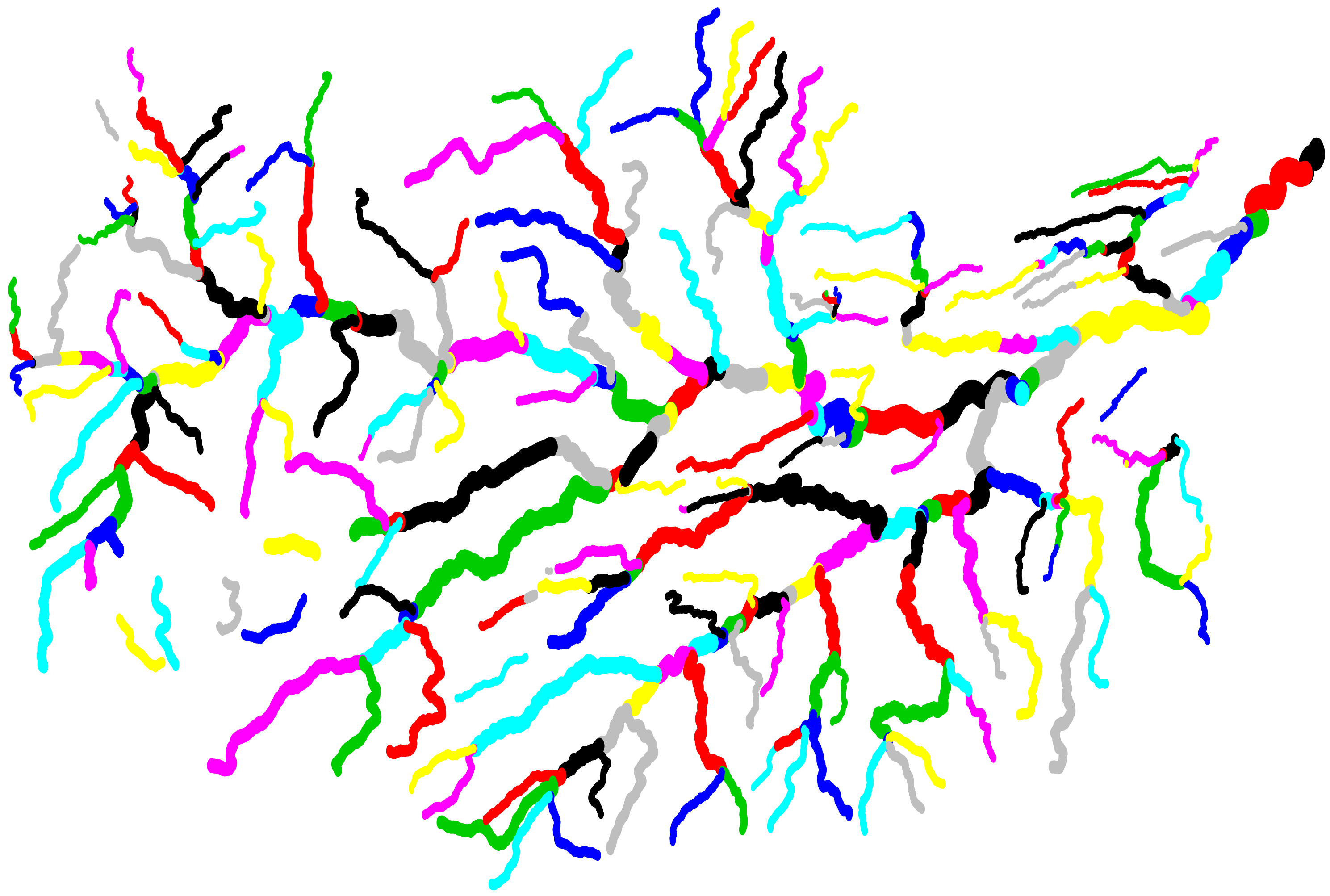
$$\hat{m}(x) = \beta_j, \quad \text{where } x \text{ lies in stream segment } j$$

Flexible regression - penalised splines



Formally, this is a b-spline approach of order 0.

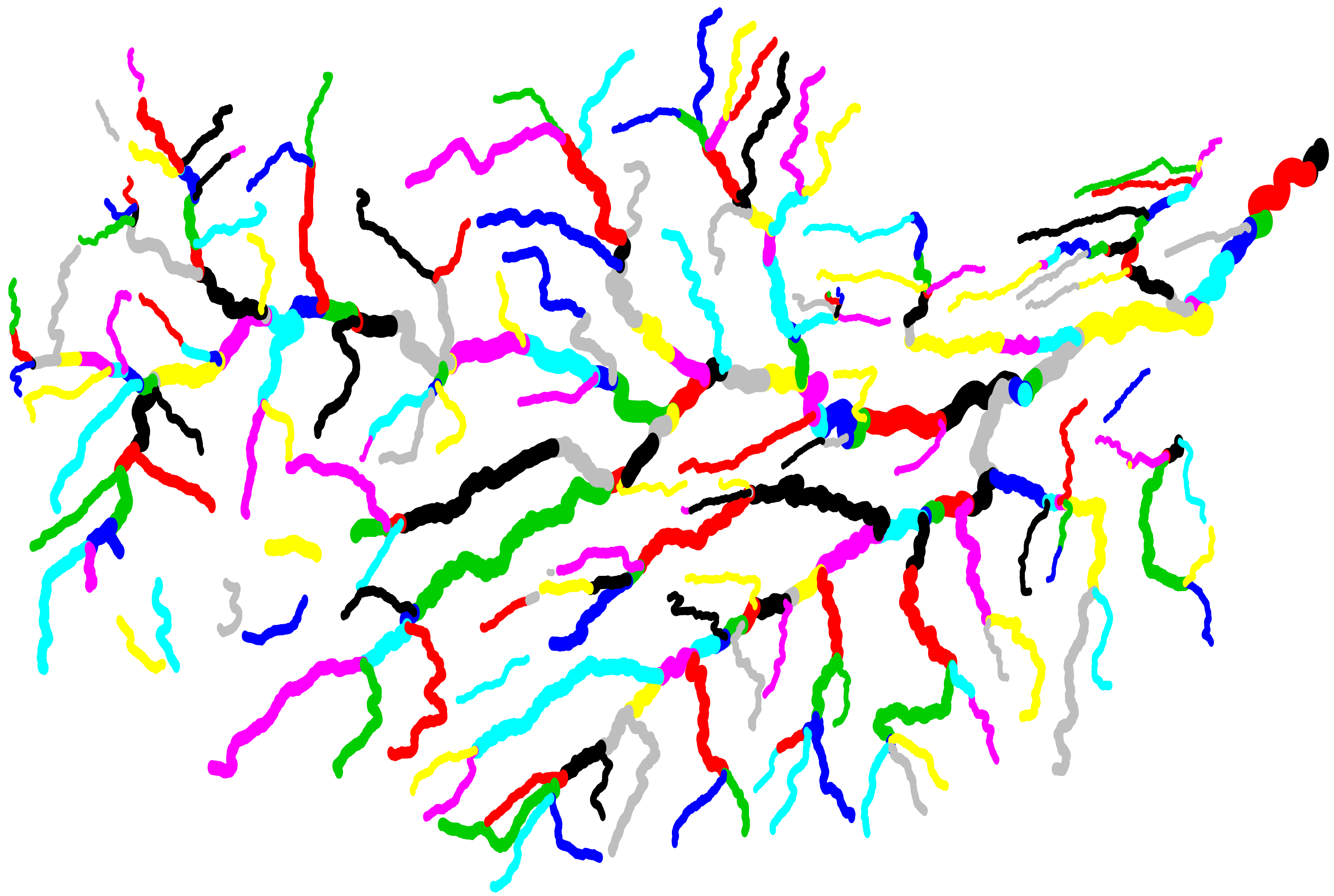
Flexible regression - penalised splines



Formally, this is a b-spline approach of order 0.

Smoothness is induced by use of a penalty, making this a *p-spline*.

Flexible regression - penalised splines

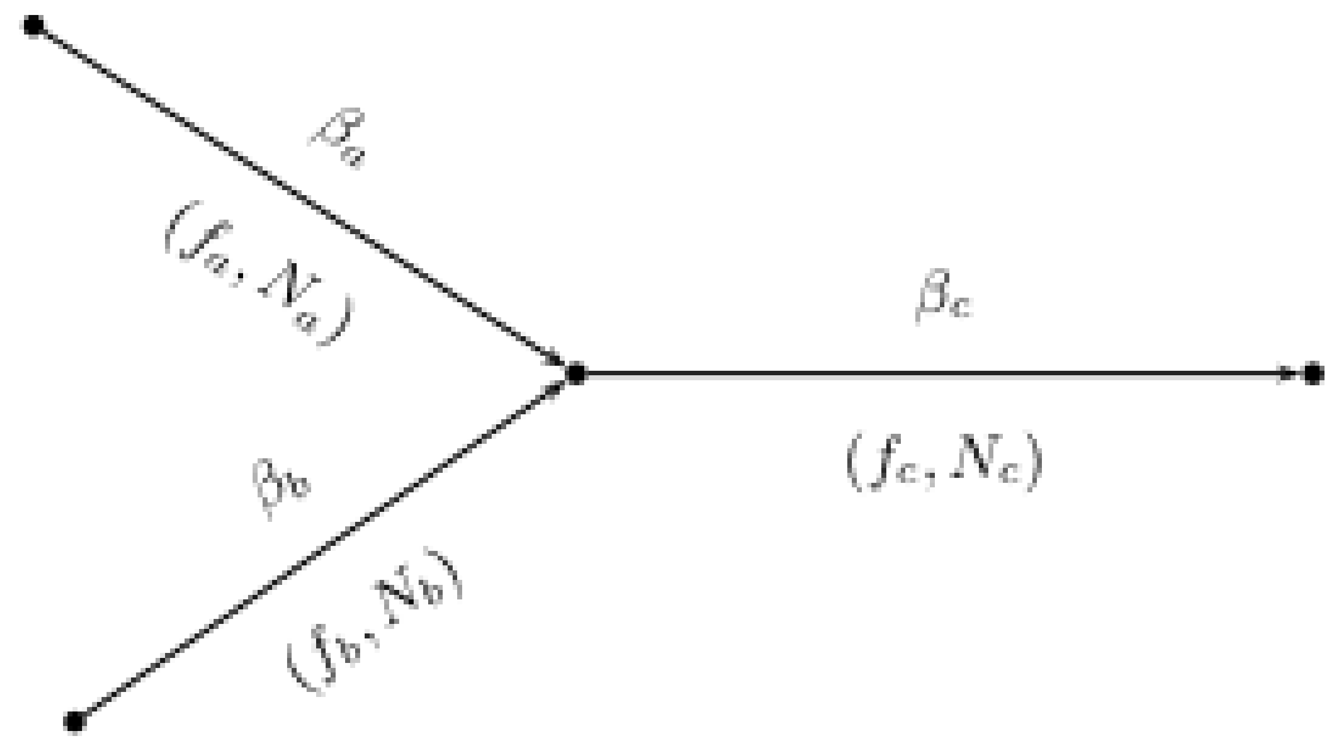


Formally, this is a b-spline approach of order 0.

Smoothness is induced by use of a penalty, making this a *p-spline*.

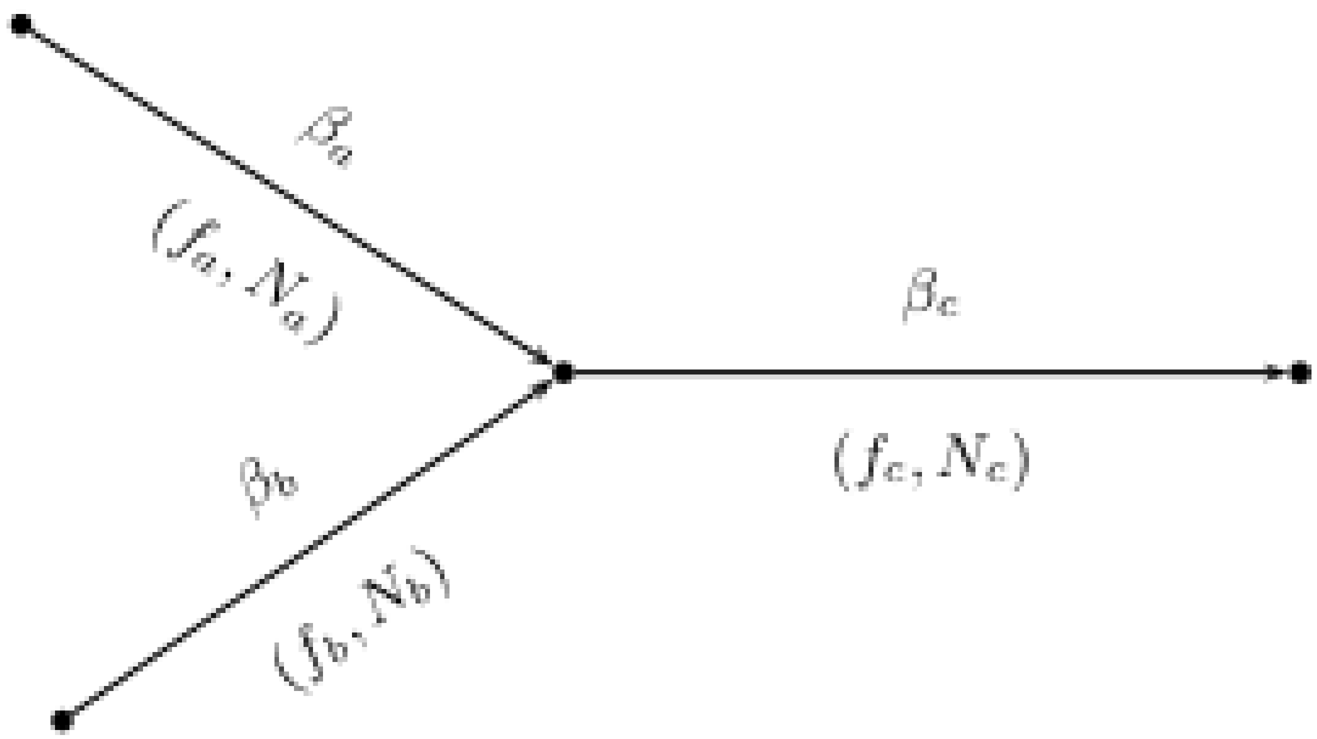
The 'smoothness' of β-values corresponding to adjacent stream units j and k, with no intervening confluence, can be measured by $(\beta_j - \beta_k)^2$.

Flexible regression - penalised splines



Where a confluence point is involved, the measure of smoothness needs to reflect the relative levels of flow in the contributing streams *a* and *b*.

Flexible regression - penalised splines

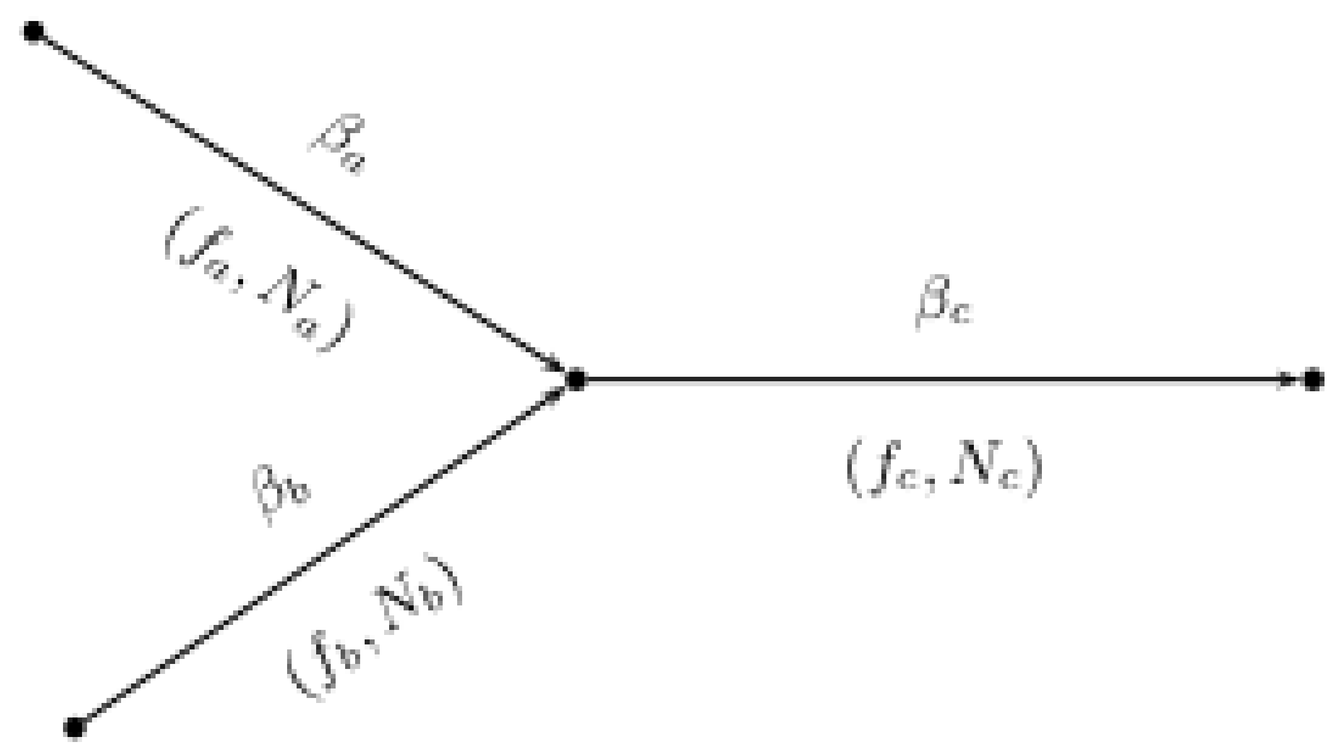


The relative flows of the inputs are $\omega_a = f_a/f_c$ and $\omega_b = f_b/f_c$.

The combined pollution input $\omega_a\beta_a + \omega_b\beta_b$ and output β_c are identical, following the principle of mass balance, if

$$\omega_a(\beta_a - \beta_c) + \omega_b(\beta_b - \beta_c) = 0.$$

Flexible regression - penalised splines



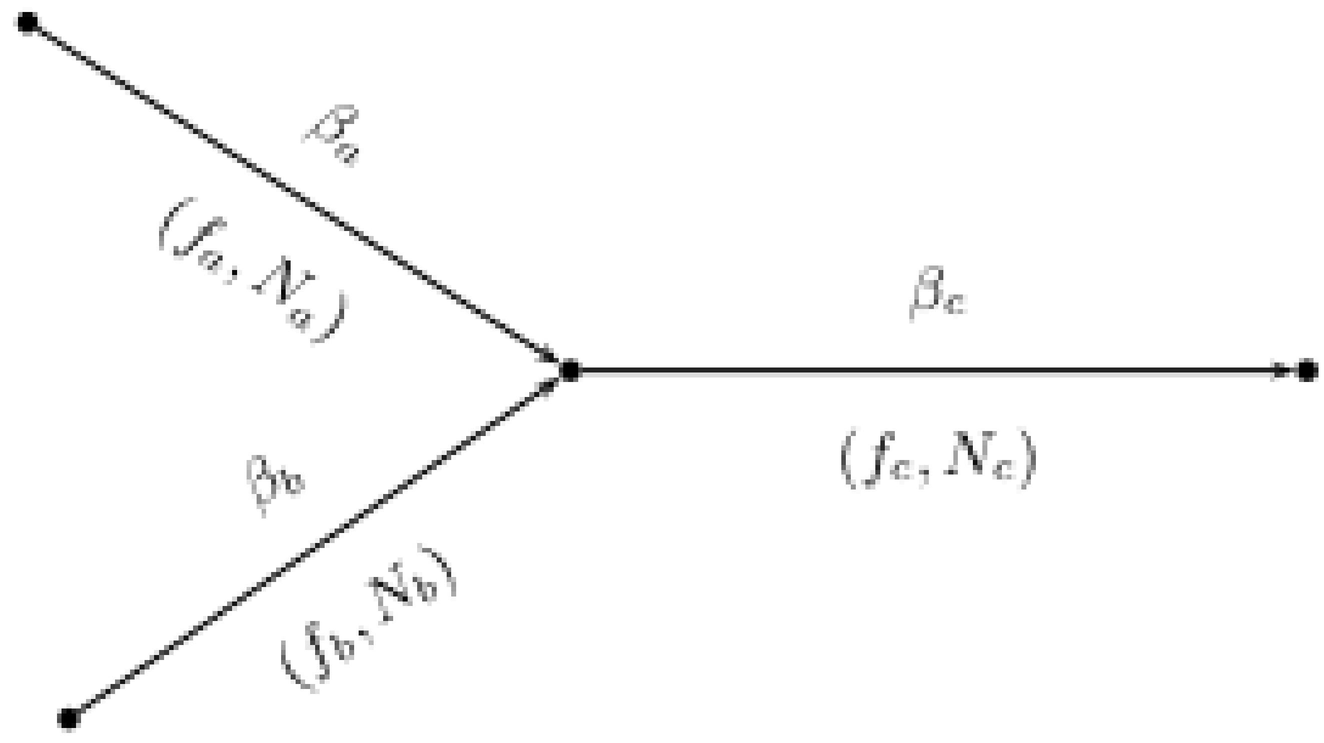
The combined pollution input $\omega_a\beta_a + \omega_b\beta_b$ and output β_c are identical, following the principle of mass balance, if

$$\omega_a(\beta_a - \beta_c) + \omega_b(\beta_b - \beta_c) = 0.$$

Smoothness across the confluence can therefore be achieved through the penalty

$$\omega_a^2(\beta_a - \beta_c)^2 + \omega_b^2(\beta_b - \beta_c)^2.$$

Flexible regression - penalised splines



Smoothness across the confluence can therefore be achieved through the penalty

$$\omega_a^2(\beta_a - \beta_c)^2 + \omega_b^2(\beta_b - \beta_c)^2.$$

This has the attractive form of combining penalties for smoothness across each flow path of the confluence, with weights determined by the relative volumes.

Flexible regression - penalised splines

- A p-spline model can be formulated as a regression model

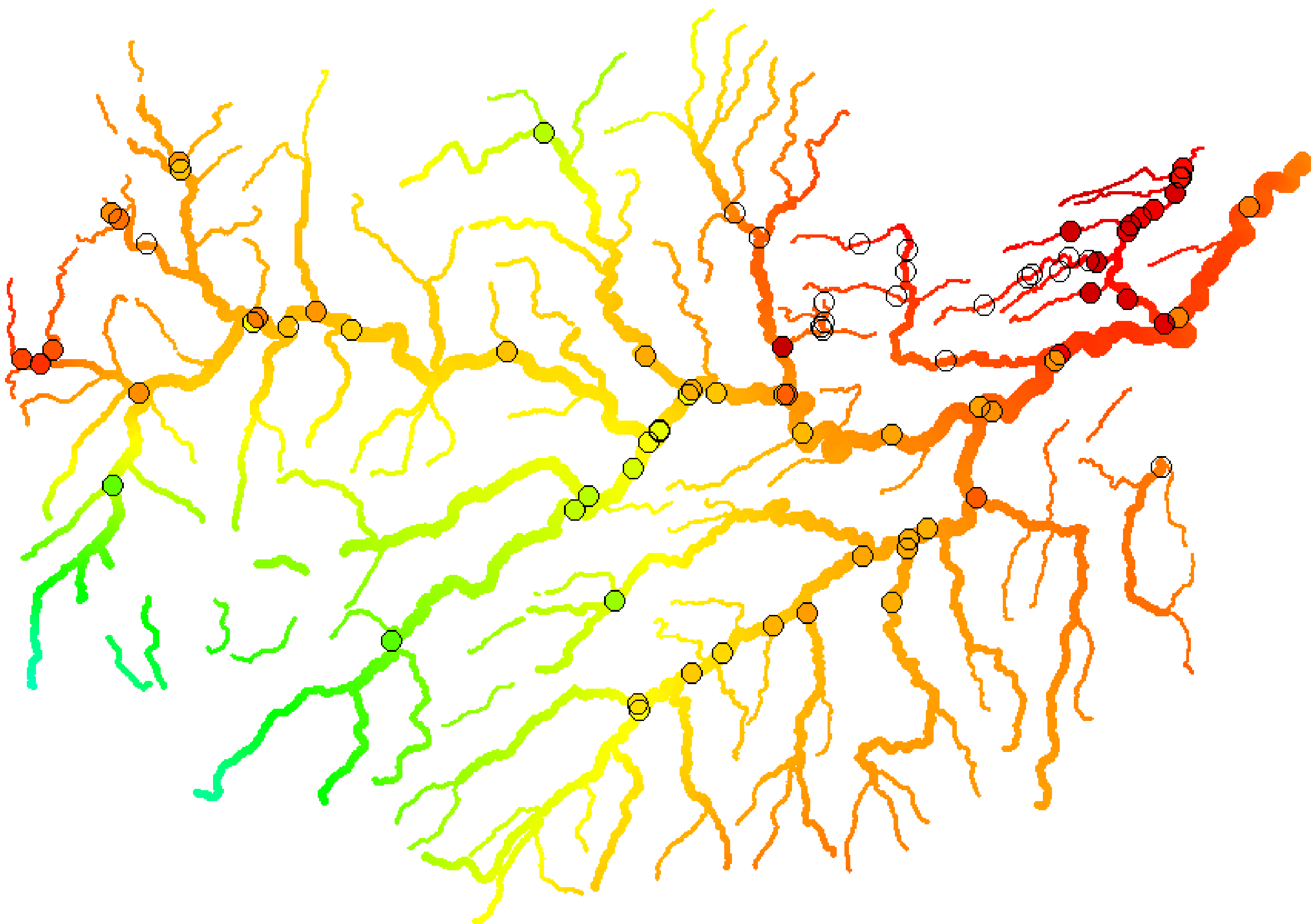
$$y = B\beta + \varepsilon,$$

  where the matrix B is simply an indicator matrix whose rows identify the stream unit containing y_i.
- The model is fitted by minimising the penalised sum-of-squares

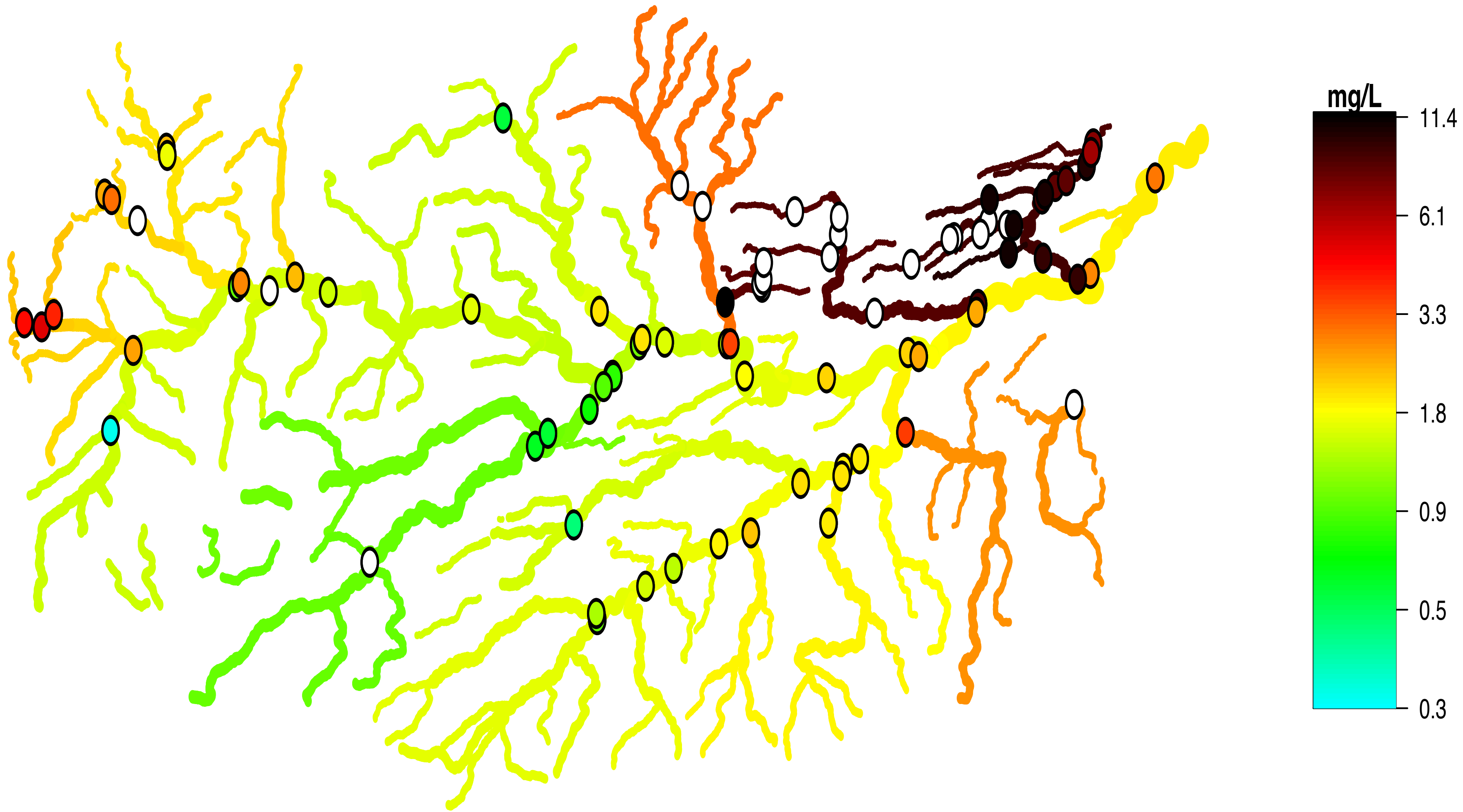
$$(y - B\beta)^T(y - B\beta) + \lambda\beta^T D^T D\beta$$

  with respect to β. The matrix D generates the penalty.
- The penalty parameter λ controls the degree of smoothing.
- The solution to this least squares problem is easily shown to be $\hat{\beta} = (B^T B + \lambda D^T D)^{-1} B^T y$.
- The linear form of this expression allows an approximate degrees of freedom to be computed as the trace of the 'hat' matrix.

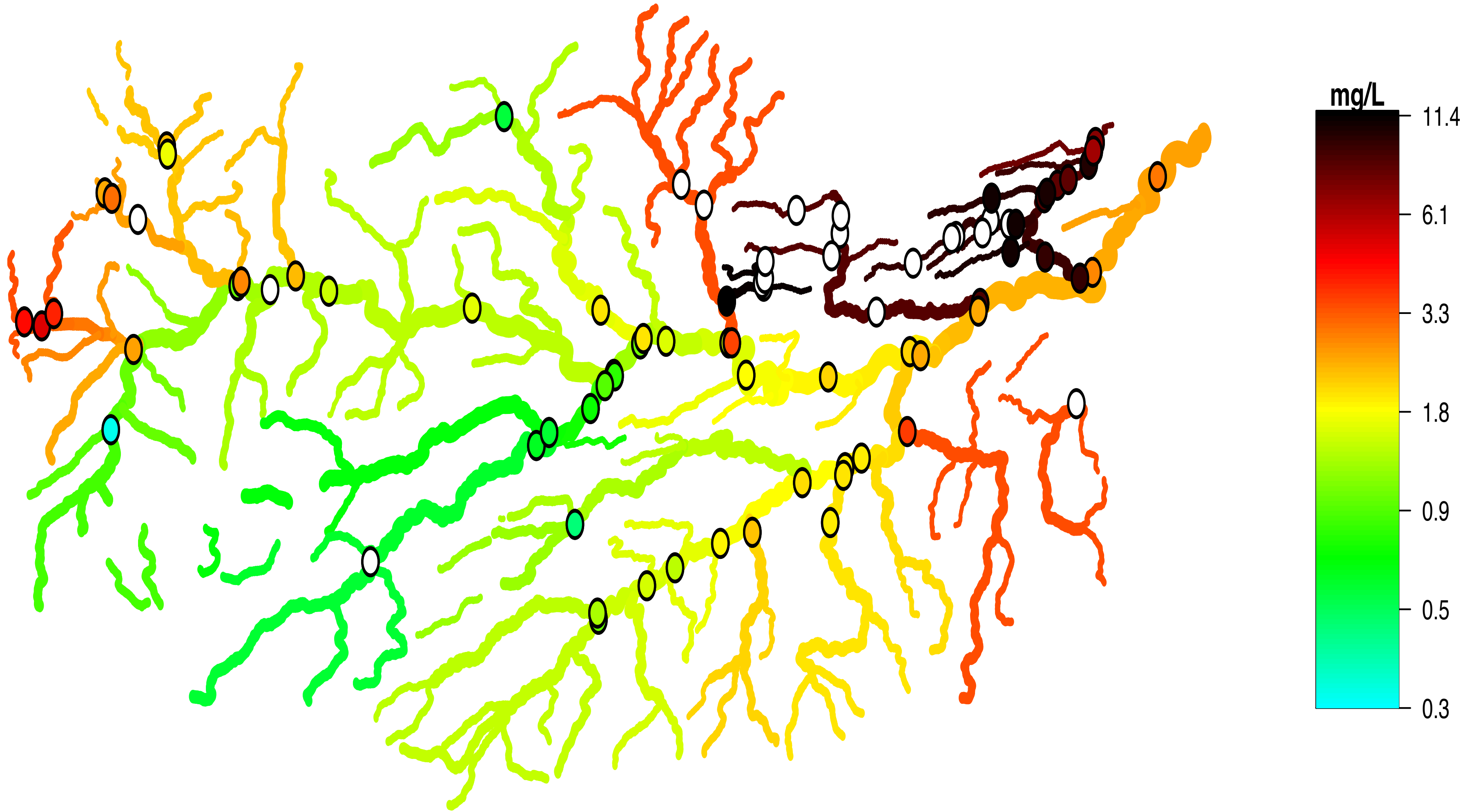
The River Tweed



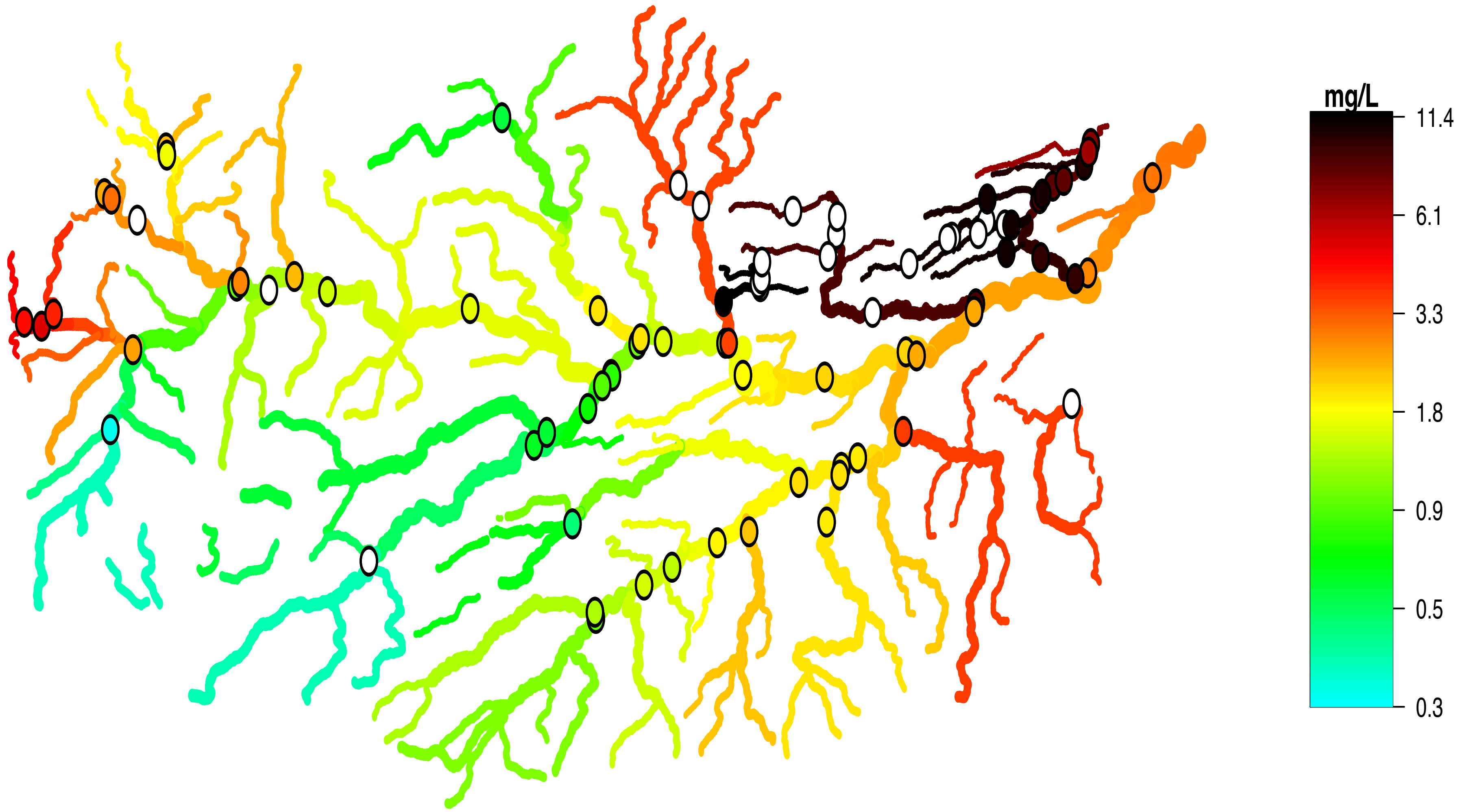
The River Tweed



The River Tweed



The River Tweed



Additive models

In order to incorporate time t_i and day of the year z_i,

$$y_i = \mu + m_s(s_i) + m_t(t_i) + m_z(z_i) + \varepsilon_i \quad (1)$$

where the three functions m_s, m_t, m_z describe spatial, temporal and seasonal trends..

Additive models

In order to incorporate time t_i and day of the year z_i,

$$y_i = \mu + m_s(s_i) + m_t(t_i) + m_z(z_i) + \varepsilon_i \tag{1}$$

where the three functions m_s, m_t, m_z describe spatial, temporal and seasonal trends..

- If each each of the trend functions is estimated by b-splines then they can be represented as $B_s\beta_s$, $B_t\beta_t$, $B_z\beta_z$ where the columns of the design matrices evaluate each basis function at the observed values of the relevant covariate.

Additive models

In order to incorporate time t_i and day of the year z_i,

$$y_i = \mu + m_s(s_i) + m_t(t_i) + m_z(z_i) + \varepsilon_i \quad (1)$$

where the three functions m_s, m_t, m_z describe spatial, temporal and seasonal trends..

- If each each of the trend functions is estimated by b-splines then they can be represented as $B_s\beta_s$, $B_t\beta_t$, $B_z\beta_z$ where the columns of the design matrices evaluate each basis function at the observed values of the relevant covariate.
- B-splines of order 0 can be used for the spatial network, while cubic b-splines would be a good choice for the temporal and seasonal effects, which operate in more standard sample spaces.

Additive models

In order to incorporate time t_i and day of the year z_i,

$$y_i = \mu + m_s(s_i) + m_t(t_i) + m_z(z_i) + \varepsilon_i \qquad (1)$$

where the three functions m_s, m_t, m_z describe spatial, temporal and seasonal trends..

- If each each of the trend functions is estimated by b-splines then they can be represented as $B_s\beta_s$, $B_t\beta_t$, $B_z\beta_z$ where the columns of the design matrices evaluate each basis function at the observed values of the relevant covariate.
- B-splines of order 0 can be used for the spatial network, while cubic b-splines would be a good choice for the temporal and seasonal effects, which operate in more standard sample spaces.
- In the presence of an overall mean parameter μ, identifiability can be achieved simply be requiring the parameter vector for each term to sum to 0.

Additive models

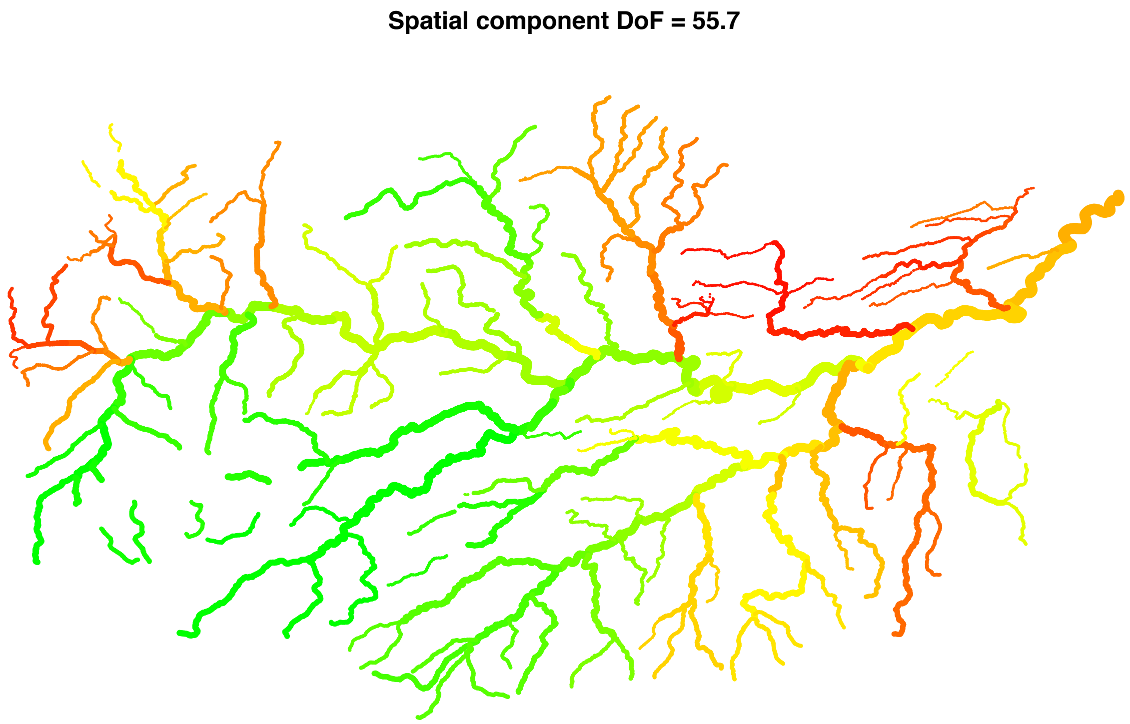
In order to incorporate time t_i and day of the year z_i,

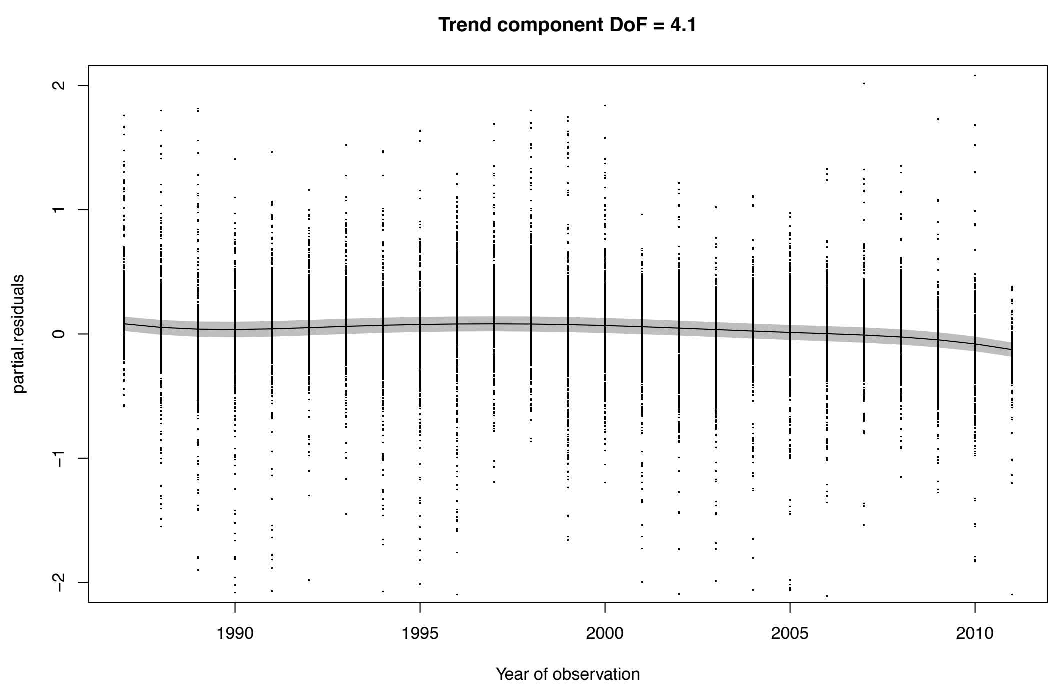
$$y_i = \mu + m_s(s_i) + m_t(t_i) + m_z(z_i) + \varepsilon_i \quad (1)$$

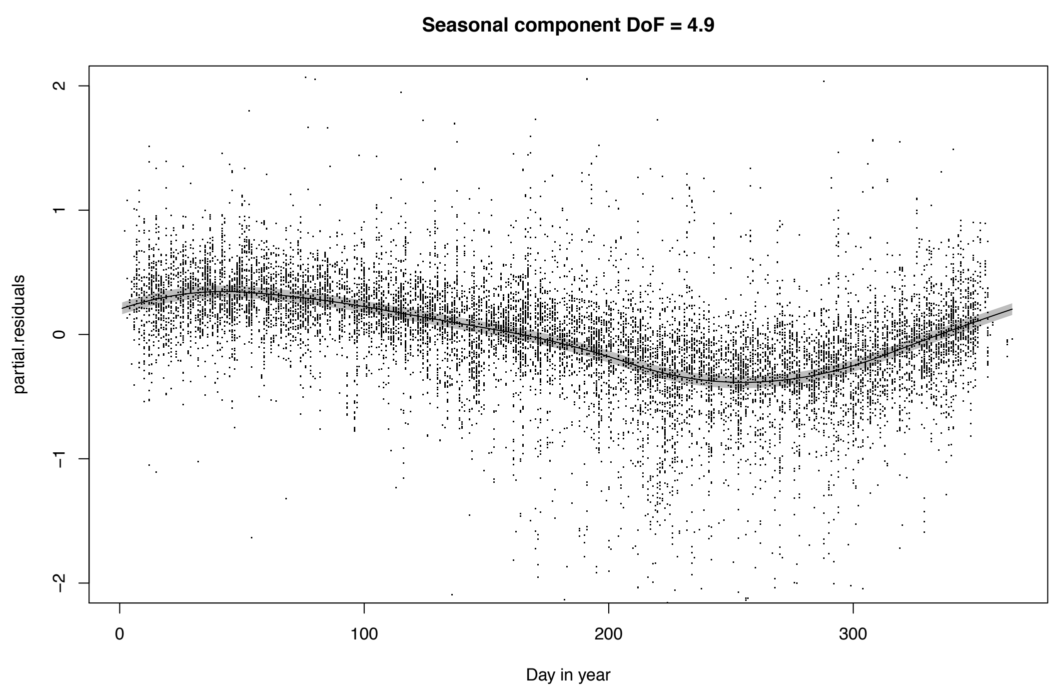
where the three functions m_s, m_t, m_z describe spatial, temporal and seasonal trends..

- If each each of the trend functions is estimated by b-splines then they can be represented as $B_s\beta_s$, $B_t\beta_t$, $B_z\beta_z$ where the columns of the design matrices evaluate each basis function at the observed values of the relevant covariate.
- B-splines of order 0 can be used for the spatial network, while cubic b-splines would be a good choice for the temporal and seasonal effects, which operate in more standard sample spaces.
- In the presence of an overall mean parameter μ, identifiability can be achieved simply be requiring the parameter vector for each term to sum to 0.
- The full model can be represented as $y = B\beta + \varepsilon$, where B combines the columns of the individual design matrices.

The River Tweed







Interaction terms

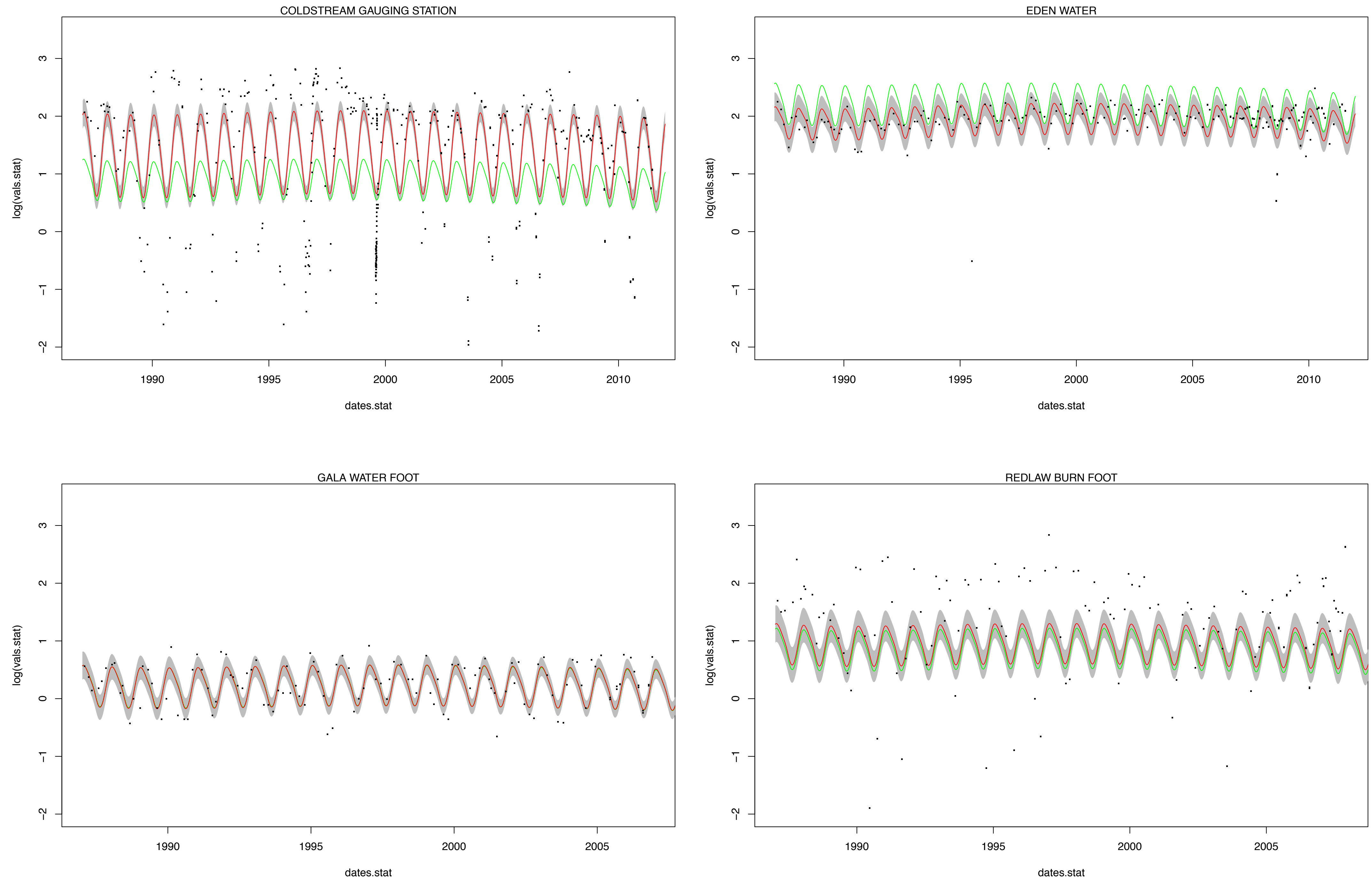
An interaction model has the form

$$y_i = \mu + m_s(s_i) + m_t(t_i) + m_z(z_i) + m_{s,t}(s_i, t_i) + m_{s,z}(s_i, z_i) + \varepsilon_i,$$

where the functions $m_{s,t}$ and $m_{s,z}$ encapsulate the adjustments required to capture the changes in time trend and seasonal effects over the river network.

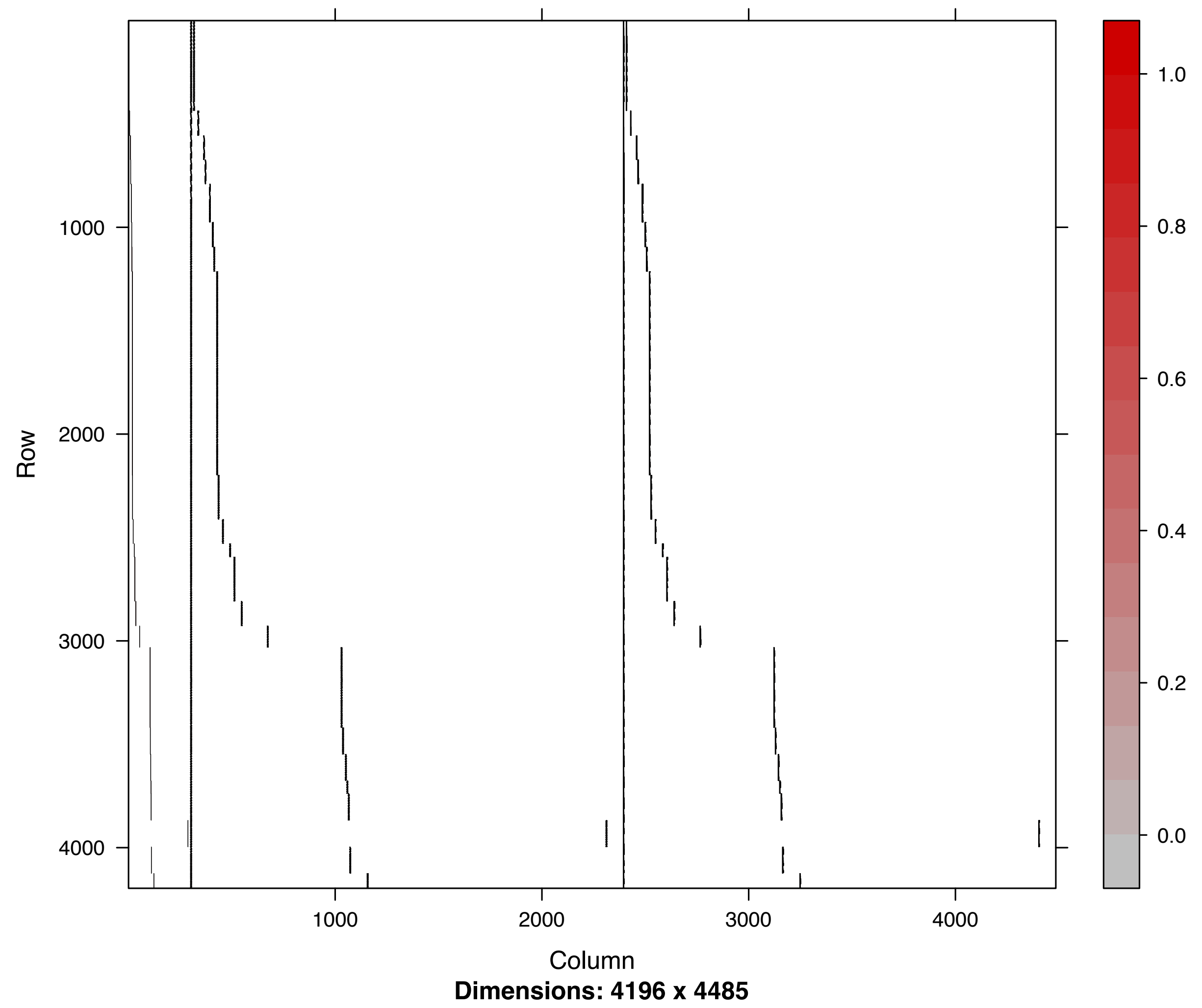
The model can be fitted by exactly the same mechanism described above.

The River Tweed



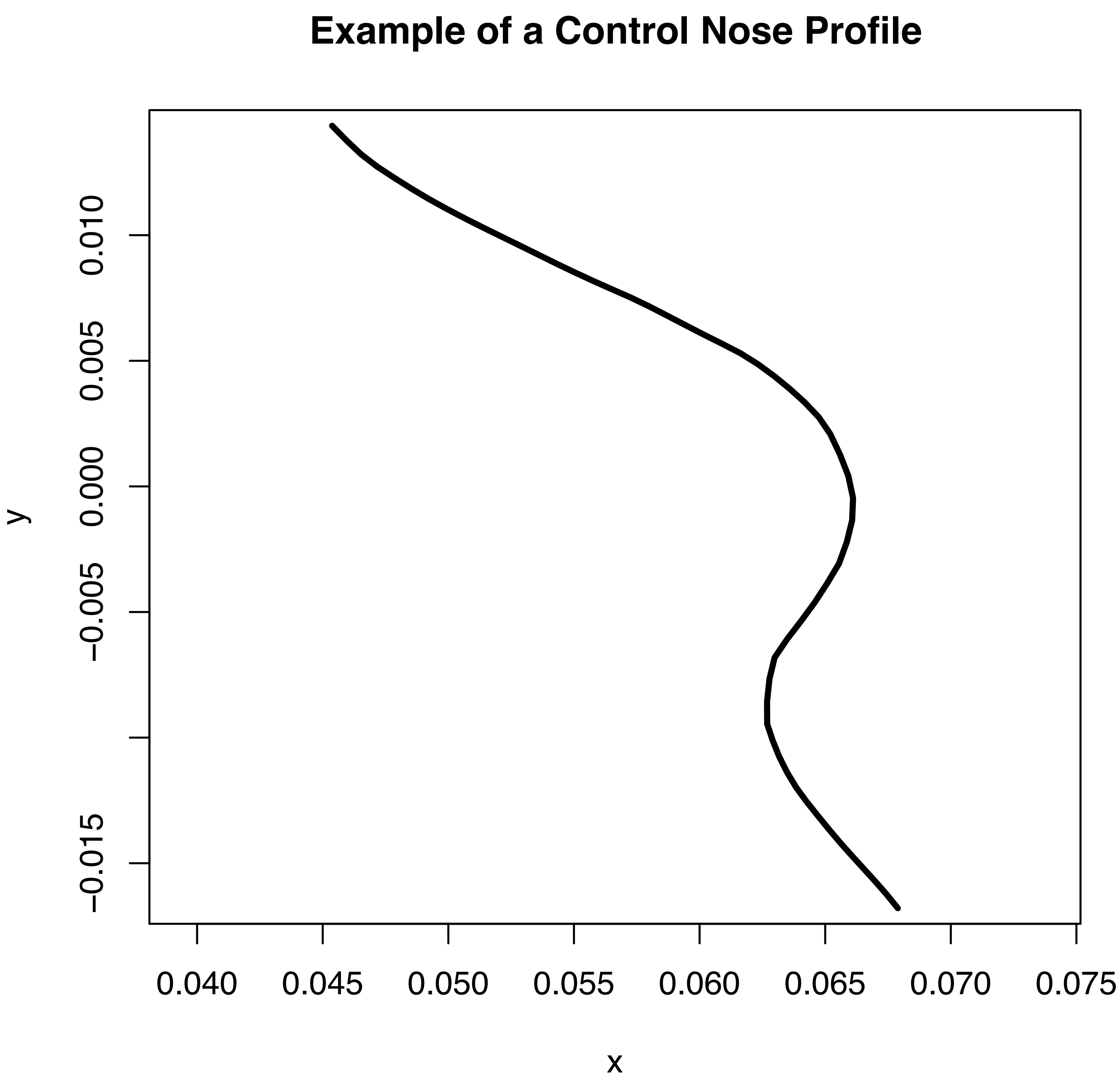
Computational issues

Calculations require the tensor product of an $n \times p$ matrix with n 1's, the rest 0's, which renders all model objects *very* sparse.



- The matrix $\mathbf{X}$ pictured above is 99.57% sparse.
- Using the R Matrix package of Bates and Mächler, $\mathbf{X}$ takes up only 1Mb of RAM and $\mathbf{X}^T\mathbf{X}$ takes 0.03s to calculate (compared to 150Mb and 66s uncompressed).
- Using a Cholesky factorisation further reduces calculation

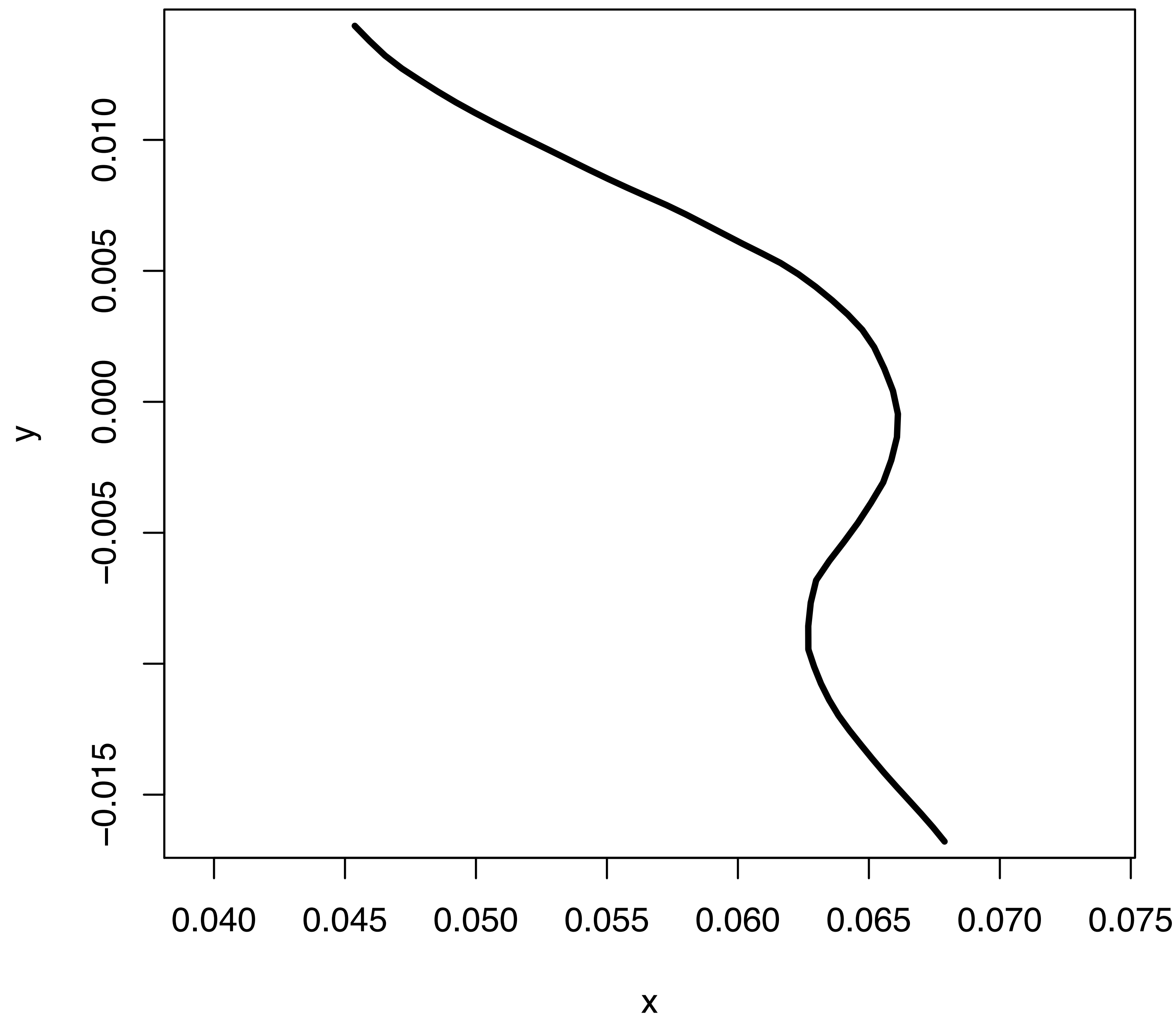
Models for curves



Curvature

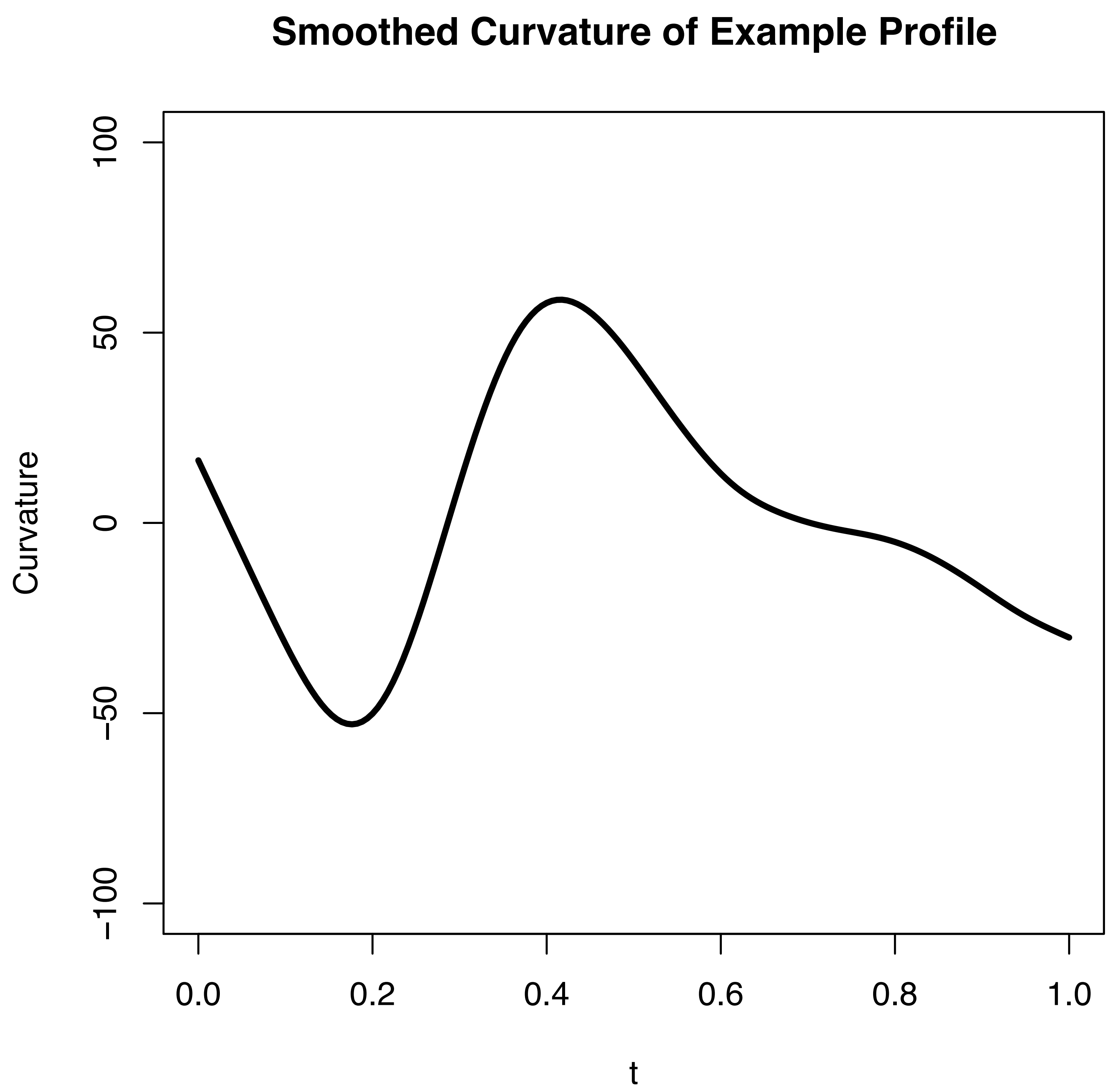
$$x(t), \quad y(t)$$

Example of a Control Nose Profile

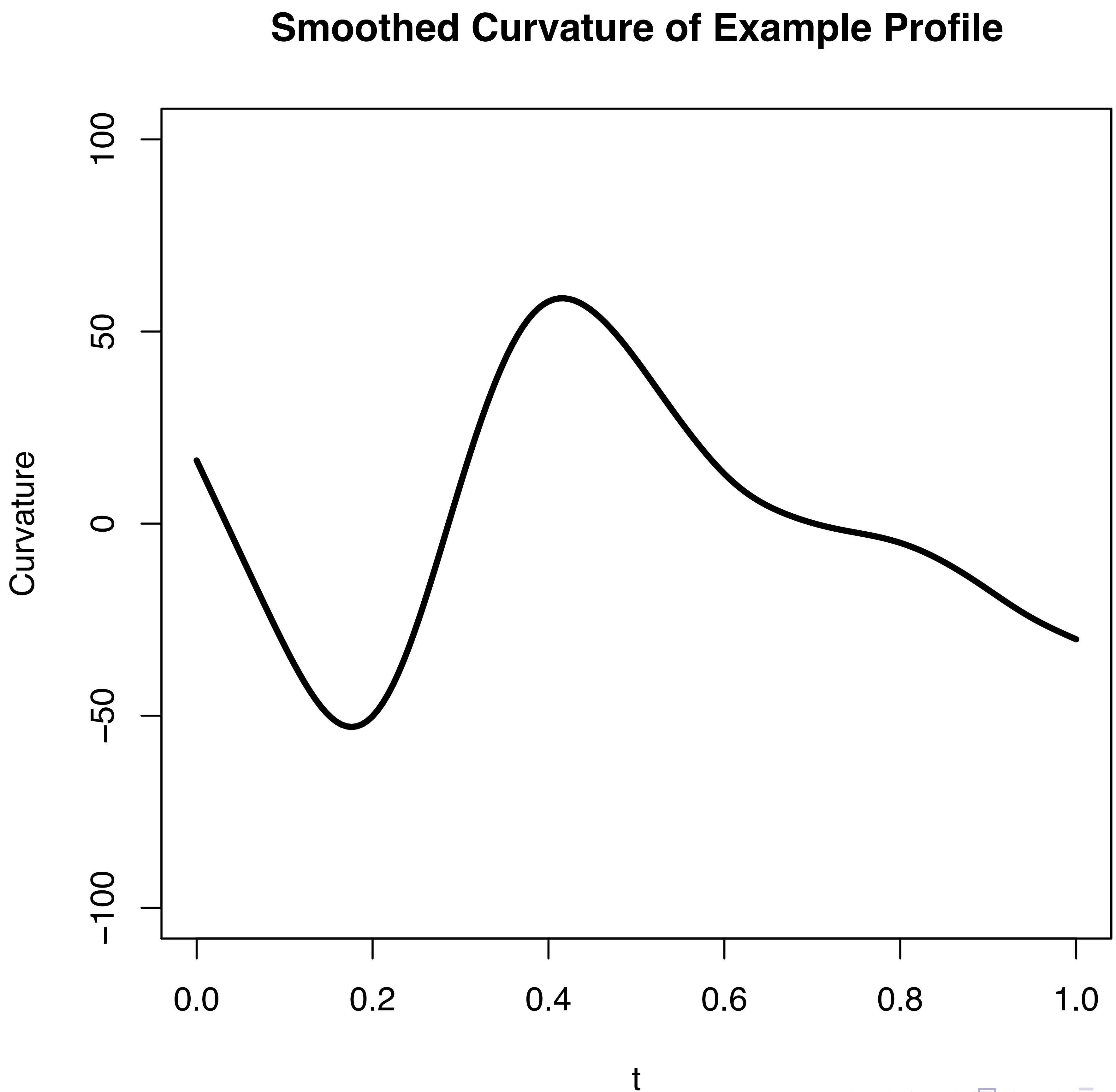


Curvature

$$\kappa(t) = \frac{x''(t)\,y'(t) - y''(t)\,x'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}$$

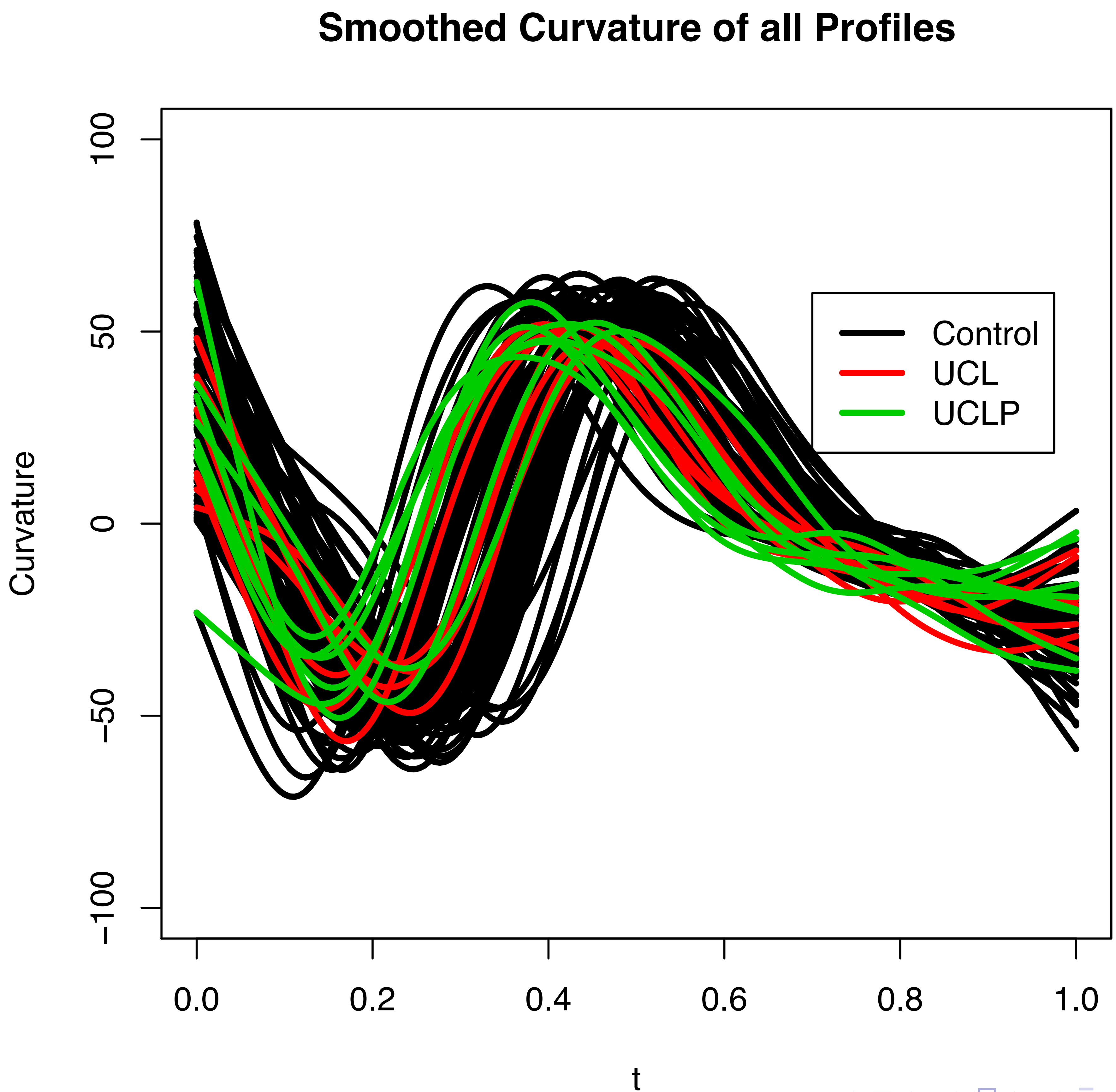


Curvature



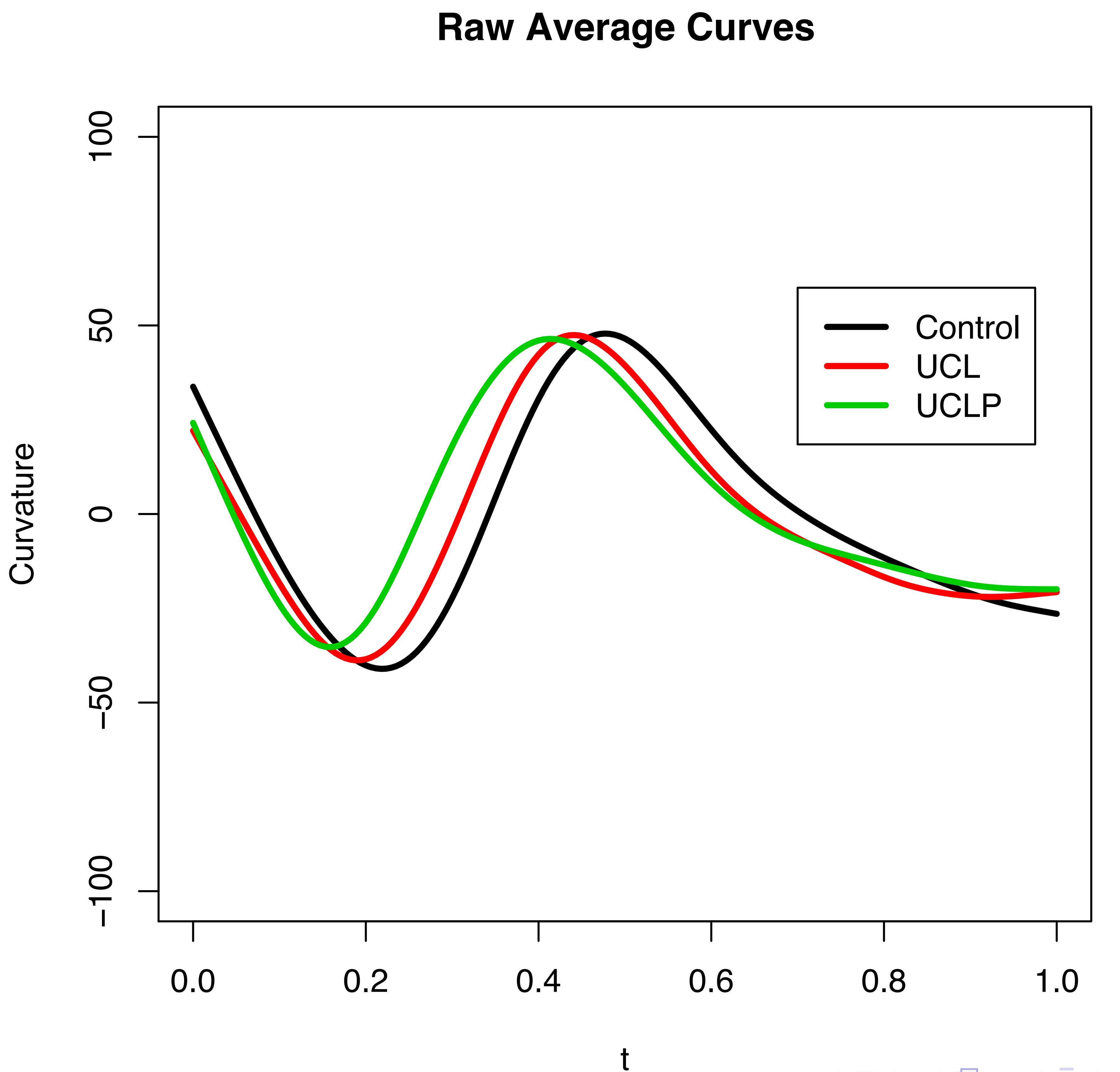
Smoothed Curvature of Example Profile
Curvature
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0
t

Curvature



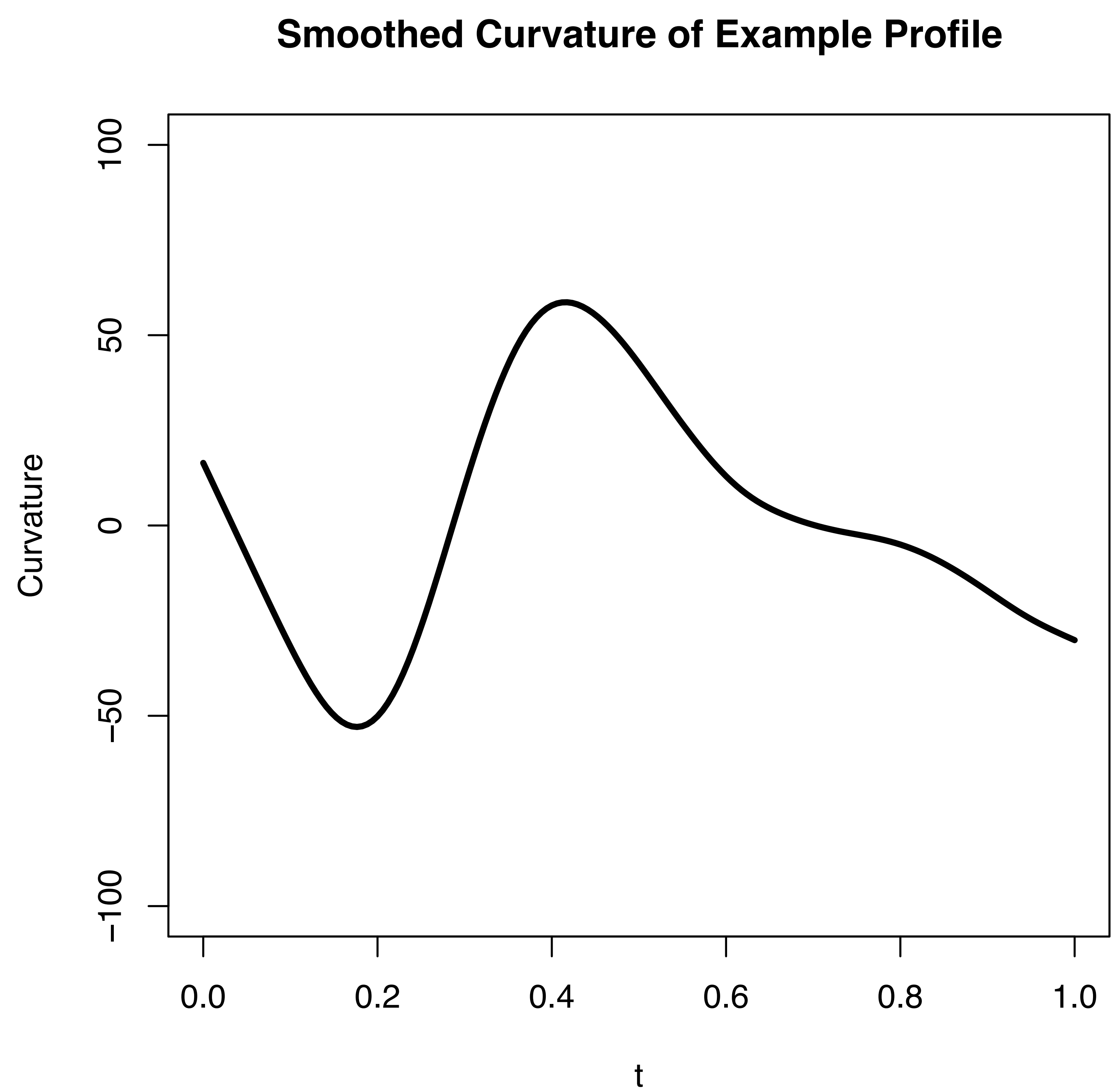
Smoothed Curvature of all Profiles
Curvature
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0
t
Control
UCL
UCLP

Curvature


Raw Average Curves
Control
UCL
UCLP
Curvature
t
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0

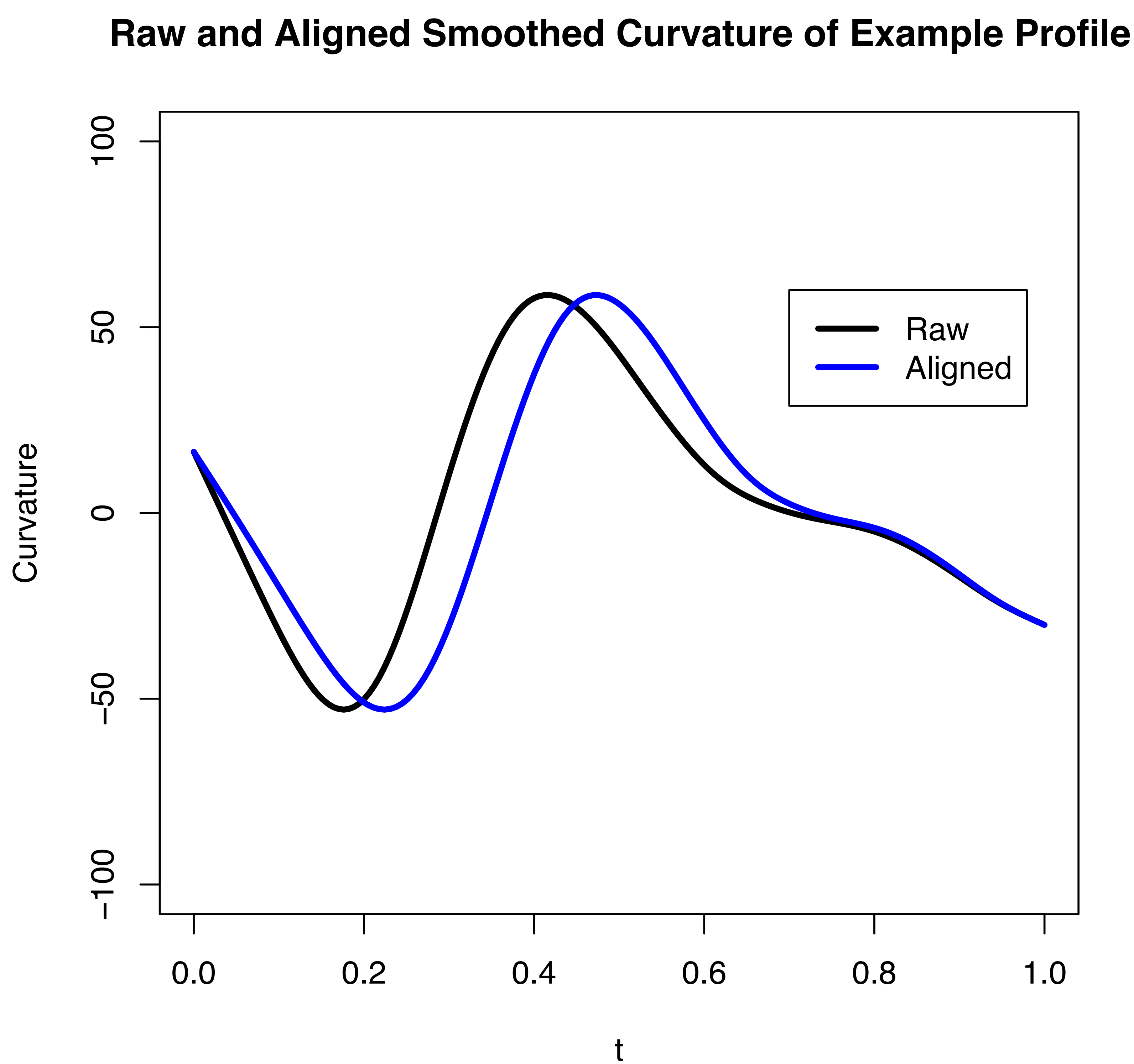
Curvature

$$\kappa(t)$$

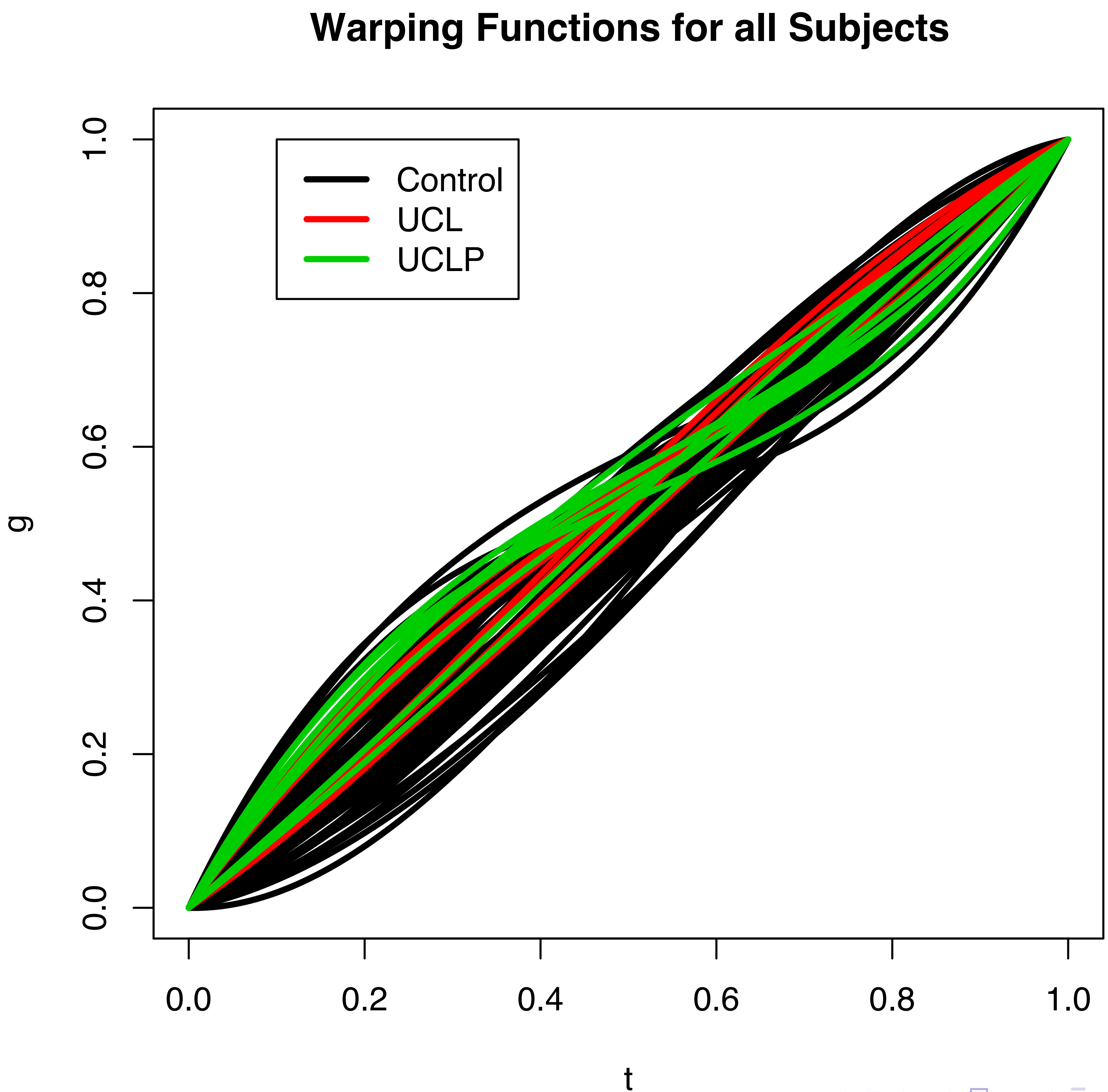


Curvature

$$\kappa(t) \quad \color{blue}{\kappa(g(t))}$$

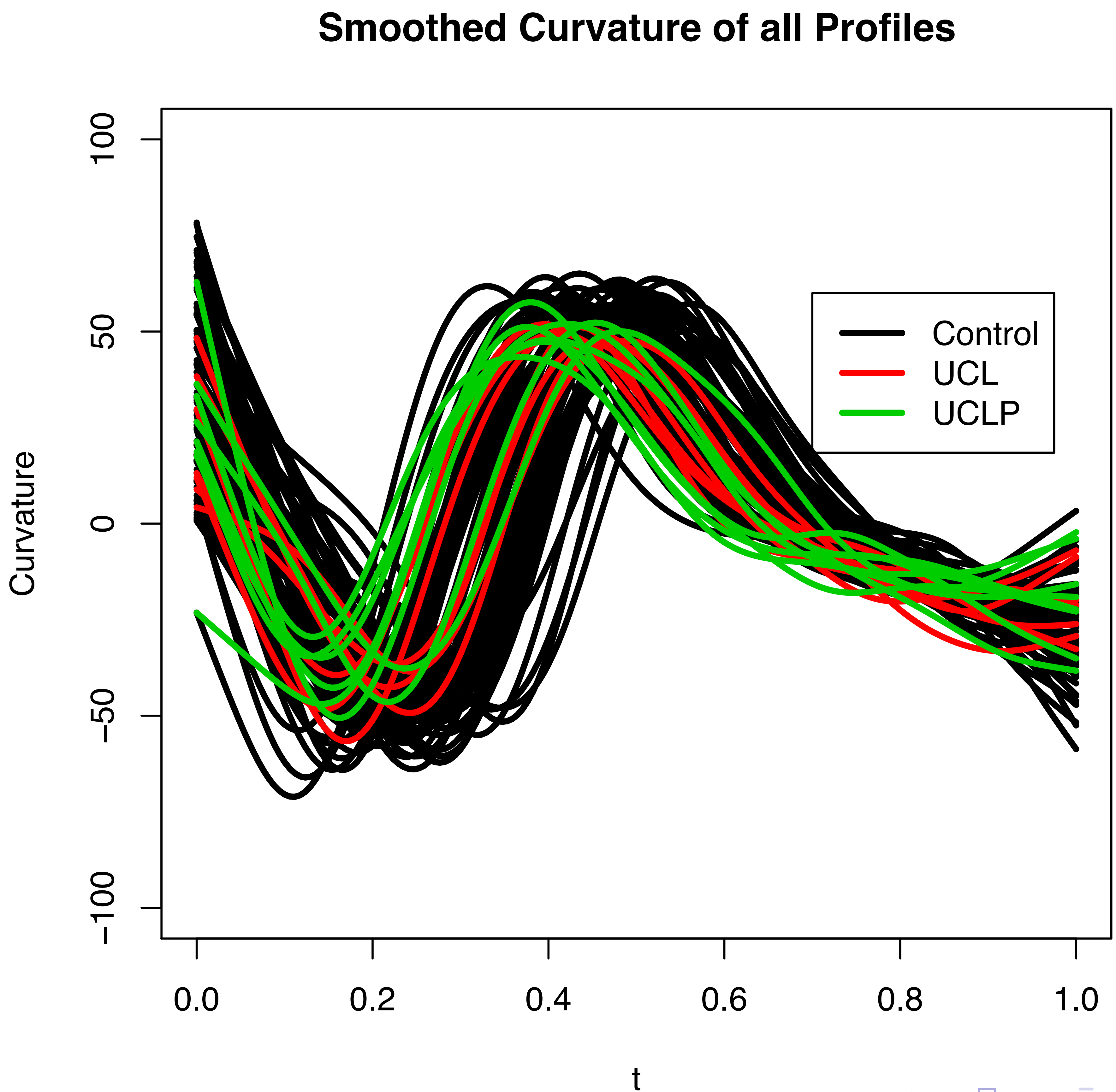


Curvature



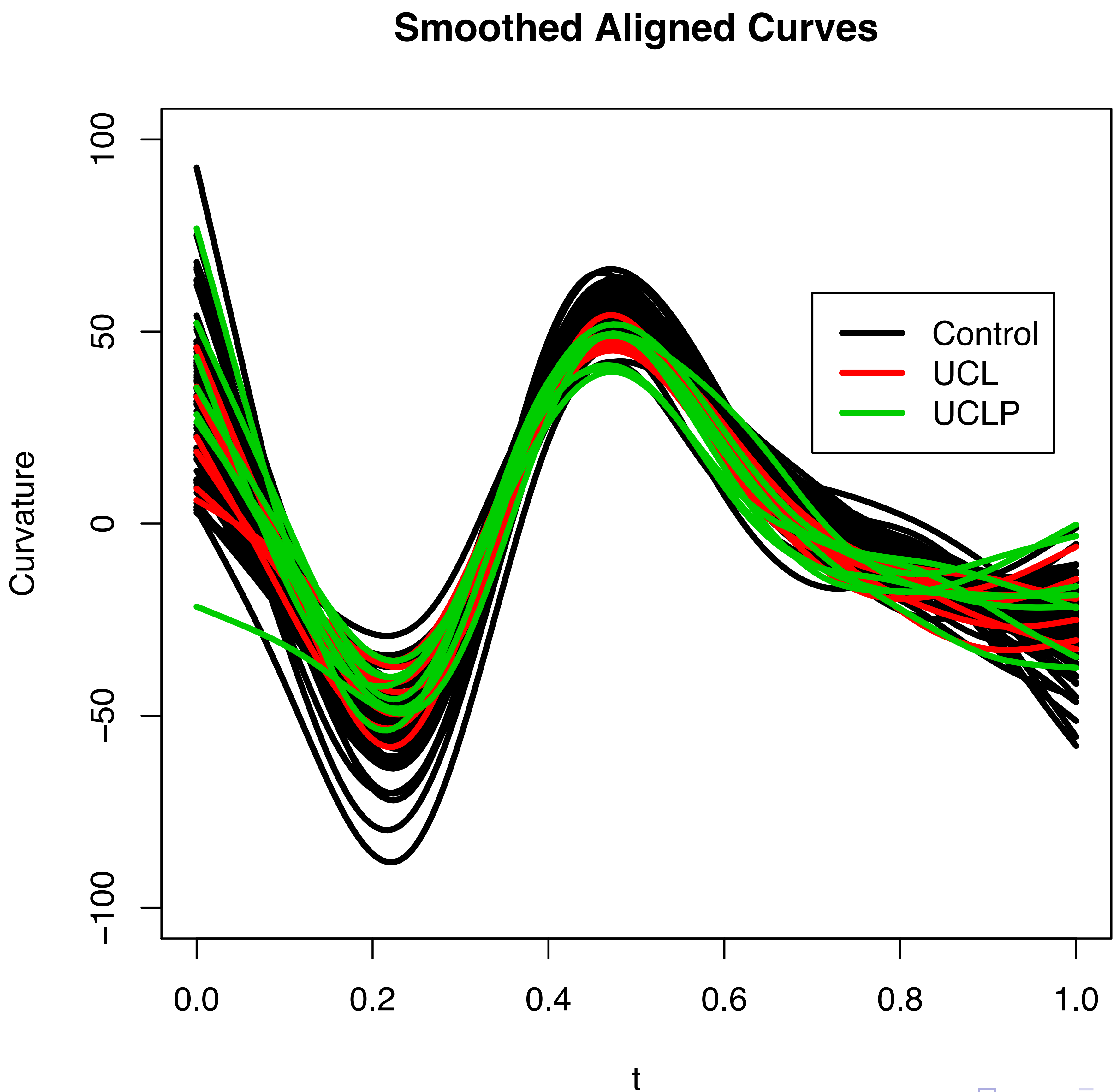
Warping Functions for all Subjects
Control
UCL
UCLP
g
t
0.0
0.2
0.4
0.6
0.8
1.0

Curvature

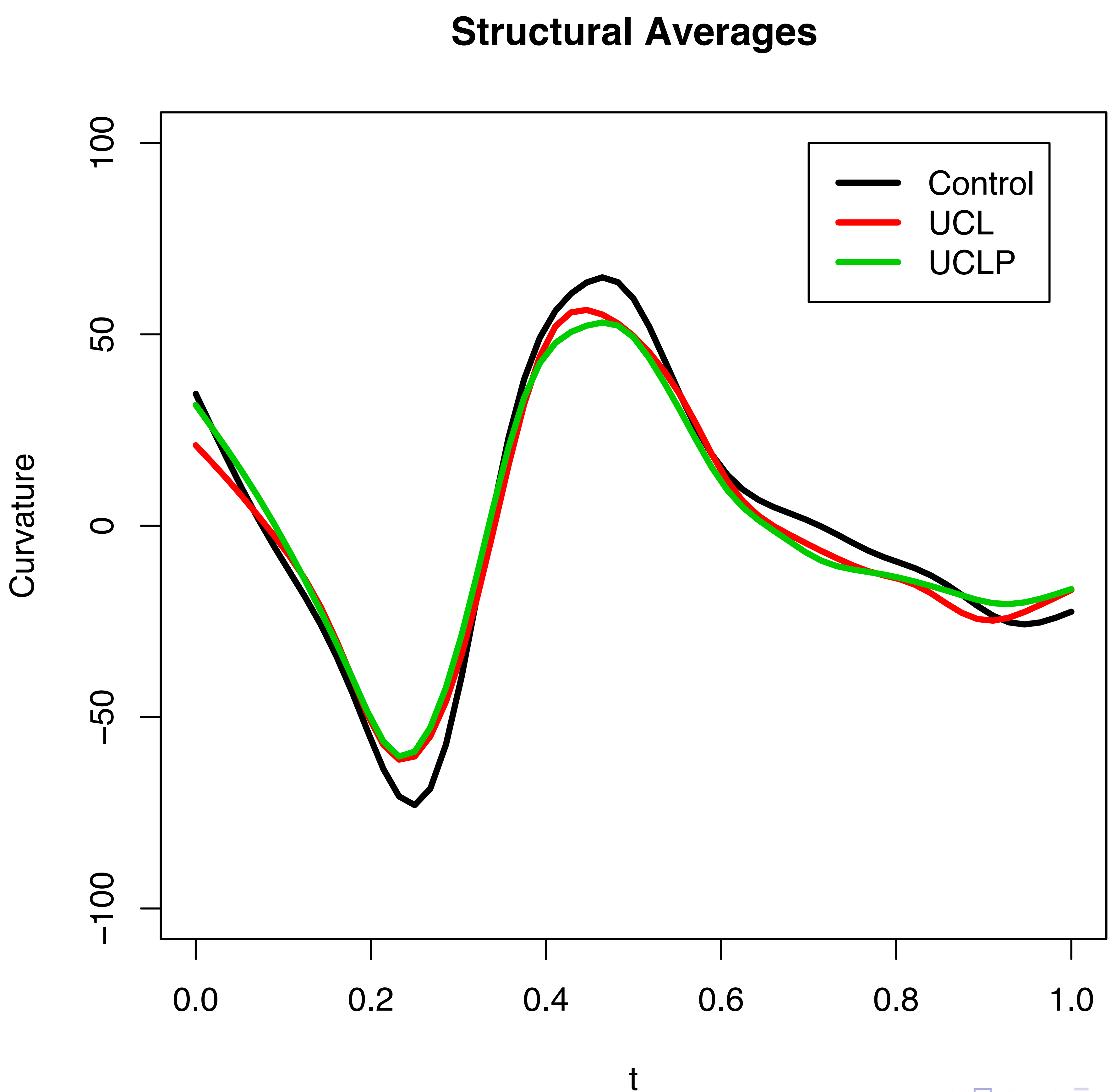


Smoothed Curvature of all Profiles
Curvature
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0
t
Control
UCL
UCLP

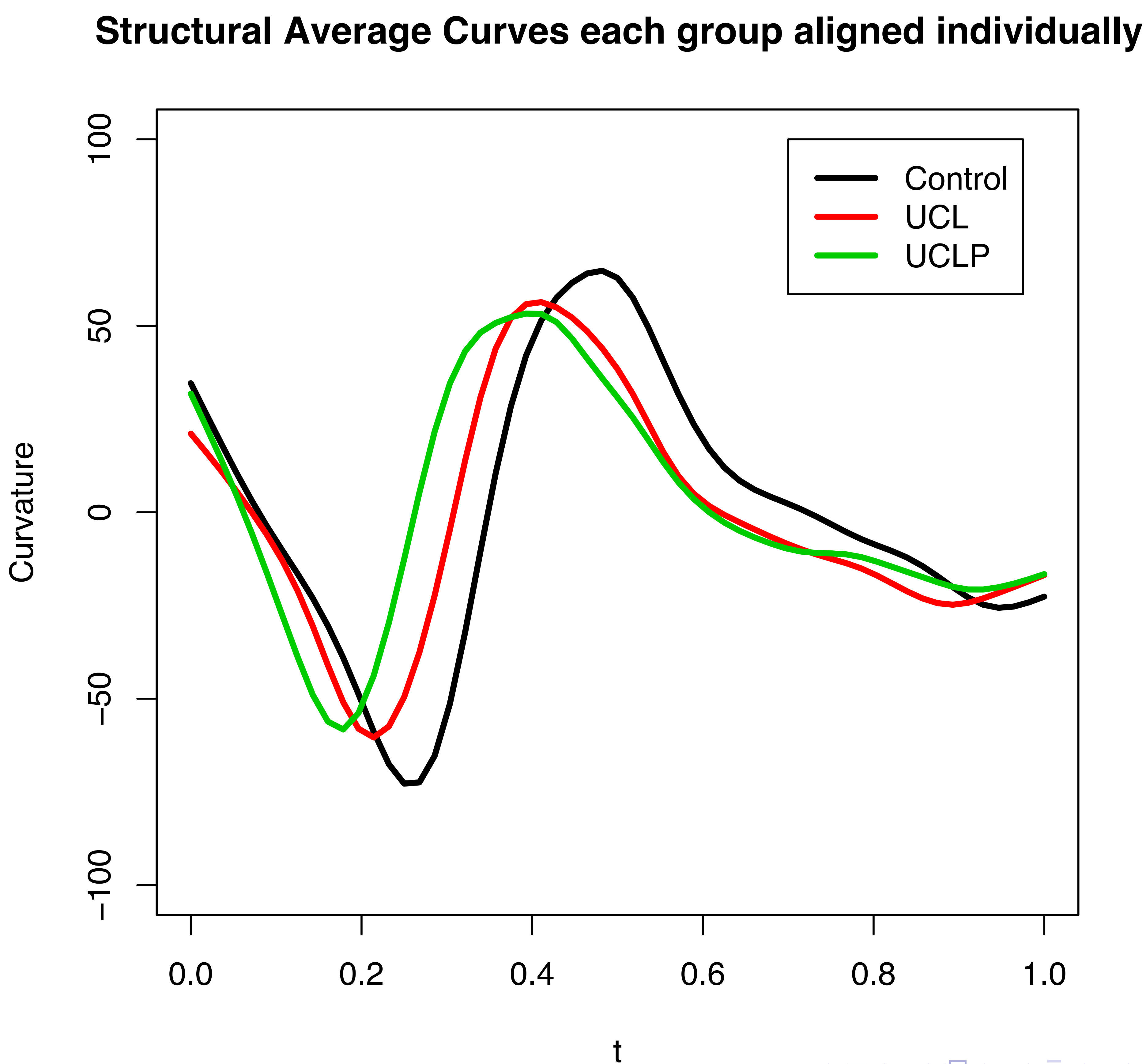
Curvature


Smoothed Aligned Curves
Curvature
t
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0
Control
UCL
UCLP

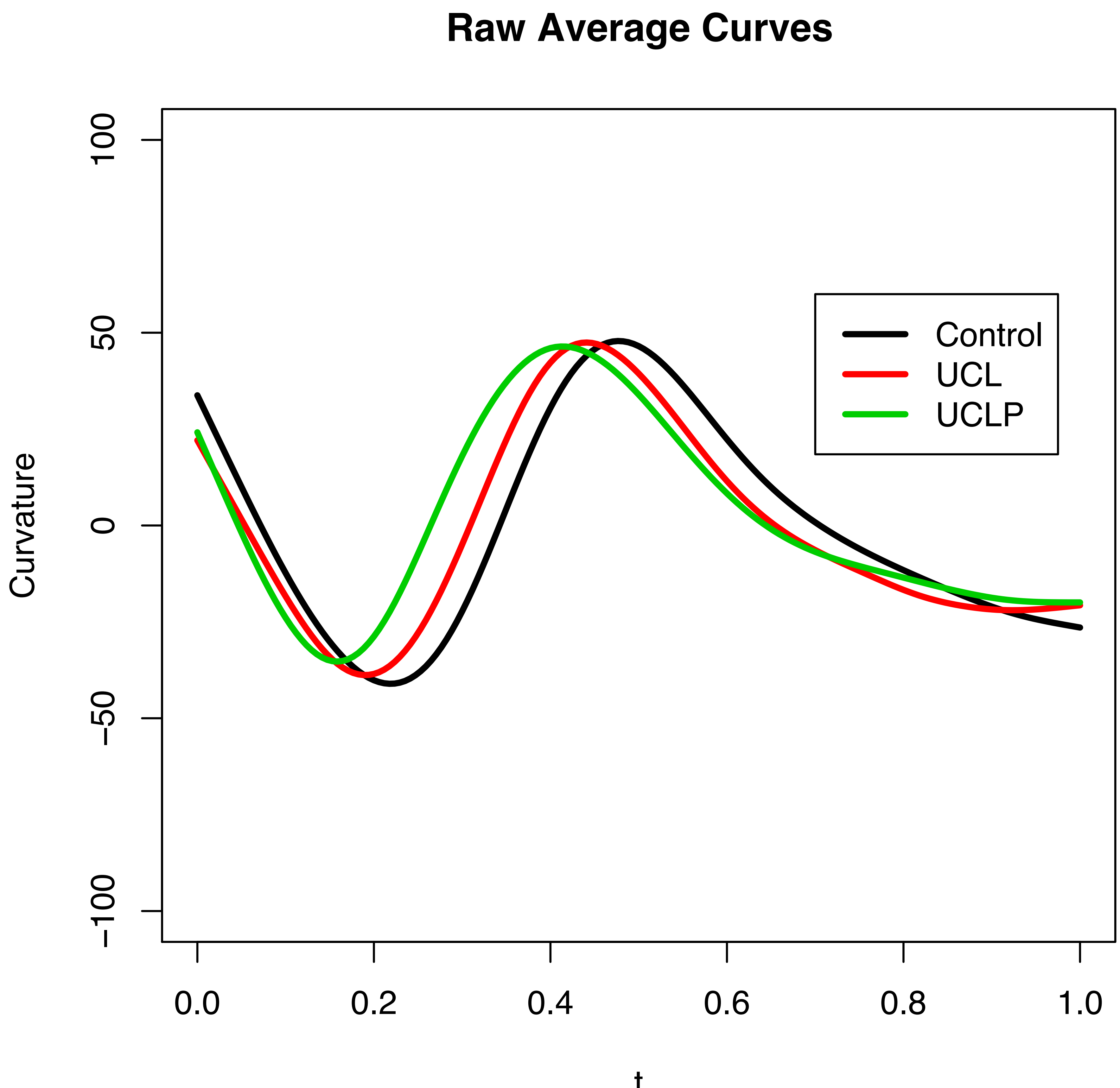
Curvature


Structural Averages
Control
UCL
UCLP
100
50
0
−50
−100
Curvature
0.0
0.2
0.4
0.6
0.8
1.0
t

Curvature


Structural Average Curves each group aligned individually
Control
UCL
UCLP
Curvature
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0
t

Curvature



Raw Average Curves
Control
UCL
UCLP
Curvature
t
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0