Session 5.1: spatio-temporal model for geostatistical data

Spatial and Spatio-Temporal Bayesian Models with `R-INLA`, University of São Paulo

30 September 2022

Learning Objectives

At the end of this session you should be able to:

- know the definition of spatio-temporal process in the geostatistics framework;
- use `inlabru` for implementing a (separable) space-time geostatistical model.

The topics treated in this lecture can be found in **Section 7.2** of the INLA book.

Outline

1. Spatio-temporal processes + a space-time hierarchical model for air pollution

2. Implementation of a spatio-temporal process using `inlabru`

Spatio-temporal processes + a space-time hierarchical model for air pollution

Spatio-temporal processes

- The concept of spatial process can be extended to the spatio-temporal case including a time dimension. The data are then defined by a process $\{y(s,t),(s,t) \in \mathcal{D} \subset \mathbb{R}^2 \times \mathbb{R}\}$ and are observed at n spatial locations and at T time points.

Spatio-temporal processes

- The concept of spatial process can be extended to the spatio-temporal case including a time dimension. The data are then defined by a process $\{y(s,t),(s,t)\in\mathcal{D}\subset\mathbb{R}^2\times\mathbb{R}\}$ and are observed at n spatial locations and at T time points.

- When spatio-temporal geostatistical data are considered, we need to define a valid **spatio-temporal covariance function** given by

$$\text{Cov}\left(y(\boldsymbol{s}_i,t),y(\boldsymbol{s}_j,u)\right)=\mathcal{C}(y_{it},y_{ju})$$

Spatio-temporal processes

- The concept of spatial process can be extended to the spatio-temporal case including a time dimension. The data are then defined by a process $\{y(s,t), (s,t) \in \mathcal{D} \subset \mathbb{R}^2 \times \mathbb{R}\}$ and are observed at n spatial locations and at T time points.

- When spatio-temporal geostatistical data are considered, we need to define a valid **spatio-temporal covariance function** given by

$$\text{Cov}\left(y(\boldsymbol{s}_i, t), y(\boldsymbol{s}_j, u)\right) = \mathcal{C}(y_{it}, y_{ju})$$

- If we assume **stationarity in space and time**, the space-time covariance function can be written as a function of the spatial Euclidean distance $\Delta_{ij} = ||\boldsymbol{s}_i - \boldsymbol{s}_j||$ and of the temporal lag $\Lambda_{tu} = |t - u|$ so that $\text{Cov}\left(y_{it}, y_{ju}\right) = \mathcal{C}(\Delta_{ij}, \Lambda_{tu})$.

Spatio-temporal processes

- The concept of spatial process can be extended to the spatio-temporal case including a time dimension. The data are then defined by a process $\{y(s,t),(s,t) \in \mathcal{D} \subset \mathbb{R}^2 \times \mathbb{R}\}$ and are observed at n spatial locations and at T time points.
- When spatio-temporal geostatistical data are considered, we need to define a valid **spatio-temporal covariance function** given by

$$\text{Cov}\left(y(\boldsymbol{s}_i,t), y(\boldsymbol{s}_j,u)\right) = \mathcal{C}(y_{it}, y_{ju})$$

- If we assume **stationarity in space and time**, the space-time covariance function can be written as a function of the spatial Euclidean distance $\Delta_{ij} = ||\boldsymbol{s}_i - \boldsymbol{s}_j||$ and of the temporal lag $\Lambda_{tu} = |t-u|$ so that $\text{Cov}\left(y_{it}, y_{ju}\right) = \mathcal{C}(\Delta_{ij}, \Lambda_{tu})$.
- If we assume **separability** the stationary space-time covariance function is decomposed into the product (or the sum) of a purely spatial and a purely temporal term:

$$\text{Cov}\left(y_{it}, y_{ju}\right) = \mathcal{C}_1(\Delta_{ij})\mathcal{C}_2(\Lambda_{tu})$$

Hierarchical spatio-temporal model for PM concentration

- We present a spatio-temporal model for particulate matter (PM10) concentration data measured daily (in $\mu g/m^3$).
- The data refer to Piemonte region (Italy) for the period from October 2005 to March 2006 (daily data).
- **Main aims**:
  - predict PM concentration in the considered continuous spatial domain, where no monitoring stations are displaced;
  - evaluate the effect of some covariates (e.g. wind speed, precipitation, temperature, emissions, altitude);
  - compute the probability of exceeding a specific threshold (e.g. $50\mu g/m^3$ fixed by the European Community for health protection).
- The spatio-temporal model we specify here is widely adopted in the air quality literature thanks to its flexibility in modeling relevant covariates as well as correlation in space and time (see Cameletti, Lindgren, Simpson, and Rue (2013); Fioravanti, Martino, Cameletti, and Cattani (2021)).

Hierarchical spatio-temporal model for PM concentration

- We denote by y_{it} the logarithm of PM10 concentrations measured at site $\boldsymbol{s}_i$, with $i = 1, \ldots, n = 24$, and day $t = 1, \ldots, T = 182$.
- The following distribution is assumed for the observations:

$$y_{it} \sim \text{Normal}(\eta_{it}, \sigma_e^2)$$

where σ_e^2 is the variance of the measurement error defined by a Gaussian white-noise process, both serially and spatially uncorrelated.

Hierarchical spatio-temporal model for PM concentration

- We denote by y_{it} the logarithm of PM10 concentrations measured at site $\boldsymbol{s}_i$, with $i = 1, \ldots, n = 24$, and day $t = 1, \ldots, T = 182$.

- The following distribution is assumed for the observations:

$$y_{it} \sim \text{Normal}(\eta_{it}, \sigma_e^2)$$

  where σ_e^2 is the variance of the measurement error defined by a Gaussian white-noise process, both serially and spatially uncorrelated.

- The linear predictor is given by

$$\eta_{it} = b_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \omega_{it}$$

  where b_0 is the intercept and $\beta_1, \ldots, \beta_M$ are the linear effects related to meteorological and orographical covariates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$.

Hierarchical spatio-temporal model for PM concentration

- The term ω_{it} refers to the **latent spatio-temporal process** (i.e. the true unobserved level of pollution) which changes in time with first order autoregressive dynamics and spatially correlated innovations:

$$\omega_{it} = a\omega_{i(t-1)} + \xi_{it}$$

with $t = 2, \ldots, T$, $|a| < 1$, $\omega_{i1} \sim \text{Normal}\left(0, \sigma^2/(1-a^2)\right)$.

Hierarchical spatio-temporal model for PM concentration

- The term ω_{it} refers to the **latent spatio-temporal process** (i.e. the true unobserved level of pollution) which changes in time with first order autoregressive dynamics and spatially correlated innovations:

$$\omega_{it} = a\omega_{i(t-1)} + \xi_{it}$$

with $t = 2, \ldots, T$, $|a| < 1$, $\omega_{i1} \sim \text{Normal}\left(0, \sigma^2/(1-a^2)\right)$.

- The term ξ_{it} is a zero-mean **Gaussian field**, assumed to be **temporally independent** and characterized by the following spatio-temporal covariance function:

$$\text{Cov}\left(\xi_{it}, \xi_{ju}\right) = \begin{cases} 0 & \text{if} \quad t \neq u \\ \text{Cov}(\xi_i, \xi_j) & \text{if} \quad t = u \end{cases}$$

for $i \neq j$, where $\text{Cov}(\xi_i, \xi_j)$ is given by Matérn spatial covariance function.

- This model is characterized by a **separable spatio-temporal covariance** as it can be rewritten as the product of a purely spatial and a purely temporal covariance function (see Cameletti, Ignaccolo, and Bande (2011)).

Hierarchical spatio-temporal model for PM concentration

- For each time point $\boldsymbol{\xi}_t \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Sigma})$ and through the SPDE approach

$$\boldsymbol{\xi}_t \rightarrow \tilde{\boldsymbol{\xi}}_t \sim \text{Normal}(\mathbf{0}, \boldsymbol{Q}_S^{-1})$$

  where the precision matrix $\boldsymbol{Q}_S^{-1}$ comes from the SPDE representation. The matrix $\boldsymbol{Q}_S^{-1}$ does not change in time - due to the serial independence hypothesis - and its dimension is given by the number of vertices of the domain triangulation.

Hierarchical spatio-temporal model for PM concentration

- For each time point $\boldsymbol{\xi}_t \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Sigma})$ and through the SPDE approach

$$\boldsymbol{\xi}_t \to \tilde{\boldsymbol{\xi}}_t \sim \text{Normal}(\mathbf{0}, \boldsymbol{Q}_S^{-1})$$

  where the precision matrix $\boldsymbol{Q}_S^{-1}$ comes from the SPDE representation. The matrix $\boldsymbol{Q}_S^{-1}$ does not change in time - due to the serial independence hypothesis - and its dimension is given by the number of vertices of the domain triangulation.

- The joint distribution of the Tn-dimensional GMRF $\boldsymbol{\omega} = (\boldsymbol{\omega}_1', \ldots, \boldsymbol{\omega}_T')'$ is

$$\boldsymbol{\omega} \sim \text{Normal}(\mathbf{0}, \boldsymbol{Q}^{-1})$$

  with $\boldsymbol{Q} = \boldsymbol{Q}_T \otimes \boldsymbol{Q}_S$, where $\otimes$ denotes the Kronecker product and $\boldsymbol{Q}_T$ is the T-dimensional precision matrix of the AR(1) process.

- For the considered model the latent process is given by $\boldsymbol{\theta} = \{\boldsymbol{\omega}, b_0, \beta_1, \ldots, \beta_M\}$ while the hyperparameter vector is $\psi = (\sigma_e^2, a, \sigma^2, r)$.

Implementation of a spatio-temporal process using `inlabru`

Piemonte data

The data are **PM10 concentrations** in 24 monitoring stations in Piemonte region in Italy for a period of 182 days from 2005-10-01 to 2006-03-31.

Data | Boundary

```
> df = readRDS("./data/Piemonte_Data.rds")
> class(df)
```

```
[1] "data.frame"
```

```
> head(df)
```

```
  Station.ID     Date     A   UTMX    UTMY   WS   TEMP   HMIX PREC   EMI PM10  logPM10 time
1          1 01/10/05  95.2 469.45 4972.85 0.90 288.81 1294.6    0 26.05   28 3.332205    1
2          2 01/10/05 164.1 423.48 4950.69 0.82 288.67 1139.8    0 18.74   22 3.091042    1
3          3 01/10/05 242.9 490.71 4948.86 0.96 287.44 1404.0    0  6.28   17 2.833213    1
4          4 01/10/05 149.9 437.36 4973.34 1.17 288.63 1042.4    0 29.35   25 3.218876    1
5          5 01/10/05 405.0 426.44 5045.66 0.60 287.63 1038.7    0 32.19   20 2.995732    1
6          6 01/10/05 257.5 394.60 5001.18 1.02 288.59 1048.3    0 34.24   41 3.713572    1
```

```
> # select only the first 50 days for reducing the computational load
> df = df[df$time <= 50,]
```

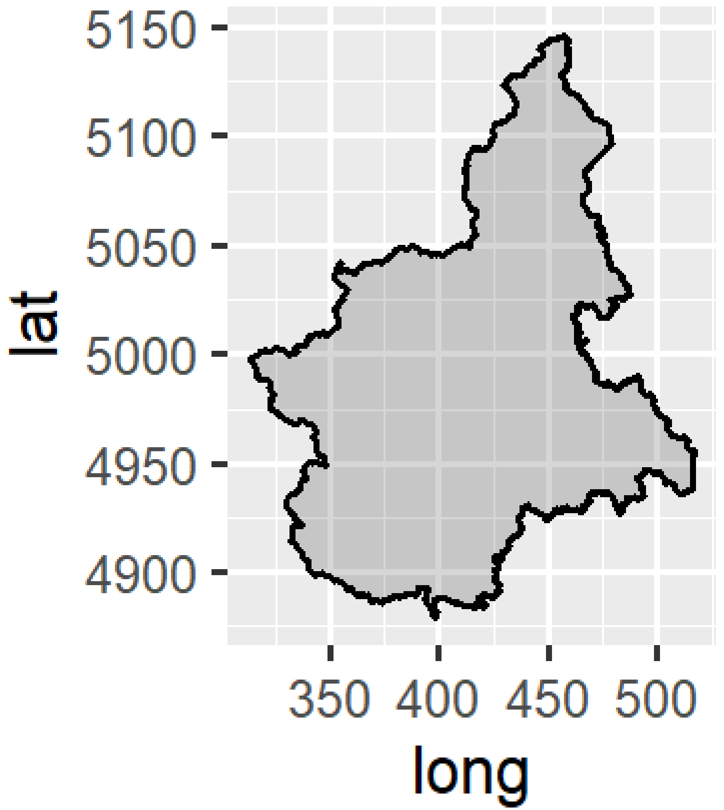
The data are **PM10 concentrations** in 24 monitoring stations in Piemonte region in Italy for a period of 182 days from 2005-10-01 to 2006-03-31.

Data | **Boundary**

```
> library(tidyverse)
> library(inlabru)
> border = readRDS("./data/Piemonte_Border.rds")
> class(border)
```

```
[1] "SpatialPolygons"
attr(,"package")
[1] "sp"
```

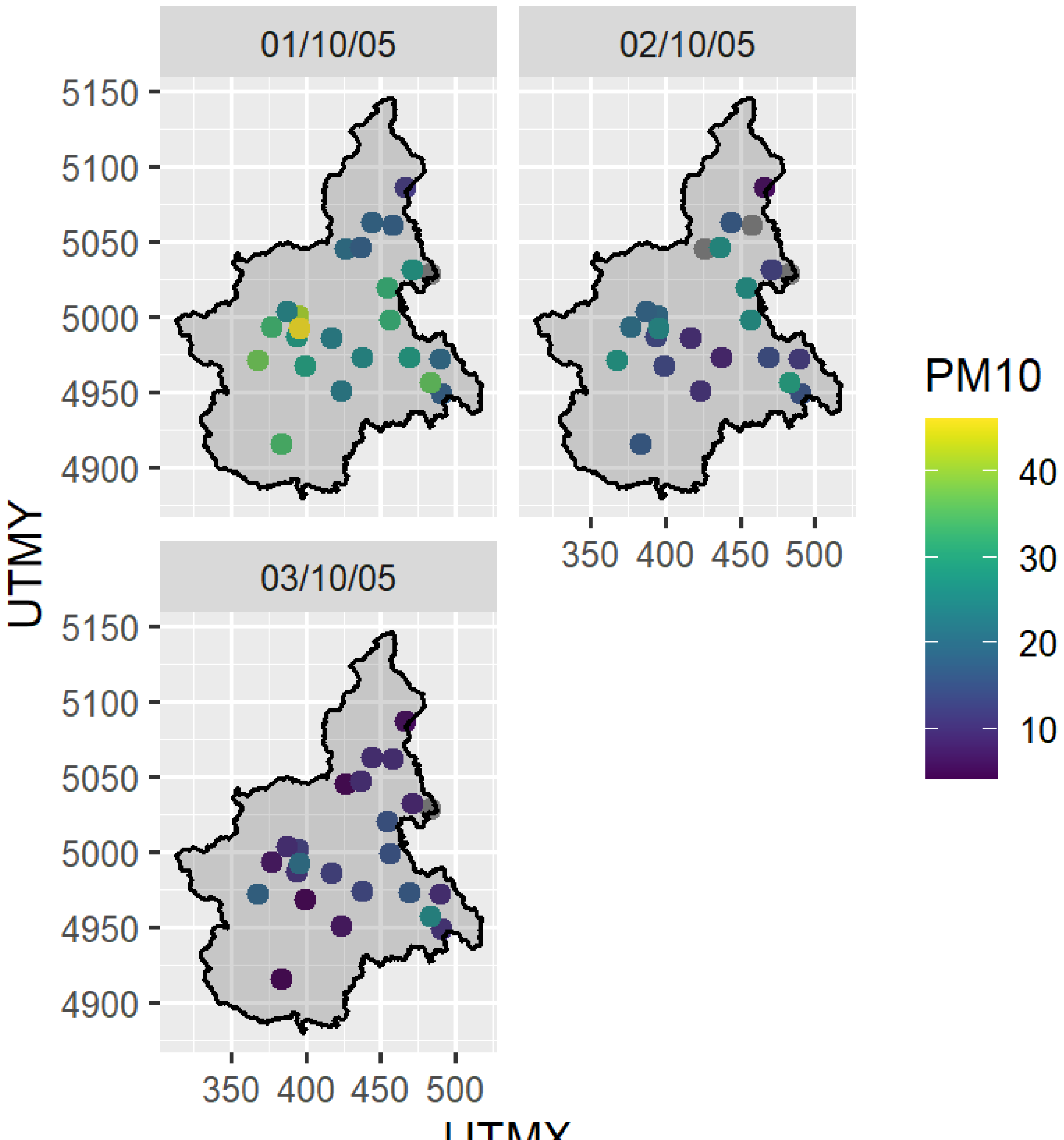
```
> ggplot()+
+   gg(border) +
+   coord_equal()
```



PM10 data

We plot PM10 concentrations measured in the 24 monitoring stations in the first **3 days** of the time series.

```
> library(tidyverse)
> library(viridis)
>
> df %>%
+   filter(time<=3) %>%
+   ggplot() +
+   geom_point(aes(UTMX, UTMY, color = PM10), siz
+   facet_wrap(.~ Date, ncol = 2, nrow = 2) +
+   scale_color_viridis() +
+   coord_equal() +
+   gg(border)
```

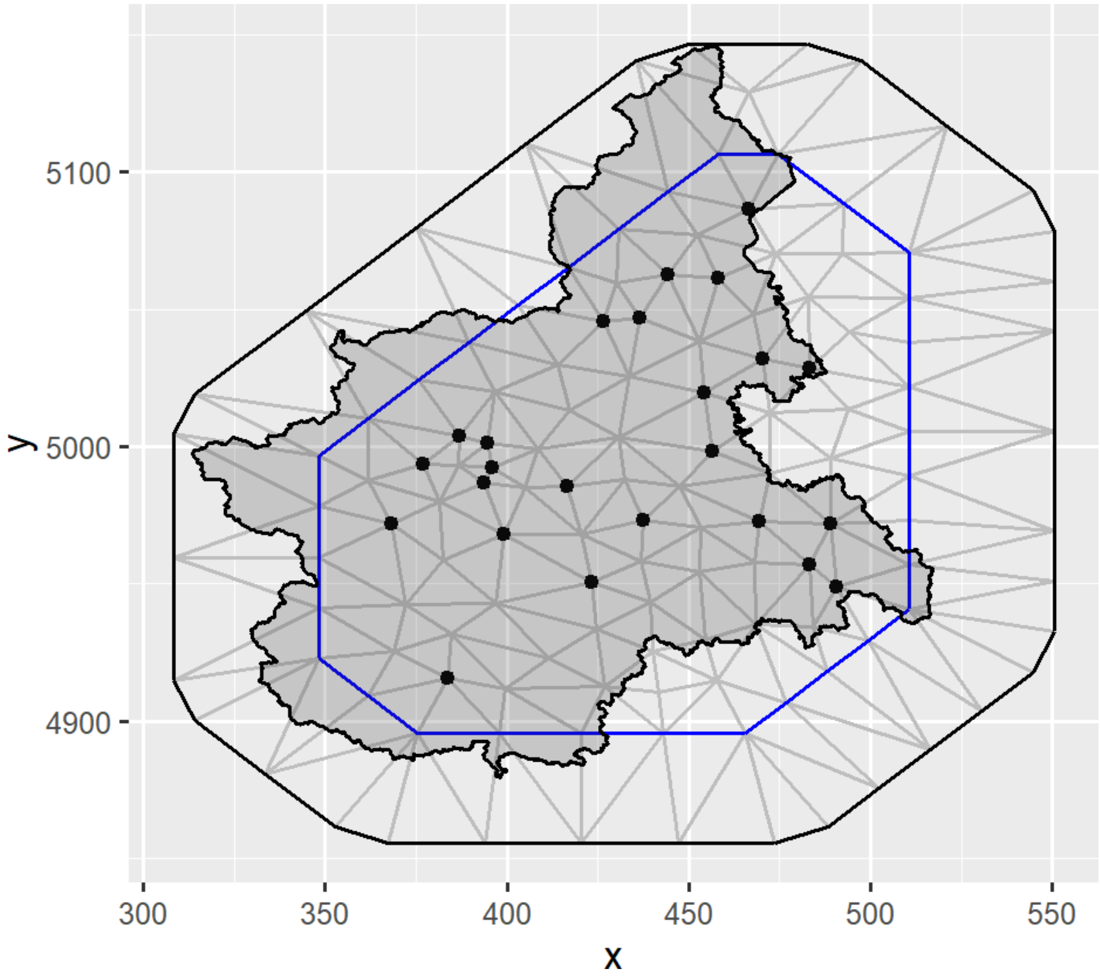


Create the mesh and the SPDE model

For the example we choose a quite rough mesh (starting from the monitoring stations):

```
> library(INLA)
> mesh = inla.mesh.2d(loc = cbind(df$UTMX, df$UTM
+                     offset = c(20, 40),
+                     max.edge = c(30, 50))
```

```
> ggplot() +
+   gg(mesh) +
+   geom_point(data = df, aes(UTMX, UTMY)) +
+   gg(border)
```



Given the mesh, it is now possible to create the SPDE model using the `inla.spde2.matern` function:

```
> spde = inla.spde2.matern(mesh = mesh)
> spde$n.spde #n. of mesh vertices
```

```
[1] 136
```

Define the model components using inlabru

We run here the model using `inlabru`. If you are interested in the alternative version based on the `inla.stack` approach, see Section 7.2 of the INLA book.

We first transform the `df` data frame into a **spatial object** (`SpatialPointsDataFrame`)

```
> coordinates(df) = c("UTMX","UTMY")
> class(df)
```

```
[1] "SpatialPointsDataFrame"
attr(,"package")
[1] "sp"
```

and then define the model **components**

```
> cmp  = logPM10 ~ Intercept(1) +
+   SPDE(coordinates, model = spde,
+         group = time, control.group = list(model = "ar1")) +
+   A + #dem(A, model = "linear") +
+   TEMP #temp(TEMP, model = "linear")
```

Using the options `group` and `control.group` we specify that at each time point the spatial locations are linked by the `spde` model object, while across time the process evolves according to an AR(1) dynamics.

Fit the space-time model using inlabru

We then define the **likelihood**

```
> lik = like(formula = logPM10 ~ Intercept + SPDE + A + TEMP,
+             family = "gaussian",
+             data = df)
```

and finally run the model with `bru`

```
> fit = bru(cmp, lik)
```

```
> fit$summary.fixed[,c("mean","0.025quant","0.975quant")]
```

```
                  mean    0.025quant  0.975quant
Intercept -2.330542652 -13.198261436 8.647721743
A         -0.001330801  -0.003800033 0.001135421
TEMP       0.021393137  -0.016957290 0.059374675
```

Prediction at the station locations

We are interested in predicting PM10 concentrations (log concentrations) at the monitoring station locations. As introduced in Section 4.2, we will use the `predict` function:

Compute predictions | Plot predictions

```
> pred_at_station = predict(fit, df, ~ Intercept + SPDE + A + TEMP, n.samples = 1000)
> head(pred_at_station)
```

```
        coordinates Station.ID     Date     A   WS   TEMP   HMIX PREC   EMI
1 (469.45, 4972.85)          1 01/10/05  95.2 0.90 288.81 1294.6    0 26.05
2 (423.48, 4950.69)          2 01/10/05 164.1 0.82 288.67 1139.8    0 18.74
3 (490.71, 4948.86)          3 01/10/05 242.9 0.96 287.44 1404.0    0  6.28
4 (437.36, 4973.34)          4 01/10/05 149.9 1.17 288.63 1042.4    0 29.35
5 (426.44, 5045.66)          5 01/10/05 405.0 0.60 287.63 1038.7    0 32.19
6  (394.6, 5001.18)          6 01/10/05 257.5 1.02 288.59 1048.3    0 34.24
  PM10  logPM10 time     mean        sd   q0.025     q0.5   q0.975   median
1   28 3.332205    1 3.343125 0.3370873 2.677242 3.336889 3.978277 3.336889
2   22 3.091042    1 3.001355 0.3617346 2.303291 3.003864 3.687795 3.003864
3   17 2.833213    1 2.999060 0.4023094 2.213962 2.995986 3.807268 2.995986
4   25 3.218876    1 3.188600 0.3527893 2.473646 3.188574 3.855951 3.188574
5   20 2.995732    1 2.882621 0.5196310 1.876061 2.866921 3.870829 2.866921
6   41 3.713572    1 3.560108 0.4107219 2.727766 3.566215 4.344945 3.566215
  mean.mc_std_err sd.mc_std_err
1      0.01065964   0.007456145
```

We are interested in predicting PM10 concentrations (log concentrations) at the monitoring station locations. As introduced in Section 4.2, we will use the `predict` function:

Compute predictions | **Plot predictions**

Select 2 stations

```
> sel = c(1, 7, 10, 19, 23, 24)
```

and plot the observed/predicted time series:

```
> as.data.frame(pred_at_station) %>%
+   dplyr::filter(Station.ID %in% sel) %>%
+   ggplot() +
+   geom_line(aes(time, logPM10, group = Station.
+   geom_line(aes(time, mean, group = Station.ID)
+   geom_ribbon(aes(time, ymin = q0.025, ymax = c
+   facet_wrap(.~Station.ID)
```

Prediction at the grid locations

We want to predict the concentration (log concentration) of PM10 for the locations in the Piemonte grid and for the first 3 days. To do this we need the values of the altitude and temperature for every point of interest in both space (regular grid) and time.

Grid data | Altitude | Temperature

```
> covariate_grid = readRDS("./data/covariate_grid.rds")
> class(covariate_grid)
```

```
[1] "SpatialPixelsDataFrame"
attr(,"package")
[1] "sp"
```

```
> head(covariate_grid@data)
```

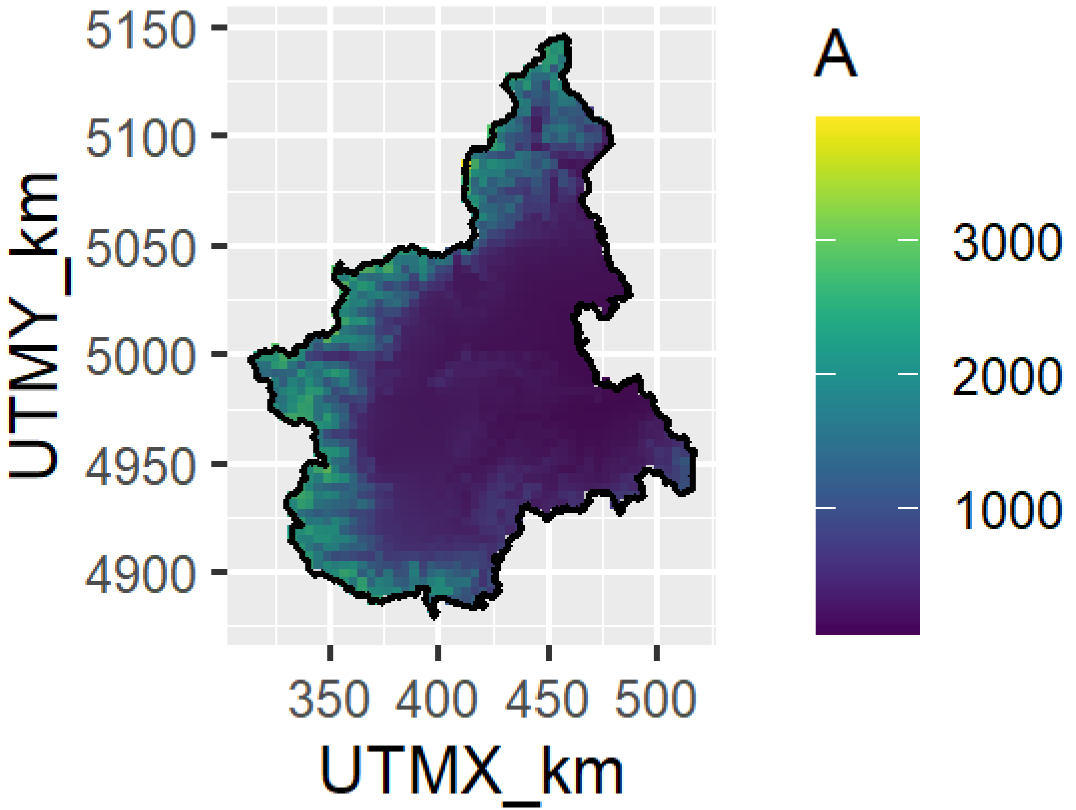
```
         A time     TEMP
1 1775.106    1 280.4226
2 1775.106    1 280.4540
3 1884.536    1 279.7775
4 2270.879    1 277.3138
5 2027.098    1 278.9060
6 1926.721    1 279.5709
```

Prediction at the grid locations

We want to predict the concentration (log concentration) of PM10 for the locations in the Piemonte grid and for the first 3 days. To do this we need the values of the altitude and temperature for every point of interest in both space (regular grid) and time.

Grid data | **Altitude** | Temperature

```
> ggplot() +
+ gg(covariate_grid, aes(fill=A)) +
+ gg(border) +
+ coord_equal() +
+ scale_fill_viridis()
```

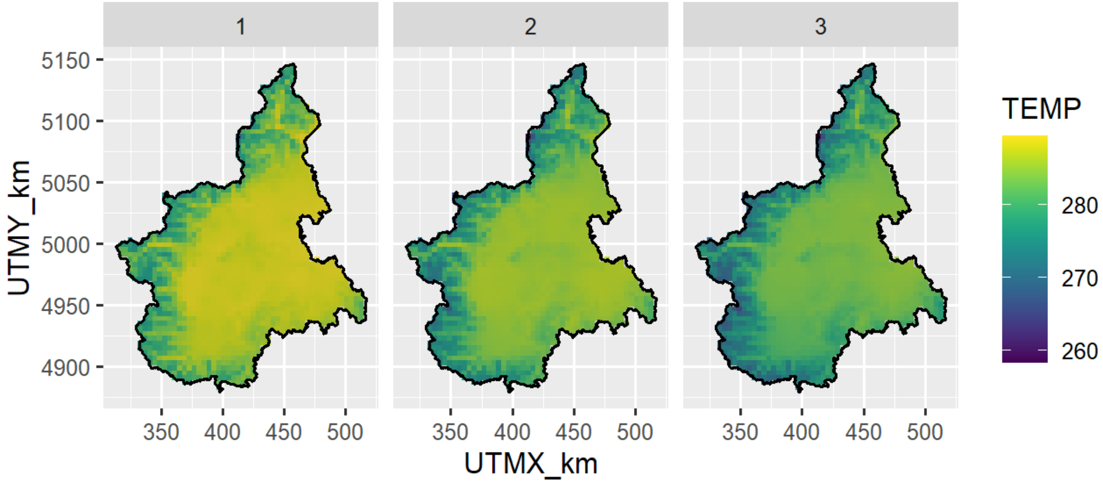


Prediction at the grid locations

We want to predict the concentration (log concentration) of PM10 for the locations in the Piemonte grid and for the first 3 days. To do this we need the values of the altitude and temperature for every point of interest in both space (regular grid) and time.

Grid data | Altitude | **Temperature**

```
> ggplot() +
+ gg(covariate_grid, aes(fill=TEMP)) +
+ facet_wrap(~ time) +
+ gg(border) +
+ coord_equal() +
+ scale_fill_viridis()
```



Prediction at the grid locations

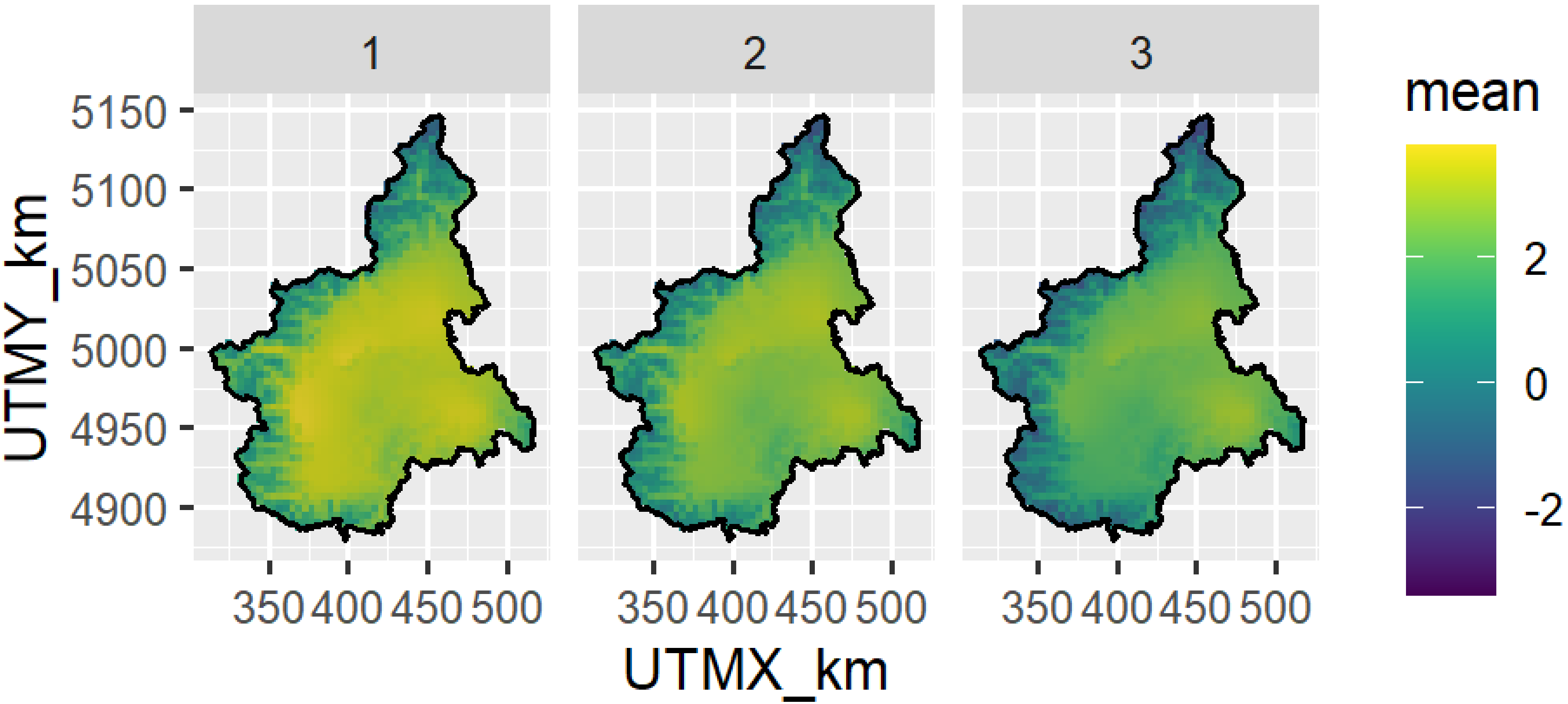
- We sample from the fitted model in order to inspect the PM10 field. As described in Section 2.2, the function `predict` is used for sampling from the posterior predictive distributions and computing posterior summary statistics. In this case we use the space-time grid (`covariate_grid`) introduced before.

```
> pred = predict(fit, covariate_grid,
+                ~ Intercept + SPDE + A + TEMP,
+                seed = 2, n.samples = 1000)
> head(pred@data)
```

```
           A time     TEMP      mean       sd    q0.025      q0.5   q0.975
174 1603.381    1 282.4919 2.0733251 1.913759 -1.528344 2.0400210 6.028608
175 1709.620    1 281.8724 1.9032338 2.047469 -1.962650 1.8500675 6.088131
176 2333.712    1 277.6419 0.9774478 2.774251 -4.226629 0.8842236 6.575013
246 1554.806    1 282.8573 2.1346265 1.848591 -1.398453 2.0925785 5.935060
247 1715.673    1 281.9928 1.8878984 2.044175 -2.011488 1.8237331 6.046883
248 2407.440    1 277.2219 0.8503734 2.846275 -4.423145 0.7317445 6.631184
       median mean.mc_std_err sd.mc_std_err
174 2.0400210      0.06051837    0.04527209
175 1.8500675      0.06474664    0.04837077
176 0.8842236      0.08772952    0.06474474
246 2.0925785      0.05845757    0.04372027
247 1.8237331      0.06464247    0.04820188
248 0.7317445      0.09000710    0.06630482
```

And finally the daily maps for PM10 (median) concentrations!

```
> ggplot() +
+   gg(pred, aes(UTMX_km, UTMY_km, fill = mean))  +
+   facet_wrap(.~ time) +
+   scale_fill_viridis() + coord_equal() + gg(border)
```

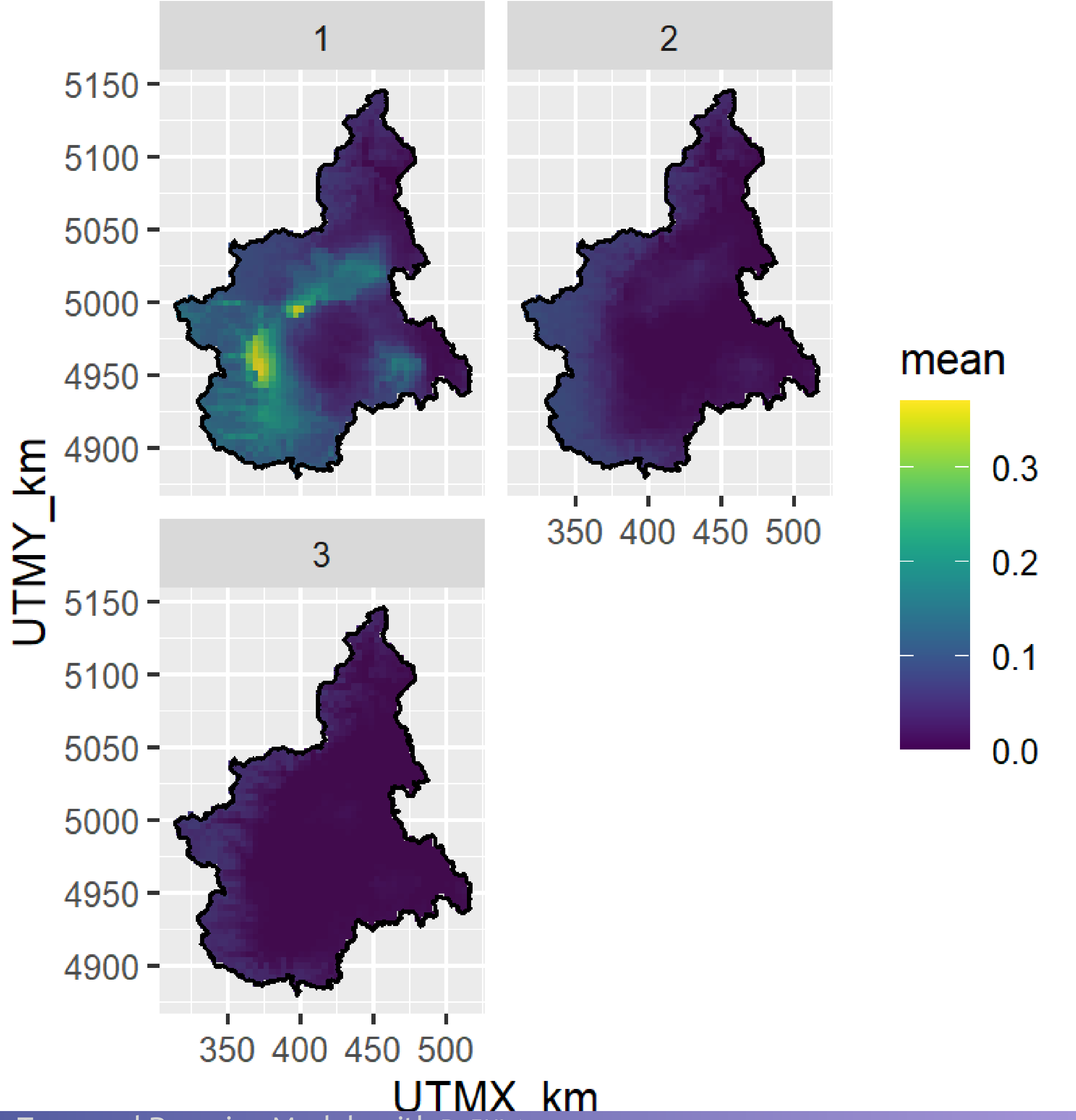


Maps for the exceedance probability

With `inlabru` `predict()` function, it is also very easy to compute the posterior probability of exceeding the 50 $\mu g/m^3$ threshold:

```
> predprob = predict(fit, covariate_grid,
+                    ~ (Intercept + SPDE + A + TEMP)
+                    seed = 2, n.samples = 500)
```

```
> ggplot() +
+   gg(predprob, aes(UTMX_km, UTMY_km, fill = mea
+   facet_wrap(.~ time, ncol = 2, nrow = 2) +
+   scale_fill_viridis() + coord_equal() + gg(bor
```



References

Cameletti, M., R. Ignaccolo, and S. Bande (2011). "Comparing spatio-temporal models for particulate matter in Piemonte". In: *Environmetrics* 22.8, pp. 985-996. DOI: https://doi.org/10.1002/env.1139.

Cameletti, M., F. Lindgren, D. Simpson, et al. (2013). "Spatio-temporal modeling of particulate matter concentration through the SPDE approach". In: *AStA Advances in Statistical Analysis* 97.2, pp. 109-131.

Fioravanti, G., S. Martino, M. Cameletti, et al. (2021). "Spatio-temporal modelling of PM10 daily concentrations in Italy using the SPDE approach". In: *Atmospheric Environment* 248, p. 118192. DOI: https://doi.org/10.1016/j.atmosenv.2021.118192.