Modelling Geostatistical Data

1 Geostatistical data

Geostatistical data are the most common form of spatial data found in environmental setting. In these data we regularly take measurements of a spatially continuous ecological or process at a set of fixed locations. This could be data from transects (e.g, where the height of trees is recorded), samples taken across a region (e.g., water depth in a lake) or from monitoring stations as part of a network (e.g., air pollution). In each of these cases, our goal is to estimate the value of our variable across the entire space.

Let D be our two-dimensional region of interest. In principle, there are infinite locations within D, each of which can be represented by mathematical coordinates (e.g., latitude and longitude). We then can identify any individual location as $s_i = (x_i, y_i)$, where x_i and y_i are their coordinates.

We can treat our variable of interest as a random variable, Z which can be observed at any location as $Z(\mathbf{s}_i)$.

Our geostatistical process can therefore be written as:

$$\{Z(\mathbf{s}); \mathbf{s} \in D\}$$

In practice, our data are observed at a finite number of locations, m, and can be denoted as:

$$z = \{z(\mathbf{s}_1), \dots z(\mathbf{s}_m)\}$$

We have observed our data at m locations, but often want to predict this process at a set of unknown locations. For example, what is the value of $z(\mathbf{s}_0)$, where $\mathbf{s}_0$ is an unobserved site?

There are two main steps in classical geostatistical analysis.

How do I produce a statistical model for the data?

How do I use my model to estimate quantities of interest?

The first part requires us to think about how our measured data points relate to each other - in other words, to understand spatial autocorrelation. The second part requires us to use that information to predict the value at unmeasured locations, and then to produce maps or summary statistics based on this.

2 Spatial autocorrelation

Spatial statistics and geostatistics quantify spatial variance and correlation based on distance, aligning with Tobler's law—nearby measurements tend to be more correlated, while this relationship weakens with distance. Spatial dependence can reveal ecological processes like species interactions and responses to environmental gradients. However, it also poses challenges for statistical analysis, as it violates the common assumption of independent data in traditional methods.

Analyzing spatial dependence can offer insights into the biological processes shaping observed patterns, such as social behavior, resource distribution, or dispersal. While measuring spatial dependence alone may not provide definitive answers, it helps generate hypotheses and refine predictions. Additionally, spatial dependence can influence assessments of conservation threats and strategies.

Spatial correlation is usually driven by some unmeasured confounding variable(s) - for example, air pollution is spatially correlated because nearby areas tend to experience similar traffic levels.

It is important that we account for these correlations in our analysis - failing to do so will lead to poor inference. For a set of geostatistical data $\mathbf{z} = \{z(\mathbf{s}_1), \dots, z(\mathbf{s}_m)\}$, we can consider the general model:

$$Z(\mathbf{s}_i) = \mu(\mathbf{s}_i) + e(\mathbf{s}_i)$$

Here $\mu(\mathbf{s}_i)$ is a mean function which models trend and covariate effects. Then $e(\mathbf{s}_i)$ is the error process which accounts for any spatial correlation which exists after accounting for $\mu(\mathbf{s}_i)$. Spatial statistics is therefore often focused on understanding the process for $e(\mathbf{s}_i)$.

We have observations at m locations

$$\mathbf{z} = \{z(\mathbf{s}_1), \dots, z(\mathbf{s}_m)\}.$$

We want to use these to obtain an estimate of $Z(\mathbf{s}_0)$ where $\mathbf{s}_0$ is an unobserved location. How do we model the spatial dependence between our observed sites $\mathbf{s}_1, \dots, \mathbf{s}_m$? What does this tell us about the dependence between our observed sites and our unobserved site $\mathbf{s}_0$?

2.1 Variograms

The first step is to assess whether there is any evidence of spatial dependency in our data.

The function describing the dependence between values of our process Z separated by different lags is known as the **autocovariance function**. This is similar to the autocorrelation function (ACF) used for temporal data.

In geostatistical models, we can summarise the covariance structure of a spatial Gaussian random field with its **variogram** $2\gamma(\cdot)$ (or semivariogram $\gamma(\cdot)$). The variogram measures the variance of the difference in the process at two spatial locations $\mathbf{s}$ and $\mathbf{s} + \mathbf{h}$ and is defined as (under weakly stationary):

$$\text{Var}[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})] = E[(Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h}))^2] = 2\gamma_z(\mathbf{h}).$$

$$\begin{aligned}
\text{Var}[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})] &= E[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})^2] - \{E[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})]\}^2 \\
&= E[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})^2] - \{E[Z(\mathbf{s})] - E[Z(\mathbf{s}+\mathbf{h})]\}^2 \\
&= E[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})^2] - \{\mu - \mu\}^2 = E[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})^2]
\end{aligned}$$

Here, $2\gamma_z(\mathbf{h})$ is the variogram, but in practice we use the **semi-variogram**, $\gamma_z(\mathbf{h})$. We use the semi-variogram because our points come in pairs, and the semi-variance is equivalent to the variance per point at a given lag.

- When the variance of the difference $Z(\mathbf{s}) - Z(\mathbf{t})$ is relatively small, then $Z(\mathbf{s})$ and $Z(\mathbf{t})$ are similar (spatially correlated).
- When the variance of the difference $Z(\mathbf{s}) - Z(\mathbf{t})$ is relatively large, then $Z(\mathbf{s})$ and $Z(\mathbf{t})$ are less similar (closer to independence).

The variogram is a function of the underlying geostatistical process Z. In practice, we only have access to m realisations of this process, and therefore we have to estimate the variogram. This is known as the *empirical variogram*.

We obtain this by computing the semi-variance for all possible pairs of observations: $\gamma(\mathbf{s}, \mathbf{t}) = 0.5(Z(\mathbf{s}) - Z(\mathbf{t}))^2$.

2.1.1 Example: Constructing a Variogram

The data `parana` from the `geoR` Package contains the average rainfall over different years for the period May to June at 123 monitoring stations in Paraná state, Brazil.

Our goal is to model the asses if there is spatial correlation in the data so that we can predict rainfall levels at unsampled locations. Our data consist of location coordinates and rainfall levels.

3 Exploratory plots

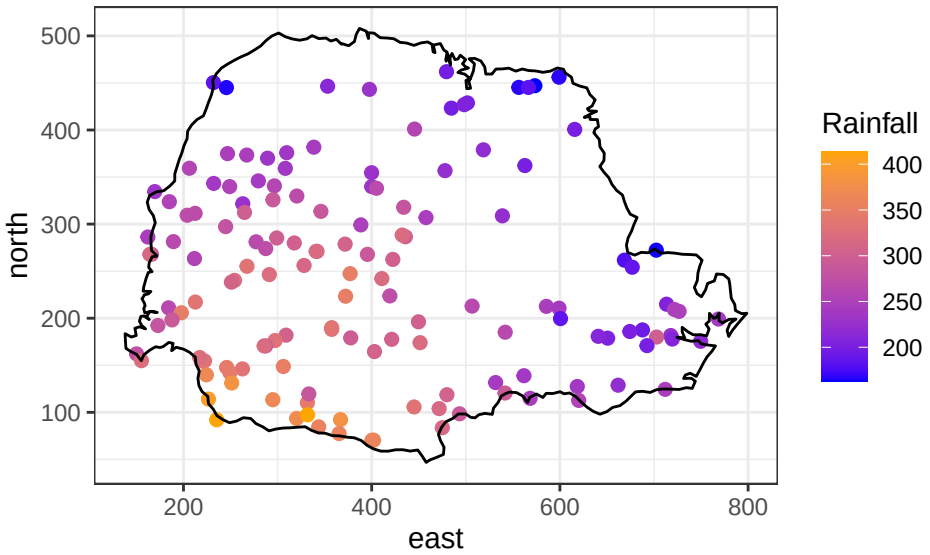


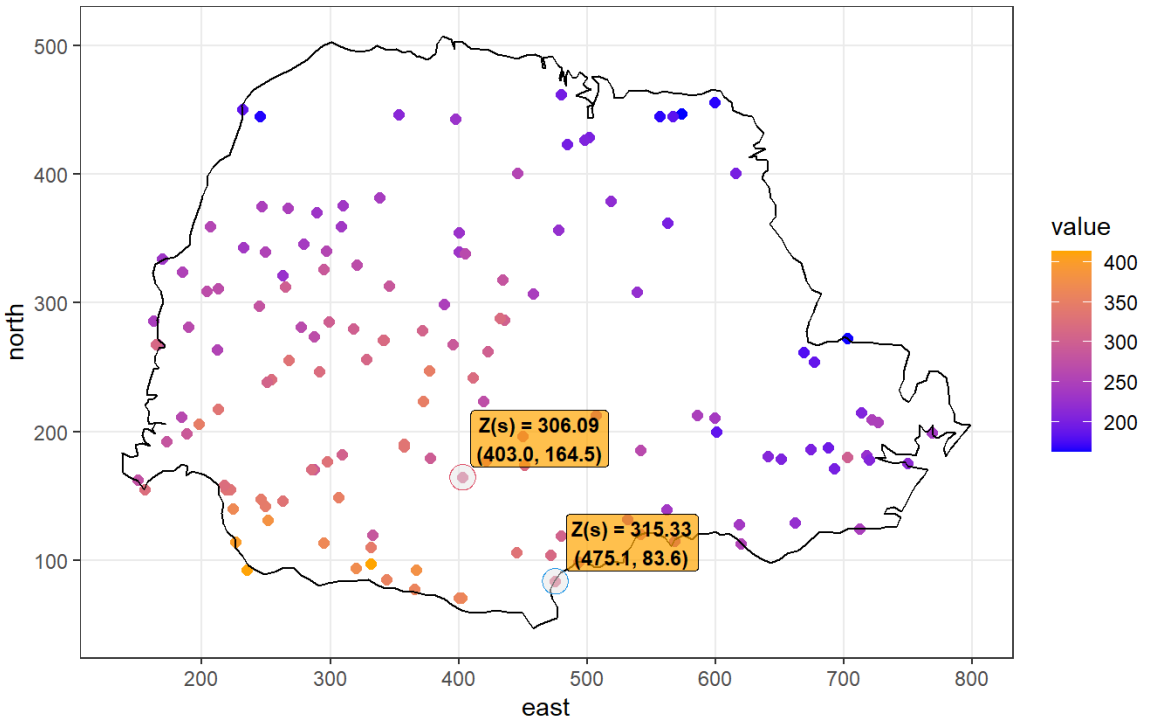
Figure 1: Rainfall values measured at 143 recording stations in Paraná state, Brazil withlow values being represented in blue and high values in red.

4 R Code

We can use `ggplot` to map the rainfall levels as follows:

```
library(geoR)
library(ggplot2)
library(sf)
library(patchwork)
library(gstat)

data.frame(cbind(parana$coords, Rainfall = parana$data)) %>%
ggplot()+
  geom_point(aes(east, north, color = Rainfall), size = 2) +
  coord_fixed(ratio = 1) +
  scale_color_gradient(low = "blue", high = "orange") +
  geom_path(data = data.frame(parana$border), aes(east, north)) +
  theme_bw()
```

To illustrate how an empirical variogram is computed, consider the two highlighted locations below.



1. We can first compute the distance between the two locations using the Euclidean distance formula

$$h = \sqrt{(475.1 - 403)^2 + (83.6 - 164.5)^2} = 108.36$$

2. Next, we compute the semi-variance between the points using their observed values

as

$$\begin{aligned}\gamma(\mathbf{s}, \mathbf{s}+\mathbf{h}) &= 0.5(Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h}))^2 \\ &= 0.5(315.33 - 306.9)^2 = 35.53\end{aligned}$$

3. We repeat this process for every possible pair of points, and plot h against $\gamma(\mathbf{s}, \mathbf{t})$ for each.

We can calculate the empirical variogram for the data using the `variogram` function from the `gstat` library. (we first need to convert our data to a `sf` spatial object). This plot shows the semi-variances for each pair of points.

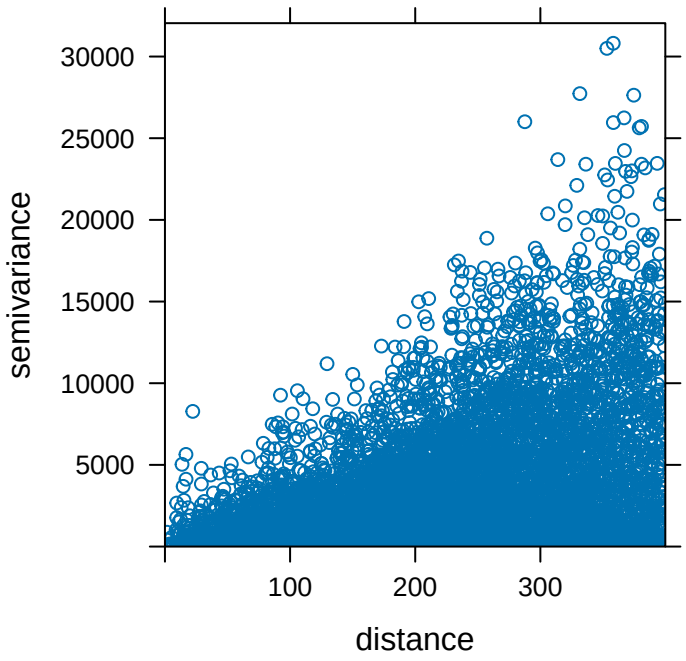
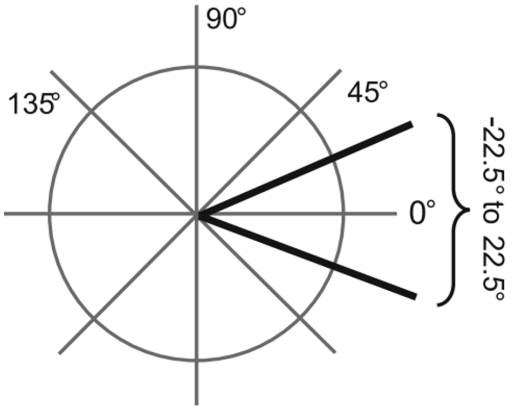


Figure 2: Empirical variogram values corresponding to the rainfall data in Paraná state, Brazil.

Notice that when we truncate the range of lag distances at which we consider spatial dependence to occur (see `max.dist` or `cutoff` for `geoR` and `gstat` respectivey in the R code above). Typically you would like set this value to approximately $1/2$ to $2/3$ of the total distance observed to ensure reliable estimates of spatial dependence. Beyond this, the number of point pairs available to estimate spatial dependence decreases (many points lack enough distant neighbors) leading to **biased estimates** of spatial dependence.

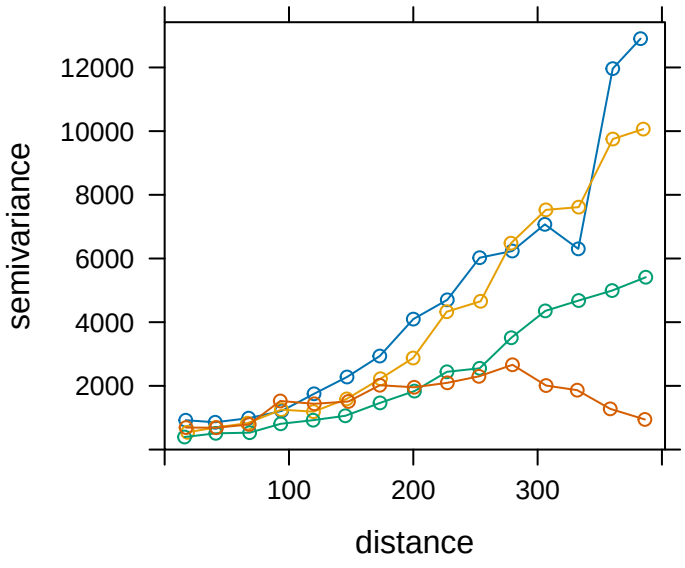
i Directional variograms

The variograms assumed *isotropy* - no directionality in spatial dependence. We can subset our data based on direction (e.g., calculating four variograms for the 0°, 45°, 90°, 135° directions where 0° cover the range from -22.5 to 22.5°) to visually consider whether there might be evidence for anisotropy in spatial dependence.



A strong difference in the empirical directional variograms indicate that anisotropy might be occurring in the data. Typically only 4 directions are considered (with windows $22.5°$) since any larger values (e.g., between 180 and 360) will provide the same patterns because the semivariance formula is symmetric.

```
emp4.geoR <- variogram(value~1,
                       data = parana_sf,
                       cutoff=400,
                       alpha = c(0, 45, 90, 135)) # specify the degree to subset the data

plot(emp4.geoR,multipanel=FALSE)
```



The directional variograms suggests anisotropy in the process of interest.

To make the variogram easier to use and interpret, we divide the distances into a set of discrete bins, and compute the average semi-variance in each. We compute this binned empirical variogram as:

$$\gamma(\mathbf{h}) = \frac{1}{2N(h_k)} \sum_{(\mathbf{s},\mathbf{t})\in N(h_k)} [z(\mathbf{s}) - z(\mathbf{t})]^2$$

Ingstat, we simply set the option `cloud = FALSE` (which is the default setting):

```
vario_binned <- gstat::variogram(value ~ 1,
                                 data = parana_sf,
                                 cloud = FALSE,
                                 cutoff = 400)
plot(vario_binned)
```

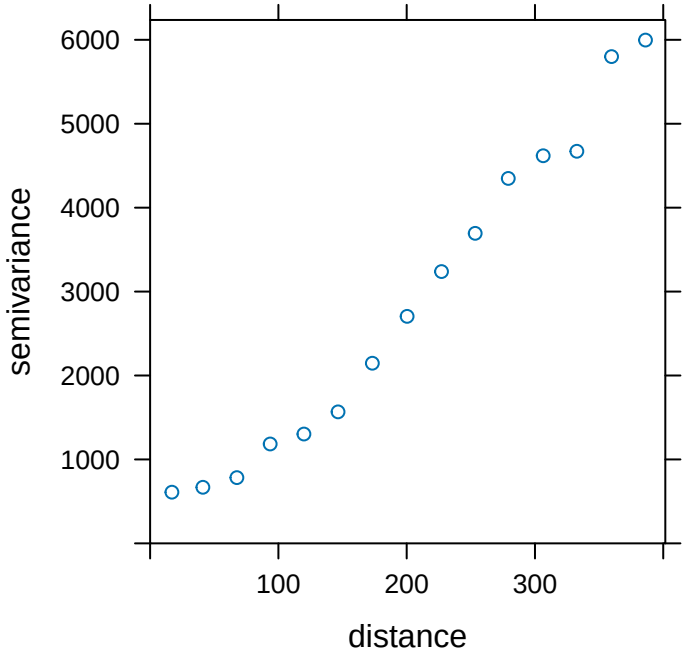


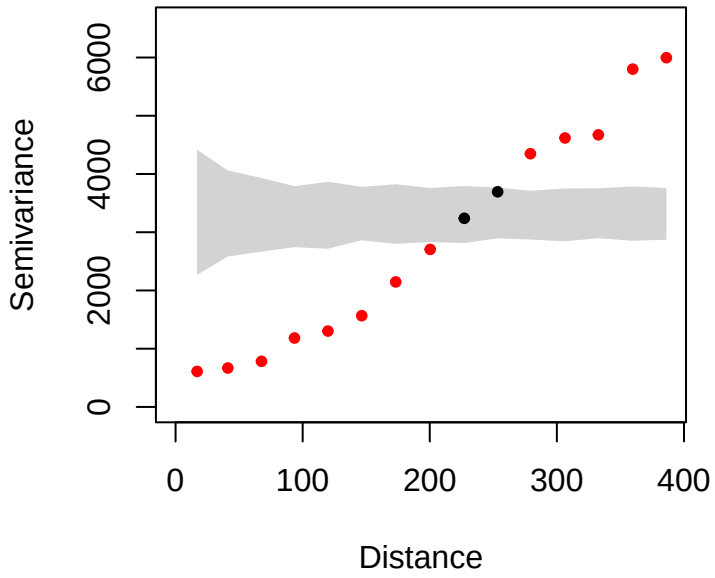
Figure 3: Averaged empirical variogram values corresponding to the rainfall data in Paraná state, Brazil.

Once we have computed an empirical variogram, we can constuct null envelope based on permutations of the data values across the locations, i.e. envelopes built under the assumption of no spatial correlation. By overlapping these envelopes with the empirical variograms we can determine whether there is some spatial dependence in our data ,e.g. if our observed variograms falls outside of the envelopes constructed under spatial randomness.

We can construct permutation envelopes on the `gstat` empirical variogram using the `envelope` function from the `variosig` R package. Then we can visualize the results using the `envplot` function:

```
library(variosig)

varioEnv <- envelope(vario_binned,
                     data = parana_sf,
                     locations = st_coordinates(parana_sf),
                     formula = value ~ 1,
                     nsim = 499)
```

```
envplot(varioEnv)
```




```
[1] "There are 13 out of 15 variogram estimates outside the 95% envelope."
```

In this example, we observe that the variogram falls outside of the null envelope at distances < 200m and also at distances above 300m.

5 Geostatistical Modelling

We treat the observed process of interest as being measured with error

$$(\text{observed value})_i = (\text{true value at location } i) + (\text{error})_i$$

alternatively

$$y_i = Z(\mathbf{s}_i) + \varepsilon_i$$

When geostatistical data are considered, we can often assume that there is a spatially continuous variable underlying the observations that can be modeled using a **random field**.

- we have a process that is occurring everywhere in space $\rightarrow$ natural to try to model it using some sort of function (of space)
- a **random field** is a random function that can generate smooth surfaces
- This is hard
- We typically make our lives easier by making everything Gaussian

5.1 Gaussian random field

A **Gaussian random field** (GRF) is a collection of random variables, where observations occur in a continuous domain, and where every finite collection of random variables has a multivariate normal distribution

$$\mathbf{z} = (z(\mathbf{s}_1), \dots, z(\mathbf{s}_m)) \sim N(\mu(\mathbf{s}_1), \dots, \mu(\mathbf{s}_m), \Sigma),$$

where $\Sigma_{ij} = \text{Cov}(z(\mathbf{s}_i), z(\mathbf{s}_j))$ is a dense $m \times m$ matrix and measures the strength of the linear dependence between $Z(\mathbf{s}_i)$ and $Z(\mathbf{s}_j)$ (As usual, we can compute the variance of $Z(\mathbf{s}_i)$ as a special case of the covariance where $\mathbf{s}_i = \mathbf{s}_j$).

We then need to use a covariance function $C_z(\cdot, \cdot)$ that depends on the distance ($\Sigma_{ij} = C_z(\mathbf{s}_i, \mathbf{s}_j)$) between two points and that

- has no negative variances
- is symmetric
- is decreasing, with maximum at distance = 0

> **i** Random fields as random functions
>
> One way to think about random fields is as a way of defining a probability distribution on functions. Given our spatial region of interest D, We want to include a spatial random effect in the model because we believe there is *some relationship* between location and our response variable, but we don't know exactly what that relationship is. Since there are an infinite number of locations in D, there are an infinite number of possible values that f could take. A random field can be thought of as defining a distribution on what functions f can be. Recall that a GRF is essentially nothing more than a rule to define the joint distribution of the values of f for any (finite) collection of locations.

5.1.1 Stationary and isotropy

Our geostatistical process can be described as **weakly stationary** if the following criteria are met:

1. $E[Z(\mathbf{s})] = \mu_z(\mathbf{s}) = \mu_z$ - a finite constant which does not depend on **s**.
2. $C_z(\mathbf{s}, \mathbf{s} + \mathbf{h}) = C_z(\mathbf{h})$ - a function that depends only on the lag **h** and not on the absolute location.

- Condition 1 states that our mean function must be constant in space, with no overall spatial trend. We typically assume that after accounting for the deterministic part of the model $\mathbb{E}[Z(\mathbf{s})] = 0$
- Condition 2 states that for any two locations, their covariance depends only on how far apart they are (their **spatial lag**, h), not their absolute position.

A geostatistical process is said to be **isotropic** if the covariance function is *directionally invariant*. This means that the covariance between two points a distance h apart is the same no matter which direction you travel in.

Mathematically, this can be denoted by

$$C_z(\mathbf{h}) = C_z(||\mathbf{h}||).$$

5.2 Defining the Gaussian Field in a model

The first step in defining a model for a random field in a hierarchical framework is to identify a probability distribution for the observations available at m sampled spatial locations and represented by the vector $\mathbf{y} = y_1, \dots, y_m$.

For example, if we assume our observations follow a Gaussian distribution then

$$Y_i \sim N(\mu_i, \tau_e^{-1}) \text{ or } Y_i = \mu_i + \varepsilon_i \ \varepsilon \sim N(0, \tau_e^{-1})$$
$$\eta_i = \mu_i = \beta_0 + \dots + z(\mathbf{s}_i)$$

- $\tau_e^{-1} = \sigma_e^2$ represents the variance of the zero-mean measurement error (equivalent to the nugget effect)
- The response mean μ_i which coincides with the linear predictor η_i is defined based on:
  - the intercept β_0 and any additional covariates
  - the realization of the latent (*unobservable*) GF $Z(\mathbf{s}) \sim \text{MVN}(0, \Sigma)$ which accounts for the spatial correlation through $\Sigma = C_z(\cdot, \cdot)$.

5.2.1 Finite basis representation

At first glance, implementing such a model in practice can seem challenging. We operate in a world with limited computational power and finite memory, so working with fully general function spaces is not feasible. To make the problem manageable, we restrict the function space for f. In other words, we define a set of possible functions using a finite collection of basic "building blocks."

In practice, for Gaussian random fields (GRFs), we instead choose a finite basis and write

$$f(\mathbf{s}) = \sum_{j=1}^{M} z_j \phi_j(\mathbf{s}),$$

where $\phi_1, \dots, \phi_M$ known (non-random) functions and coefficients $\mathbf{z}_1, \dots, \mathbf{z}_M$ that we wish to estimate. Under this setup, defining a GRF is equivalent to specifying a joint probability distribution for the coefficient vector $\mathbf{z} = [z_1, \dots, z_M]^\top$. In summary, practical modelling requires restricting functions to finite bases for computational tractability, which leads us to place distributions on coefficients and carefully manage covariance or precision structures. This naturally highlights the importance of selecting a sensible covariance function, as it determines the dependence structure, computational efficiency, and ultimately how well the model captures the underlying spatial process.

5.3 The Matérn Field

A commonly used covariance function is the Matérn covariance function. The covariance of two points which are a distance h apart is:

$$\Sigma = C_\nu(h) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}h}{\rho} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}h}{\rho} \right) \tag{1}$$

- $\Gamma(\cdot)$ is the gamma function
- $K_\nu(\cdot)$ is the modified Bessel function of the second kind.
- Parameters σ^2, ρ and ν are non-negative values of the covariance function.
  - σ^2 is the *spatially structured* variance component
  - ρ is the range of the spatial process
  - ν controls smoothness of the spatial process.

Big n problem!

The disadvantage of the modelling approach involving the spatial covariance function is known as "big *n* problem" and concerns the computational costs required for algebra operations with dense covariance matrices (such as Σ).

In particular dense matrix operations scale cubically with the matrix size, given by the number of locations where the process is observed. Various approximation techniques have therefore been developed to address this issue; for example, the `mgcv` framework often uses low-rank approximations for smoothing splines.

The key to the approach in INLA is to avoid using the covariance matrix as much as possible and to instead work with the precision matrix. INLA deals with the computational challenges of GRFs by considering fields with sparse precision matrices. Such fields are Gaussian *Markov* random fields!. This is the basis for the main approach we will discuss, where a computationally efficient alternative is provided by the SPDE approach for representing GRFs as GMRFs.

5.4 The SPDE approach

The SPDE approach to modelling GMRFs was introduced by Lindgren et. al. (2011), and defines a (Matérn) GRF as the solution of a stochastic partial differential equation (SPDE)

$$(\kappa^2 - \Delta)^{\alpha/2} Z(s) = W(s)$$

What is this?

- $W(s)$ is random noise
- $Z(s)$ is the smooth process we want
- $(\kappa^2 - \Delta)^{\alpha/2}$ is an operator that "*smooths*" the white noise.
- κ and α are parameters

"solving the SPDE" means Find a random function $Z(t)$ such that the equality $(\kappa^2 - \Delta)^{\alpha/2} Z(t) = W(t)$ holds in distribution. Lindgren et. al. (2011) represent the solution as

$$Z(t) = \sum_{i=1}^{T} \psi_i(t) w_i$$

Where

- $\psi_i(s)$ are (known) basis functions for nodes $i = 1, \dots, T$
  - $\psi_i(s_i) = 1$
  - $\psi_i(s_j) = 0 \ \forall \ i \neq j$

  - Linear between neighboring nodes

- w_i are (unknown) weights
  - the field value $Z(s)$ is a **linear interpolation** between the two neighboring weights

This solution, which completely defined by a Gaussian vector of weights with zero mean and a sparse precision matrix, is approximated using a finite combination of piece-wise linear basis functions.

In practice, we approximate the GRF using a triangulated mesh. Then, the SPDE approach represents the continuous spatial process as a continuously indexed Gaussian Markov Random Field (GMRF). In other words, we construct an appropriate lower-resolution approximation of the surface by sampling it in a set of well designed points and constructing a piece-wise linear interpolant.

Note

The solution, as presented in Lindgren et. al. (2011) is a Matérn field Equation 1 where $\nu = \alpha - d/2$. For example, when $\alpha = 2 \Rightarrow \nu = 1$ since $d = 2$ we have that:

$$
\begin{aligned}
Z(s) &= \sum_{i=1}^{K} \psi_i(s) w_i \\
\mathbf{w} &\sim N(\mathbf{0}, Q^{-1}) \leftarrow \text{GMRF} \\
Q^{-1} &= \tau^2(\kappa^4 \mathbf{C} + 2\kappa^2 \mathbf{G} + \mathbf{G}\mathbf{C}^{-1}\mathbf{G})
\end{aligned}
$$

- $\mathbf{C}$ is diagonal with entries $C_{ii} = \int \psi_i(s) \mathrm{d}s$ and measures how much of the domain each basis function covers.
- $G_{ij} = \int \nabla\psi_i(s) \nabla\psi_j(s) \mathrm{d}s$ reflects the connectivity of the mesh nodes.
- because each basis function overlaps only with nearby ones, the resulting precision matrix is sparse, meaning each coefficient depends directly only on its neighbors

In summary

- The continuous stationary and isotropic Matérn GRF is the solution of a SPDE and is represented as

$$
Z(s) = \sum_{i=1}^{K} \psi_i(s) w_i
$$

- The weights vector $\mathbf{w} = (w_1, \ldots, w_K)$ is Gaussian with a **sparse** precision matrix $\longrightarrow$ Computational convenience
- The field has two parameters
  - The range ρ
  - The marginal variance σ^2
- These parameters are linked to the parameters of the SPDE
- We need to assign prior to them

5.5 Penalized Complexity Priors

Penalized Complexity (PC) priors proposed by Simpson et al. (2017) allow us to control the amount of spatial smoothing and avoid overfitting.

- PC priors shrink the model towards a simpler baseline unless the data provide strong evidence for a more complex structure.
- To define the prior for the marginal precision σ^{-2} and the range parameter ρ, we use the probability statements:
  - Define the prior for the range $\text{Prob}(\rho < \rho_0) = p_\rho$
  - Define the prior for the range $\text{Prob}(\sigma > \sigma_0) = p_\sigma$

6 Example: Modelling Rainfall in Brazil

Lets revisit the Paraná data containing the average rainfall over different years for the period May to June at 123 monitoring stations in Paraná state, Brazil.

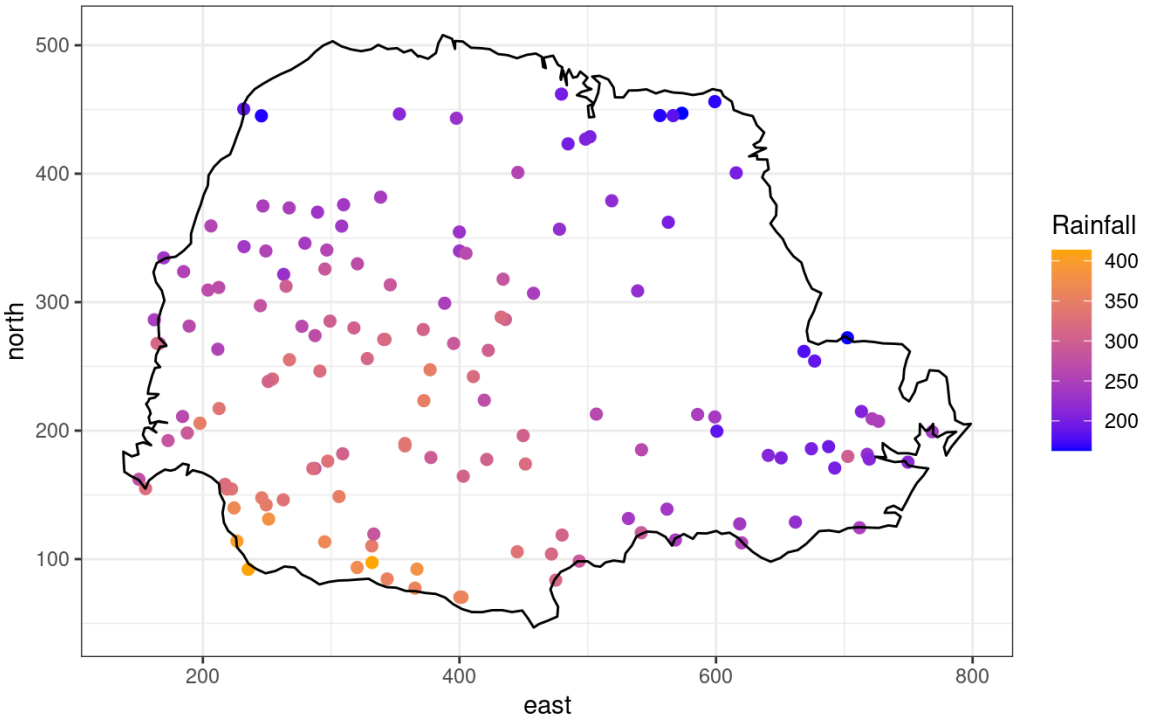


Figure 4: Rainfall values measured at 143 recording stations in Paraná state, Brazil with low values being represented in blue and high values in red.

```
# load some libraries
library(fmesher)
library(inlabru)
library(INLA)


parana_df <- data.frame(x = parana$coords[, 1],
```

```
                          y = parana$coords[, 2],
                          value = parana$data)

# convert this to an sf object

parana_sf <- st_as_sf(parana_df,
              coords = c("x", "y"))
```

6.1 The Model and SPDE representation

- **Stage 1** Model for the response

$$y(s)|\eta(s) \sim \text{Normal}(\mu(s), \sigma_e^2)$$

- **Stage 2** Latent field model

$$\eta(s) = \mu(s) = \beta_0 + Z(s)$$

  - A global intercept β_0
  - A Gaussian field $Z(s)$

- **Stage 3** Hyperparameters

  - Precision for the observational error $\tau_e = 1/\sigma_e^2$
  - Range and sd in the Gaussian field $\sigma_\omega, \tau_\omega$

6.2 The workflow

When fitting a geostatistical model we need to go through the following steps:

1. Build the mesh
2. Define the SPDE representation of the spatial GF. This includes defining the priors for the range and sd of the spatial GF
3. Define the components of the linear predictor. This includes the spatial GF and all potential covariates
4. Define the observation model

1. The mesh

First, we need to create the mesh used to approximate the random field. For this purpose we use the function `fm_mesh_2d` from the `fmesher` package. One way to build the mesh is to start from the locations where we have observations, these are contained in the dataset.

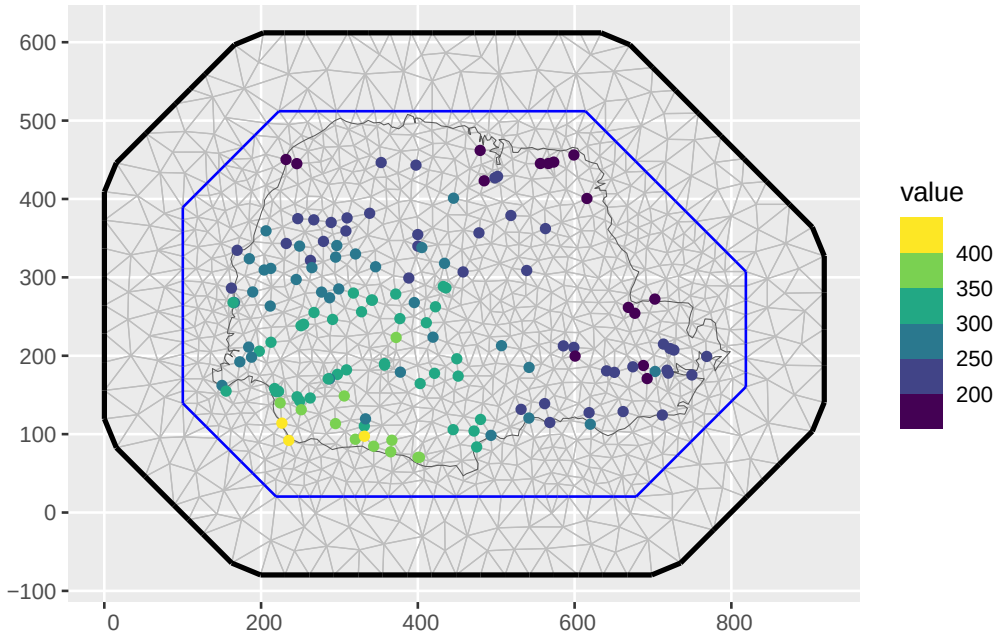
```
library(fmesher)
library(inlabru)
library(INLA)

mesh <- fm_mesh_2d(
  loc = parana_sf,
```

```
  offset = c(50, 100),
  cutoff = 1,
  max.edge = c(30, 60)
)
```

Here

- `max.edge` for maximum triangle edge lengths
- `offset` for inner and outer extensions (to prevent *edge effects*)
- `cutoff` to avoid overly small triangles in clustered areas

You can use the `plot(mesh)` function to visualize the mesh.



Note

General guidelines for creating the mesh

1. Create triangulation meshes with `fm_mesh_2d()`
2. Move undesired boundary effects away from the domain of interest by extending to a smooth external boundary
3. Use a coarser resolution in the extension to reduce computational cost (`max.edge=c(inner, outer)`)
4. Use a fine resolution (subject to available computational resources) for the domain of interest (inner correlation range) and filter out small input point clus-

ters (0 < `cutoff` < inner)
5. Coastlines and similar can be added to the domain specification in `fm_mesh_2d()` through the `boundary` argument.

Task

Look at the documentation for the `fm_mesh_2d` function typing

```
?fm_mesh_2d
```

play around with the different options and create different meshes.
The *rule of thumb* is that your mesh should be:

- fine enough to well represent the spatial variability of your process, but not too fine in order to avoid computation burden
- the triangles should be regular, avoid long and thin triangles.
- The mesh should contain a buffer around your area of interest (this is what is defined in the `offset` option) in order to avoid boundary artefact in the estimated variance.

2. Define the SPDE representation of the spatial GF

To define the SPDE representation of the spatial GF we use the function `inla.spde2.pcmatern`. This takes as input the mesh we have defined and the PC-priors definition for ρ and σ (the range and the marginal standard deviation of the field).

PC priors Gaussian Random field are defined in (Fuglstad et al. 2018). From a practical perspective for the range ρ you need to define two paramters ρ_0 and p_ρ such that you believe it is reasonable that

$$P(\rho < \rho_0) = p_\rho$$

while for the marginal variance σ you need to define two parameters σ_0 and p_σ such that you believe it is reasonable that

$$P(\sigma < \sigma_0) = p_\sigma$$

The Paraná state is around 663.8711 kilometers width by 464.7481 kilometers height. The PC-prior for the practical range is built considering the probability of the practical range being less than a chosen distance. We chose to set the prior considering the median as 100 kilometers.

```
spde_model =  inla.spde2.pcmatern(mesh,
                                  prior.sigma = c(1, 0.5),
                                  prior.range = c(100, 0.5))
```

3. Define the components of the linear predictor

We have now defined a mesh and a SPDE representation of the spatial GF. We now need to define the model components:

```
cmp = ~ Intercept(1) + space(geometry, model = spde_model)
```

NOTE since the data frame we use (`parana_sf`) is an `sf` object the input in the `space()` component is the geometry of the dataset.

4. Define the observation model

Our data are Gaussian distributed so we can define the observation model as:

```
# define model predictor formula
eta = value ~ Intercept  + space

# build the observation model
lik = bru_obs(formula = eta,
              data = parana_sf,
              family = "gaussian")
```

5. Run the model

Finally we are ready to run the model

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

6.3 Explore the results: Posterior summaries

The results of fitting the mode above are:

	Mean	2.5%	97.5%
Intercept	249.714	232.748	264.983
Precision for the Gaussian observations	4.482	3.197	5.511
Range for space	57.328	46.234	70.330
Stdev for space	46.654	41.222	52.736

Here,

- `Intercept` represents the average rainfall values
- `Precision for the Gaussian observations` are the observational errors
- `Range for space` is the correlation of the Matérn field
- `Stdev for space` is the marginal std deviation of the Matérn field

6.4 Spatial Predictions

In geostatistical applications, the main interest resides in the spatial prediction of the spatial latent field or of the response variable in new locations

- Suppose we observe a spatial process $Z(s) : s \in \mathcal{D}$ at locations $s_1, \dots, s_n$.
- Our goal: **predict the variable of interest at an unobserved location** $s_0 \in \mathcal{D}$.

given the data $y = (y_1, \dots, y_n)$, what can we say about $Z(s_0)$?

- Rather than a single guess, we want a **full uncertainty-aware prediction**.
- In a Bayesian setting, prediction is a **probabilistic task**.

The key lies in the **posterior predictive distribution**

$$\pi(\tilde{Y} \mid y) = \int \pi(\tilde{Y} \mid \Theta, y)\, \pi(\Theta \mid y)\, d\Theta,$$

where Θ denotes *all* latent components and hyperparameters.

- The **prediction likelihood** $\pi(\tilde{Y} \mid \Theta, y)$ depends on the task:
  - extrapolation(e.g. forecasting of an AR(1)): $\pi(Y_{n+1} \mid \Theta, y_n)$,
  - **interporlation**: $\pi(Y_i \mid y_{i-1}, y_{i+1}, \Theta)$,
- Spatial prediction fits naturally into this framework:
  - $\tilde{Y}$ may represent $Z(s_0)$, η_0, or the response at $y(s_0)$,
  - conditioning reflects the assumed spatial dependence.
- INLA approximates $\pi(\Theta \mid y)$ efficiently, enabling **full uncertainty propagation** when predicting over $s_0 \in \mathcal{D}$.

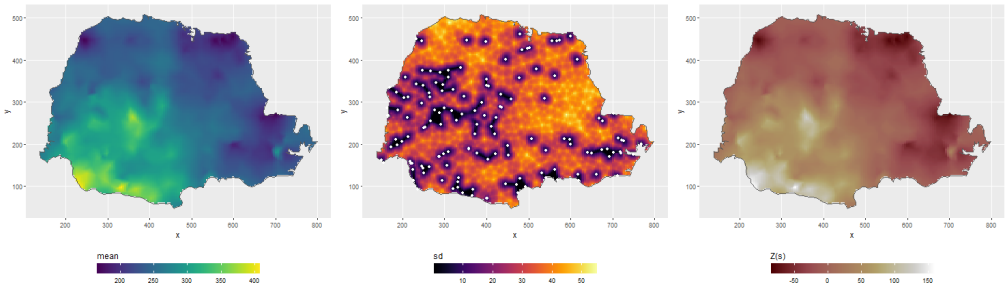
We now want to extract the estimated posterior mean and sd of spatial GF. To do this we first need to define a grid of points where we want to predict. We do this using the function `fm_pixel()` which creates a regular grid of points covering the mesh

```
# mask layer to the state border
border <- st_as_sf(data.frame(parana$borders),coords =c("east","north")) %>%
   summarise(geometry = st_combine(geometry)) %>%
  st_cast("POLYGON")
# resolution of our predictions
dims = c(150, 150)
# data frame for predictions
pred.df <- fm_pixels(mesh,dims = dims,mask =border,  format = "sf")
```

then compute the prediction for both the spatial GF and the linear predictor (spatial GF + intercept)

```
pred <- predict(fit,pred.df,
                formula ~ data.frame(
    spde =  space,
    eta = Intercept + space
  )
)
```

Finally, we can plot the maps



Now lets look at a more complex example (we will dicuss it in more detail during the lecture and the lab session)

7 Case Study: Modelling Pacific Cod Biomass Density

7.1 Exploring the Data

In the next example, we will explore data on the Pacific Cod (*Gadus macrocephalus*) from a trawl survey in Queen Charlotte Sound.

- The dataset the biomass density (kg/km^2) of Pacific cod in the area swept for a given survey in 2003 as well as depth covariate information.

Figure 6 A suggests some sort of quadratic effect of depth values, while Figure 6 B indicates a near Gaussian distribution for the log biomass density values and an important amount of zeros in our data.

If we consider only the locations where biomass > 0, the Envelope Variogram (Figure 7) of the log-density some unaccounted spatial variation at larger distances, which is not too bad. However, we then have a dilemma:

- If we omit the zeros, we'll get a good, accurate model fit for non-zero data, but we'll be throwing away all the data with zeros
- If we include the zeros, we won't be throwing any data away, but we'll get a strange-fitting model that both under- and over-predicts values.
- So what do we do?

```
[1] "There are 2 out of 15 variogram estimates outside the 95% envelope."
```

7.2 A first non-spatial model

A sensible non-spatial hierarchical Bayesian LGM for these that can be represented as follows:

First, we need to specify an observational model for our data:

- **Stage 1** Model for the response

$$y(s)|\eta(s) \sim \text{Tweedie}(p, \mu_i, \phi)$$

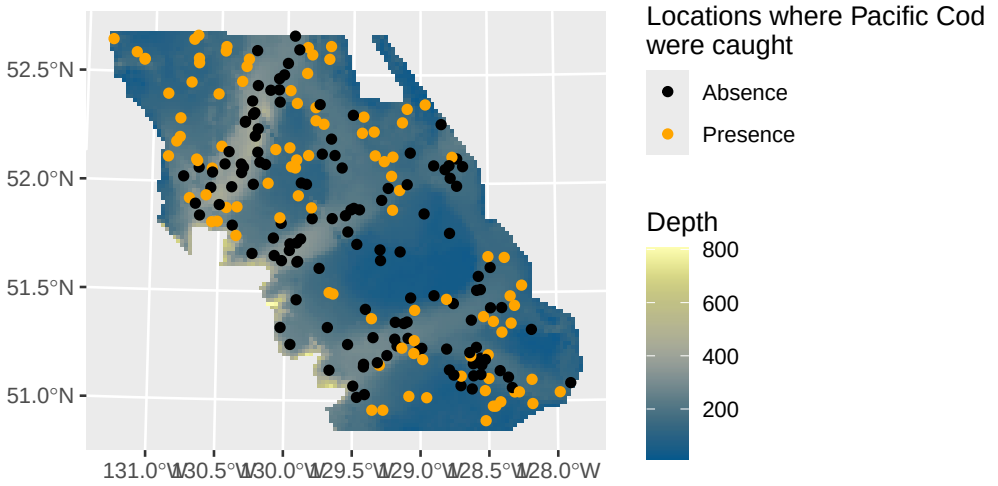


Figure 5: Map of the locations where Pacfic Cod were caught and the depth if the study area

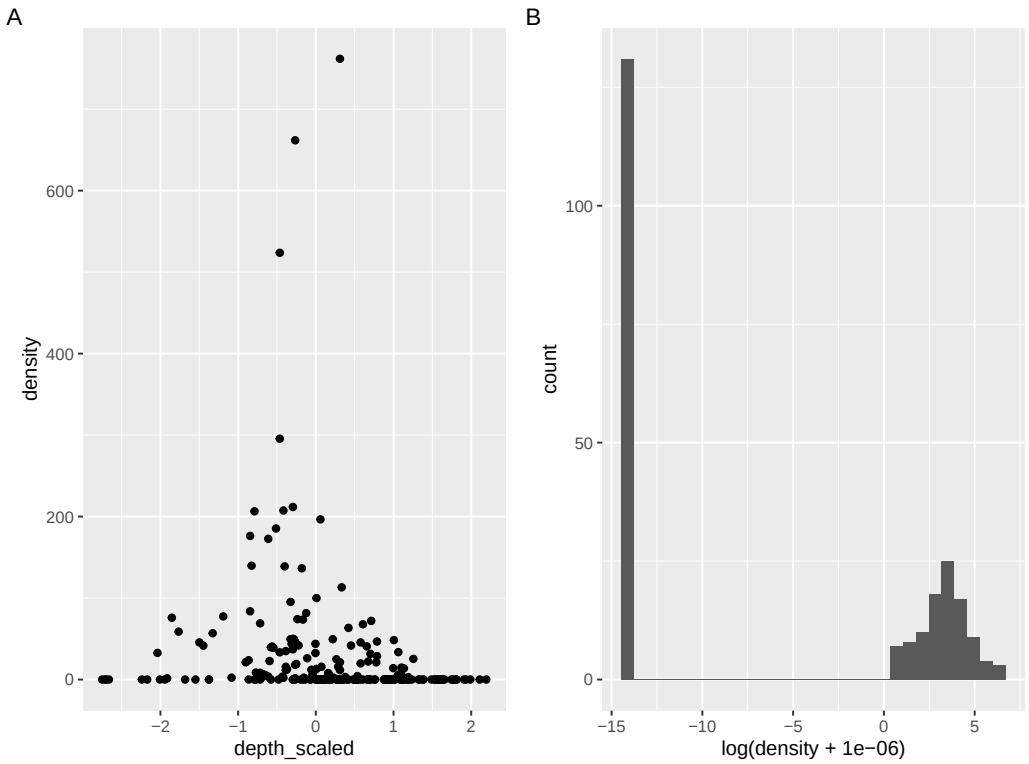


Figure 6: plot A shows the relationship bewteen the biomass density and the squared-depth values (previously scaled). Plot B shows the distirbution of log biomass density

- Here, the Tweedie distribution account for positive continuous density values that also contain zeros
- p determines the shape of the variance function (shifts from a Poisson distribution at $p = 1$ to a gamma distribution at $p = 2$)
- $\mu(s) = \exp\ \eta(i)$ is the mean linked to linear predictor by the log link.
- ϕ = dispersion parameter .

Then, we need to specify the components of the latent field:

- **Stage 2** Latent field model

$$\eta(s) = \exp(\mu(s)) = \beta_0 + \beta_1 x(s) + \beta_2 x(s)^2$$

  - A global intercept β_0
  - A effect of covariate $x(s)$ (depth)
  - A quadratic effect of covariate $x(s)$ (depth)

And finally, state what the hyperparameters of our model are:

- **Stage 3** Hyperparameters
  - dispersion parameter ϕ
  - power parameter p

Notice that these hyperparameter are associated with our observational model only as we did include any random effect on the linear predictor. The results for fitting this model are summarised in Table 2. We can see that β coefficients suggest log-biomass peaks at

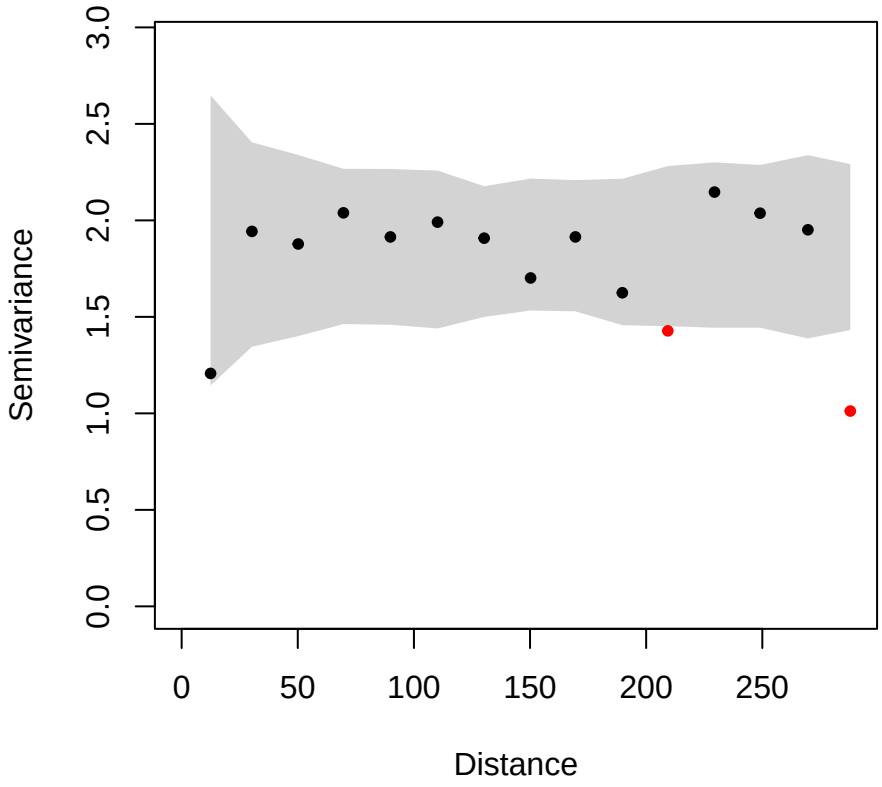


Figure 7: Monte Carlo Envelop for log biomass density values at locations where biomass in > 0.

Table 2: Non spatial Tweedie Model posterior summaries

INLA Model Results

Posterior summaries of fixed effects and hyperparameters

Parameter	Mean	2.5% Quantile	97.5% Quantile
Intercept	3.866	3.690	4.043
depth	-1.269	-1.480	-1.059
depth2	-1.077	-1.256	-0.898
p parameter for Tweedie	1.643	1.617	1.668
Dispersion parameter for Tweedie	3.804	3.530	4.093

an intermediate depth within the study range and decreases toward both shallower and deeper extremes. Then, $\phi > 1$ indicates overdispersion relative to the variance function. Potentially caused by unobserved heterogeneity. This suggest a spatially explicit model might be more appropriate here. However, Tweedie models fitted to biomass usually have convergence issues when there are large spatial areas with many zeros.

7.3 A multilikelihood Hurdle Geostatistical Model

Here we present a multilikelihood approach to jointly model the the log-biomass density while accounting for the presence of zeros in our data. A two-part model can be constructed to accommodate zero-inflated continuous data by combining separate likelihoods: one for the occurrence (e.g., Bernoulli) and one for the conditional positive amount (e.g., log-normal). The primary advantage of this framework is the ability to model the probability of an event and its magnitude independently.

- **Stage 1** Model for the response(s)

$$y_i|\eta_i^{(1)} \sim \text{Binomial}(1, \pi_i)$$
$$\log(z_i)|\eta_i^{(2)} \sim \text{Normal}(\mu_i, \tau_e^{-1})$$

  - We then define a likelihood for each outcome.
    - $y_i = \begin{cases} 1 & \text{if fishes have been caught at location } \mathbf{s}_i \\ 0 & \text{otherwise} \end{cases}$
    - $z_i = \begin{cases} NA & \text{if no fish were caught at location } \mathbf{s}_i \\ \text{biomass density at location } \mathbf{s}_i & \text{otherwise} \end{cases}$

This structure is equivalent to a **Hurdle-log-Normal model**, where the overall expected value of log biomass is given by the product $\pi_i * \mu_i$, with μ_i representing the conditional expectation from the log-normal component.

Next we define the components of our linear predictor. Notice how we are defining this model in a LGM framework:

- **Stage 2** Latent field model

$$\eta_i^{(1)} = \text{logit}(\pi_i) = X'\beta + \xi_i$$
$$\eta_i^{(2)} = \mu_i = X'\alpha + \omega_i$$

  - $\{\alpha, \beta\}$ = Intercepts + covariate effects.

Table 3: Posterior summaries of Hurdle model fitted to the biomass density data

Parameter	Mean	2.5% Quantile	97.5% Quantile
α_0	3.397	3.074	3.721
α_1	-0.321	-0.737	0.094
α_2	-0.238	-0.613	0.137
β_0	0.531	0.170	0.892
β_1	-1.163	-1.574	-0.751
β_2	-0.829	-1.133	-0.525
τ_ϵ^2	0.542	0.404	0.706
$\rho^{[1]}$	123.585	10.921	547.788
$\tau_{d,1}$	2.127	0.175	8.914
$\rho^{[2]}$	124.444	10.989	553.038
$\tau_{d,2}$	2.135	0.176	8.955

  - $\{\xi, \omega\}$ = are the Gaussian fields with Matérn covariance (separate for each outcome).

For the occurrence of fish, the linear predictor gets mapped to the logit of the probability of the Bernoulli model while the linear predictor for the biomass density is mapped to the mean of a log normal distribution.

- **Stage 3** Hyperparameters

The hyperparameter for the model are:

- observational error (nugget) τ_e
- Matérn field(s) parameters $\{\rho^{(1)}, \rho^{(2)}, \tau_d^{(1)}, \tau_d^{(2)}\}$

7.4 Posterior summaries

Table 3 shows the posterior summaries of the Hurdle model fitted to the biomass density data. Here,

- α_0 is the baseline catching probability on the logit scale
- β_0 is the predicted log(biomass density at the average depth (since these have been scaled)
- Coefficients α_1, α_2 refer to the change in the log-odds of catching fish as we increase 1 depth unit and unit^2 respectively.
- Coefficients β_1, β_2 indicate that the log-biomass decreases with depth.
- $\rho^{[1]}, \rho^{[2]}$, suggest spatial correlation decays at 123.58 and 124.44 Km respectively (the extension of the study is approx 46,000 km^2)
- unstructured variability is given by τ_e^{-1} while ,$\{\tau_{\delta,1}^{-1}, \tau_{d,2}^{-1}\}$ represent the spatially structured variability.

7.5 Model comparison

Note that in the hurdle model there is no direct link between the parameters of the two observation parts. However, the two likelihoods could share some of the components; for example the Matérn field could be used for both predictors. Thus, we will fit a model that estimates this field jointly and compare it with our two previous models (we will cover how to fit this model on the lab)

Model	DIC	WAIC	MLIK
Tweedie	1,599.690	1,612.043	-979.532
Hurdle	1,286.774	1,286.528	-682.897
Hurdle 2	1,286.595	1,286.353	-681.726

7.6 Spatial prediction

Lastly we can compute the spatial prediction for the biomass density. To do so we need to compute:

- $\pi(s)$ = Catching probability
- $\mathbb{E}[Z(s)|Y(s)] = \exp\left(\mu(s) + \frac{1}{2\tau_e}\right)$
- $\mathbb{E}(Z(s)) = \pi(s) \times \mathbb{E}[Z(s)|Y(s)]$