RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

INLA and inlabru with spatial patterns

Thierry Onkelinx

Overzicht

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

Checking spatial auto-correlation

www.INBO.be

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

Checking spatial auto-correlation

Pearson residuals

www.INBO.be

Definition

- components?

Definition

- components?
- observed value (y_i), fitted value ($\hat{y}_i$), mean squared error (MSE)

Definition

- components?
- observed value (y_i), fitted value ($\hat{y}_i$), mean squared error (MSE)
- formula

Definition

- components?
- observed value (y_i), fitted value ($\hat{y}_i$), mean squared error (MSE)
- formula
- $$pr_i = \frac{y_i - \hat{y}_i}{\sqrt{MSE}}$$

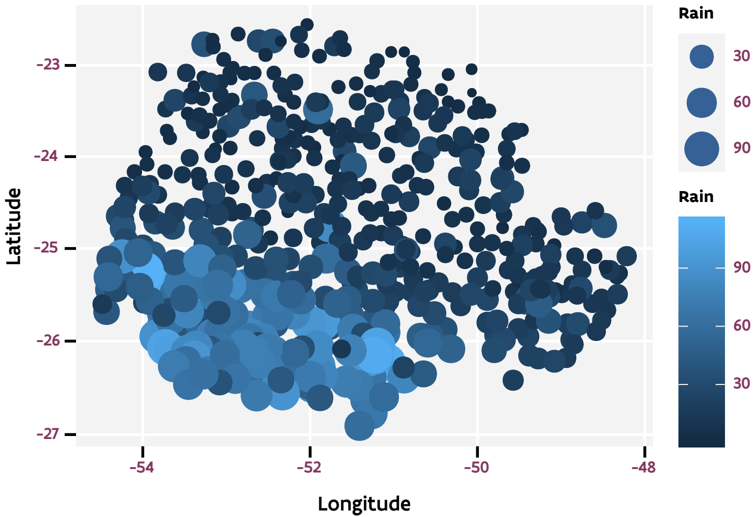
www.INBO.be

Flanders
State of the Art

Definition

- components?
- observed value (y_i), fitted value ($\hat{y}_i$), mean squared error (MSE)
- formula
- $$pr_i = \frac{y_i - \hat{y}_i}{\sqrt{MSE}}$$
- MSE (variance) depends on distribution! Check it using `inla.doc("name_of_your_distribution")`.

Example data: rainfall in Parana state, Brazil



Calculate Pearson residuals

```
model_iid <- inla(Rain ~ Xc + Yc, family = "gamma", data = dataset,
                  control.compute = list(waic = TRUE))
dataset %>%
  mutate(
    mu = model_iid$summary.fitted.values$mean,
    sigma2 = mu ^ 2 / model_iid$summary.hyperpar[1, "mean"],
    Pearson_iid = (Rain - mu) / sqrt(sigma2)
  ) -> dataset
```

Challenge 1

- What is the mean for your model?
- What is the variance for your model? Hint: `inla.doc("your distribution")`
- Calculate the Pearson residuals for your model

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

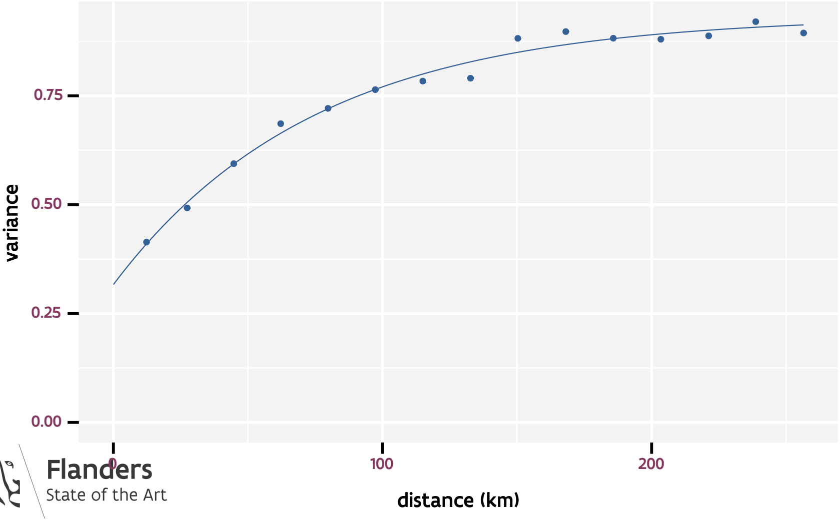
Checking spatial auto-correlation

Variogram

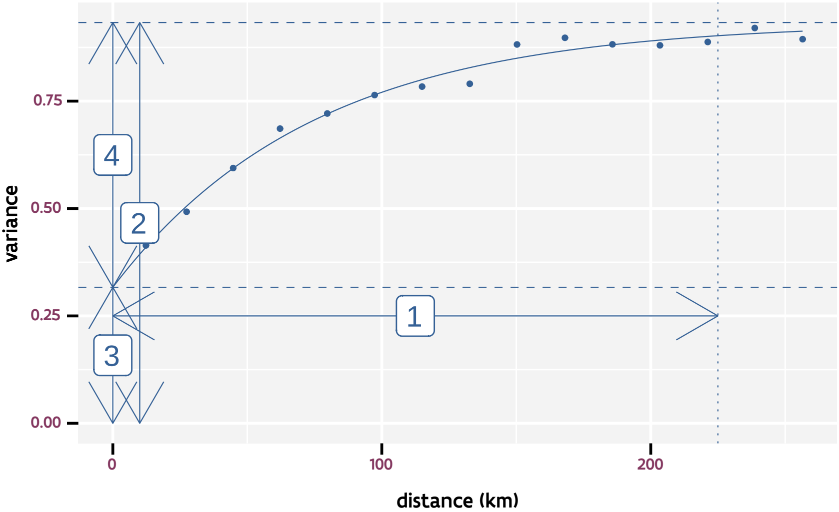
www.INBO.be

Definition

```
vg_default <- variogram(Pearson_iid ~ 1, locations = ~X + Y,
                        data = as.data.frame(dataset), cressie = TRUE)
```

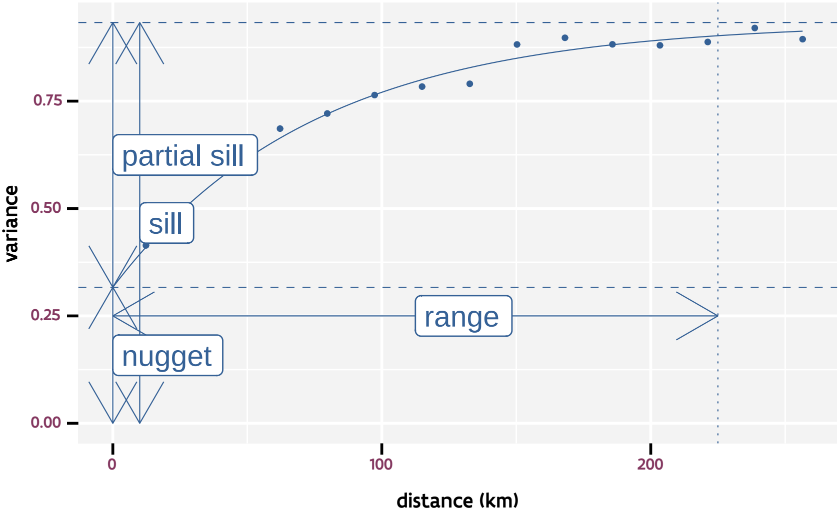


Important characteristics



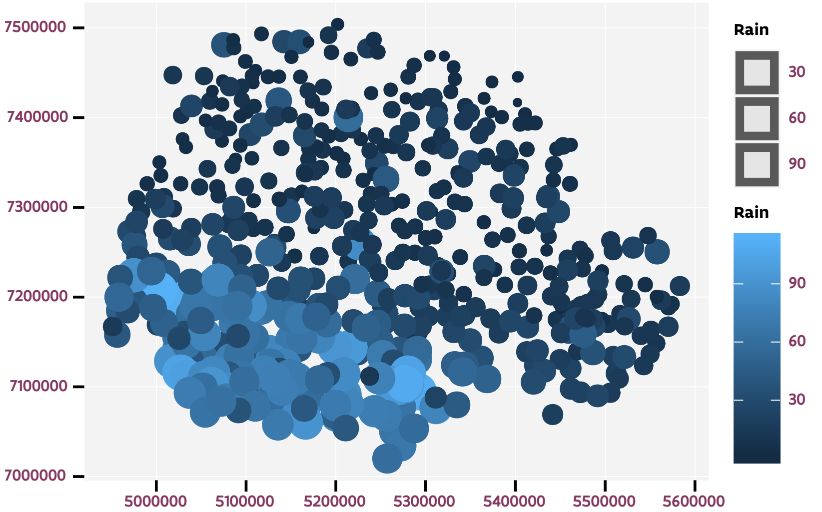
Flanders
State of the Art

Important characteristics



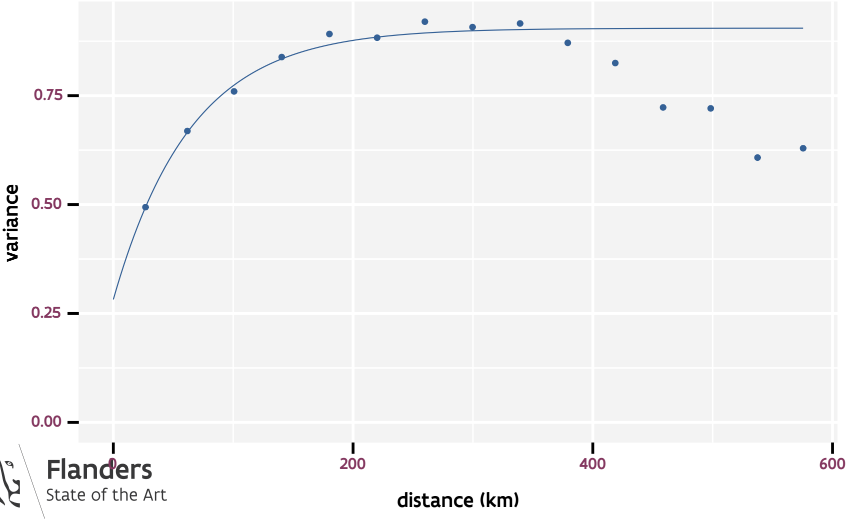
Flanders
State of the Art

Projected example data

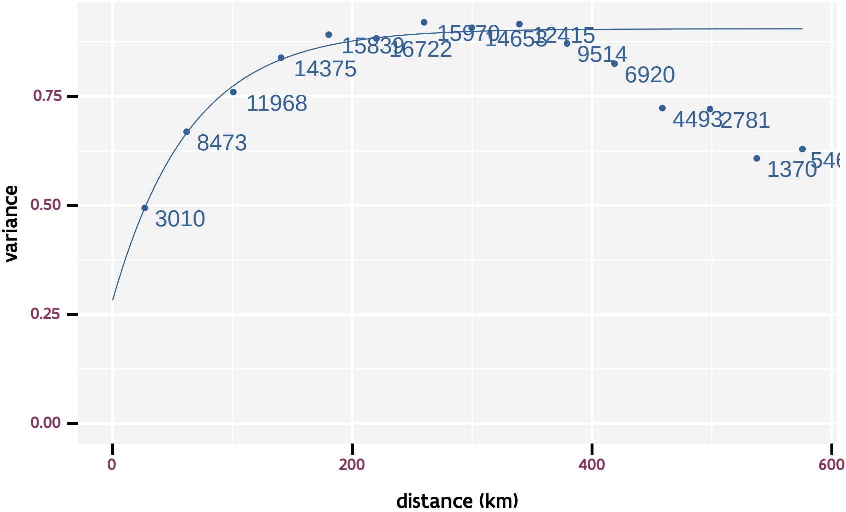


Increased cutoff

```
vg_large <- variogram(Pearson_iid ~ 1, locations = ~X + Y, cressie = TRUE,
                      data = as.data.frame(dataset), cutoff = 600e3)
```

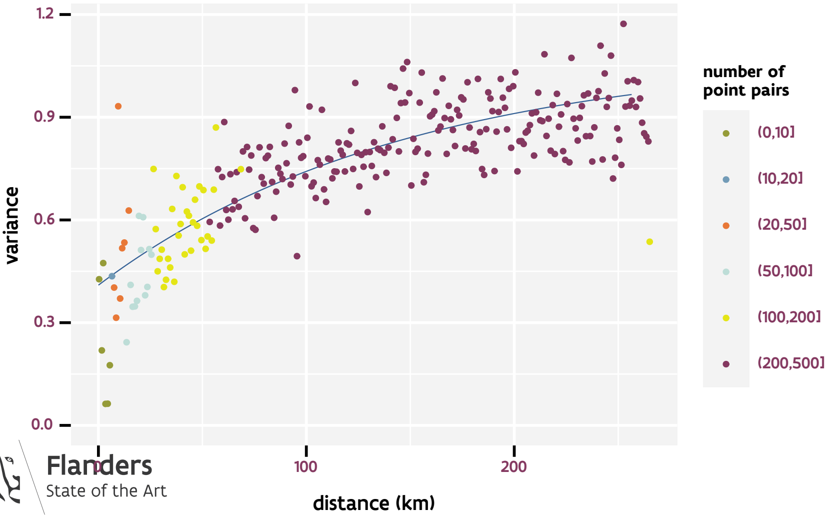


Number of point pairs is important



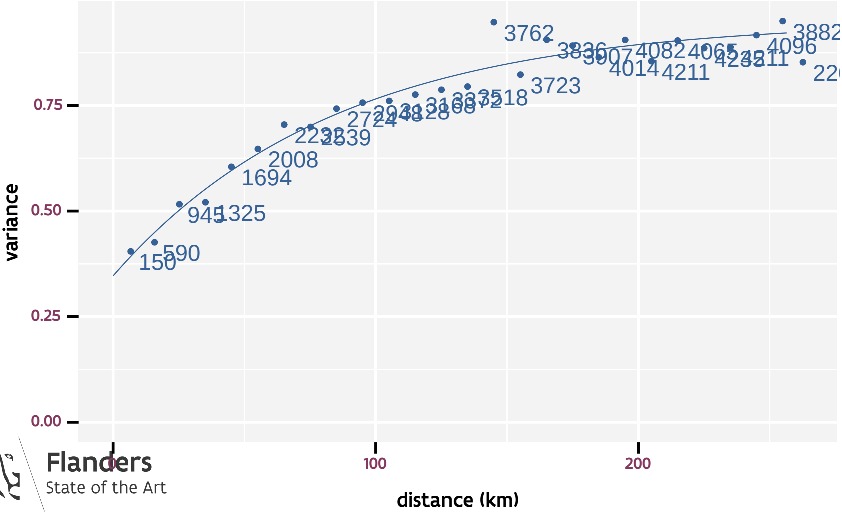
Too small width leads to unstable variograms

```
vg_small <- variogram(Pearson_iid ~ 1, locations = ~X + Y, cressie = TRUE,
                      data = as.data.frame(dataset), width = 1e3)
```



Sensible small width yields the most informative variogram

```r
vg_final <- variogram(Pearson_iid ~ 1, locations = ~X + Y, cressie = TRUE,
                      data = as.data.frame(dataset), width = 10e3)
```



Challenge 2

- What is the minimum binwidth for your data?
- Calculate the variogram for your model
- What is the approximate range of of the variogram?
- What is the nugget, sill and partial sill?

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

Prepare the model

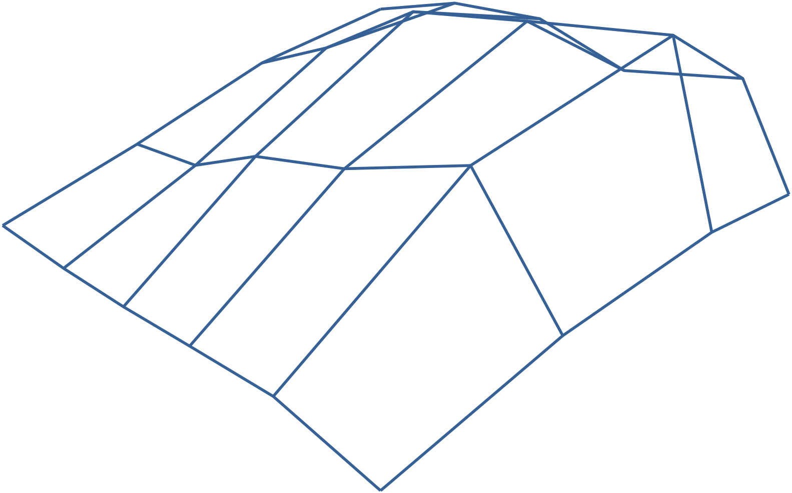
RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of the Art

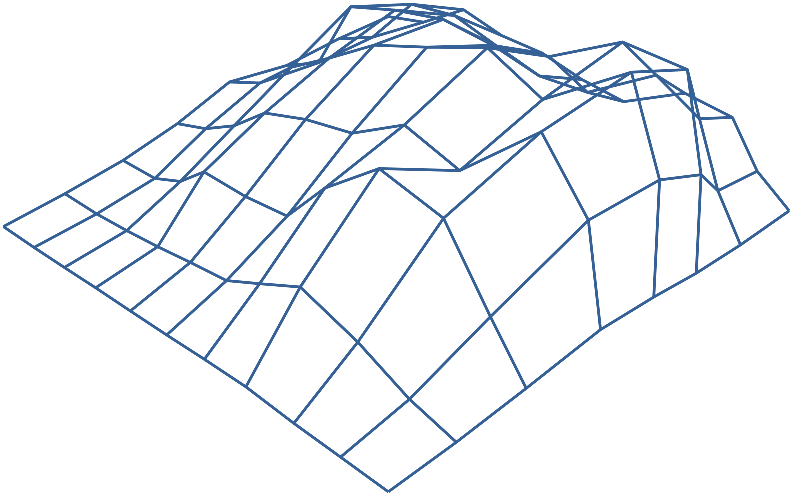
Prepare the model

Creating a mesh

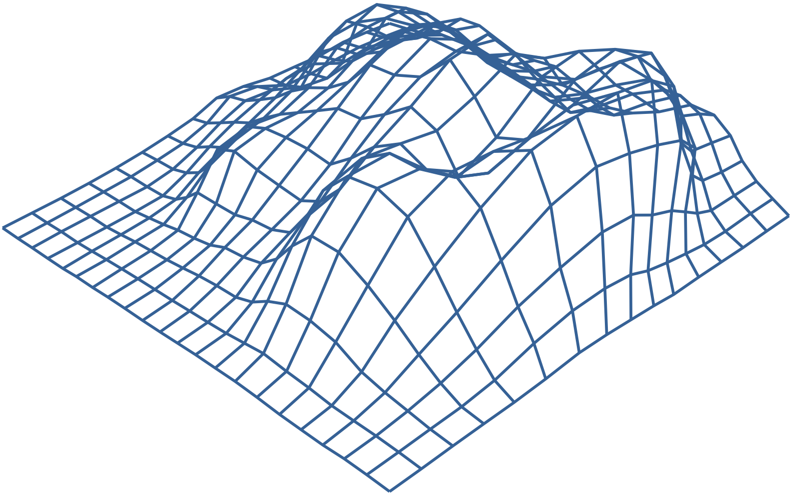
Size of a mesh I



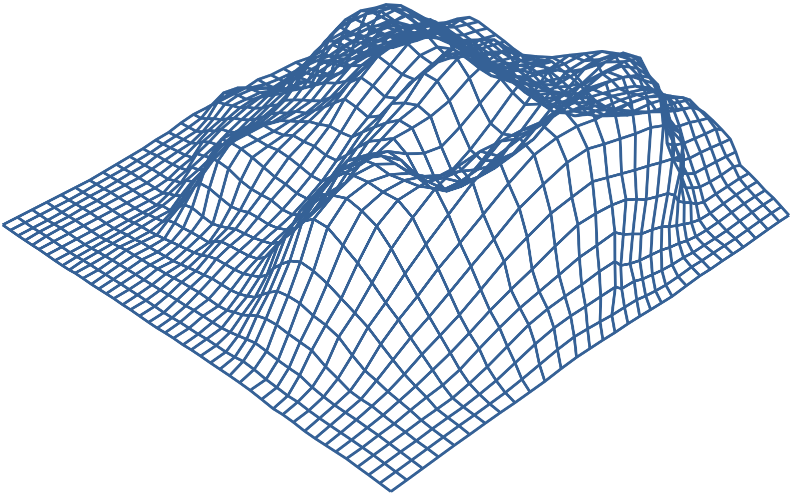
Size of a mesh II



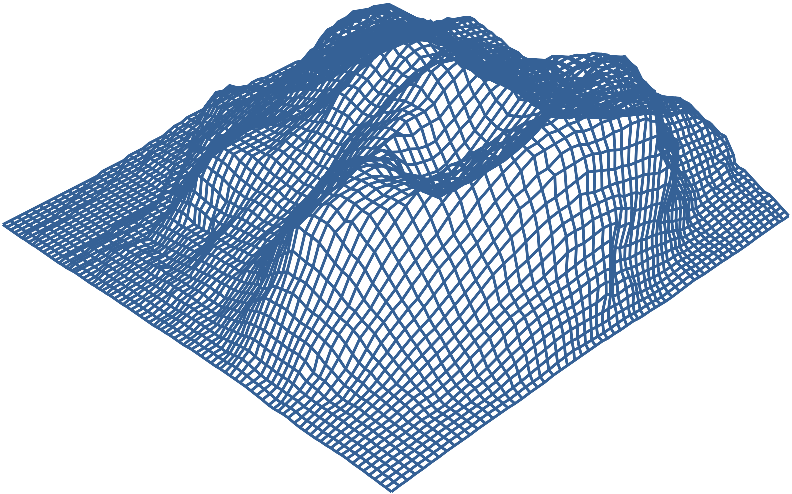
Size of a mesh III



Size of a mesh IV



Size of a mesh V

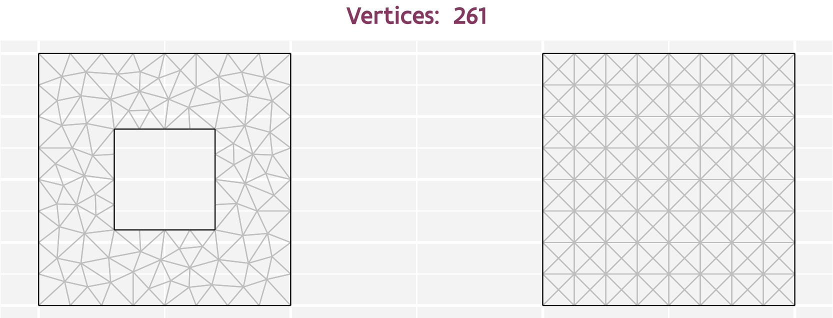


Guidelines

- equilateral triangles work best
- edge length should be around a third to a tenth of the range
- avoid narrow triangles
- avoid small edges
- add extra, larger triangles around the border
- simplify the border

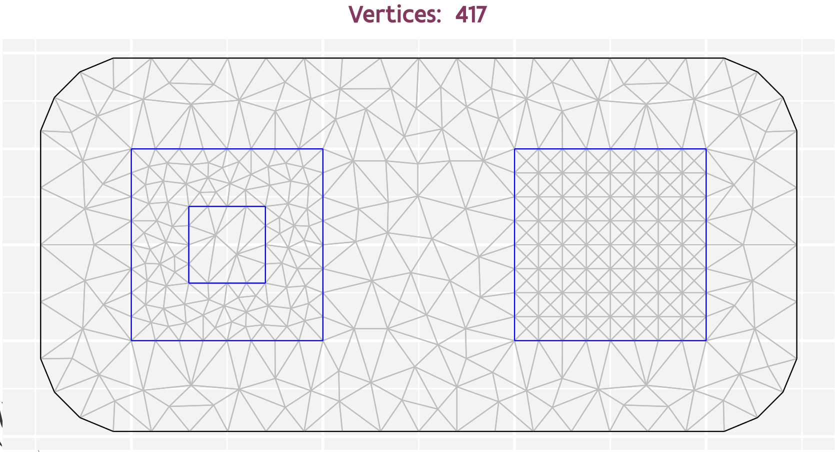
Mesh only within the border

```
mesh <- inla.mesh.2d(boundary = border, max.edge = 0.15)
ggplot() + gg(mesh) + coord_fixed() + theme_map() +
  ggtitle(paste("Vertices: ", mesh$n))
```



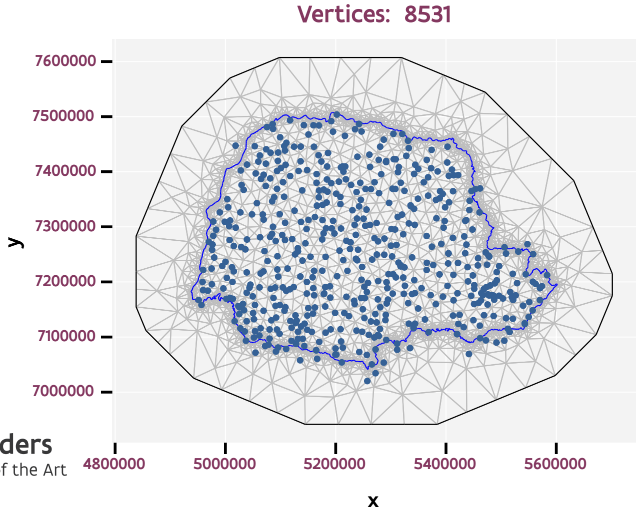
Mesh going outside the border

```
mesh <- inla.mesh.2d(boundary = border, max.edge = c(0.15, 0.3))
ggplot() + gg(mesh) + coord_fixed() + theme_map() +
  ggtitle(paste("Vertices: ", mesh$n))
```



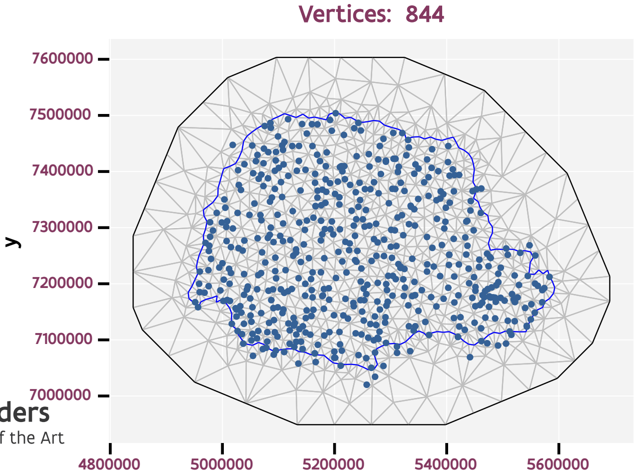
Mesh for rainfall data

```
mesh <- inla.mesh.2d(boundary = boundary, max.edge = c(30e3, 100e3))
ggplot(dataset) + gg(mesh) + geom_sf() + ggtitle(paste("Vertices: ", mesh$n)) +
  coord_sf(datum = st_crs(5880))
```



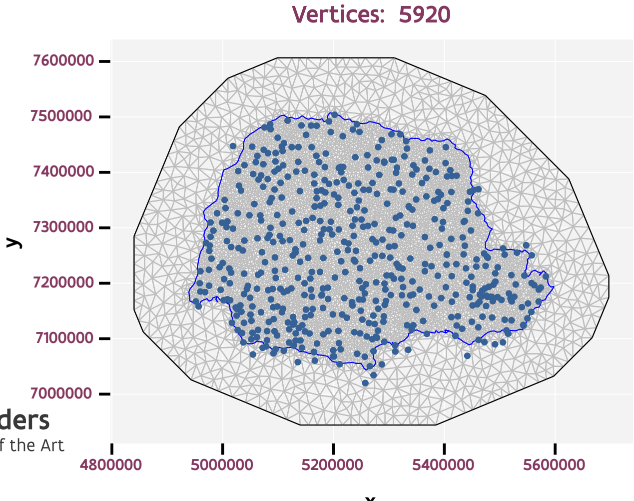
Use cutoff to simplify mesh

```
mesh1 <- inla.mesh.2d(boundary = boundary, max.edge = c(30e3, 100e3),
                      cutoff = 10e3)
ggplot(dataset) + gg(mesh1) + geom_sf() +
  ggtitle(paste("Vertices: ", mesh1$n)) + coord_sf(datum = st_crs(5880))
```



Finer mesh for final model run

```
mesh2 <- inla.mesh.2d(boundary = boundary, max.edge = c(10e3, 30e3),
                      cutoff = 5e3)
ggplot(dataset) + gg(mesh2) + geom_sf() +
  ggtitle(paste("Vertices: ", mesh2$n)) + coord_sf(datum = st_crs(5880))
```



Flanders
State of the Art

Challenge 3

- What are the relevant `max.edge` and `cutoff` for a course mesh?
- What are the relevant `max.edge` and `cutoff` for a smooth mesh?
- Create a course and a smooth mesh for your data

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

Prepare the model

Creating an SPDE model

SPDE using penalised complexity priors

Stochastic **P**artial **D**ifferential **E**quations

- `prior.range = c(r, alpha_r)`: $P(\rho < r) < \alpha_r$
- `prior.sigma = c(s, alpha_s)`: $P(\sigma > s) < \alpha_s$

```
spde1 <- inla.spde2.pcmatern(mesh1, prior.range = c(100e3, 0.5),
                             prior.sigma = c(0.9, 0.05))
spde2 <- inla.spde2.pcmatern(mesh2, prior.range = c(100e3, 0.5),
                             prior.sigma = c(0.9, 0.05))
```

Challenge 4

- What are relevant priors for the range and sigma for your data
  - Hint: see challenge 2
- Make the SPDE models for your data

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

Fitting the model

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

Fitting the model

Only the data

www.INBO.be

The stack for the observed data

```
A1 <- inla.spde.make.A(mesh = mesh1, loc = st_coordinates(dataset))
stack1 <- inla.stack(
  tag = "estimation", ## tag
  data = list(Rain = dataset$Rain), ## response
  A = list(A1, 1), ## projector matrices (SPDE and fixed effects)
  effects = list(
    list(site = seq_len(spde1$n.spde)), ## random field index
    dataset %>%
      as.data.frame() %>%
      transmute(Intercept = 1, Xc, Yc) ## fixed effect covariates
  )
)
```

Model fit

INLA

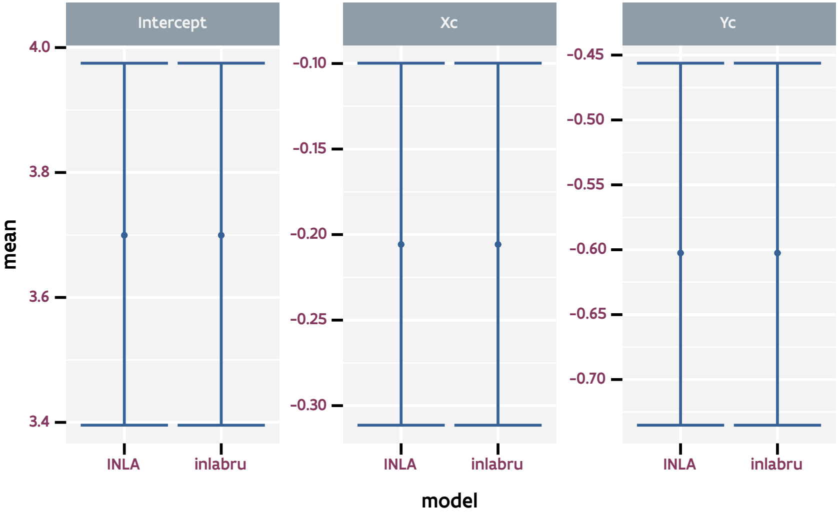
```
model_spde1 <- inla(Rain ~ 0 + Intercept + Xc + Yc + f(site, model = spde1),
                    family = "gamma", data = inla.stack.data(stack1),
                    control.predictor = list(A = inla.stack.A(stack1)),
                    control.compute = list(waic = TRUE)
                  )
```

inlabru

```
bru_spde1 <- bru(Rain ~ Xc + Yc + site(map = st_coordinates, model = spde1),
                 family = "gamma", data = dataset)
```

```
bru_spde1 <- bru(Rain ~ Xc + Yc + site(map = coordinates, model = spde1),
                 family = "gamma", data = as_Spatial(dataset))
```

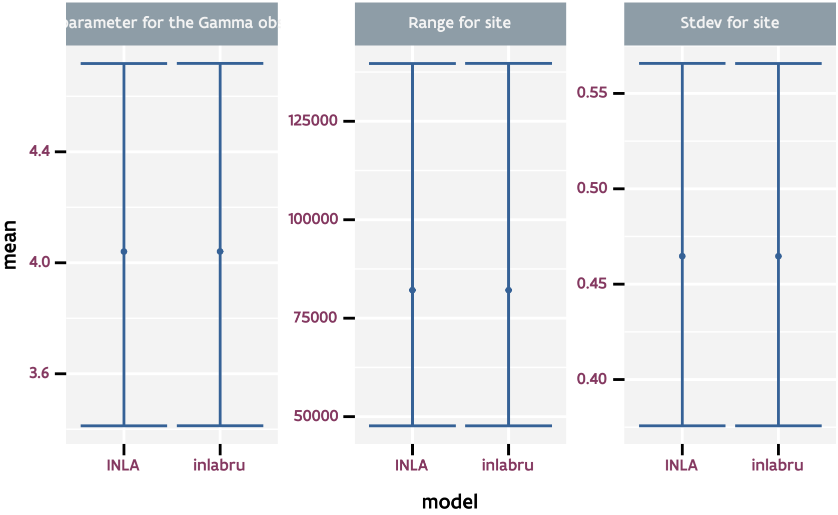
Comparison of fixed effect parameters



Flanders
State of the Art

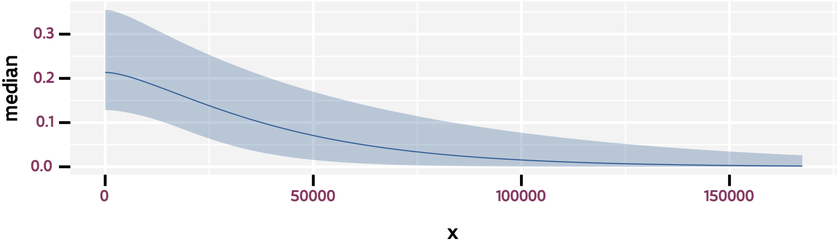
www.INBO.be

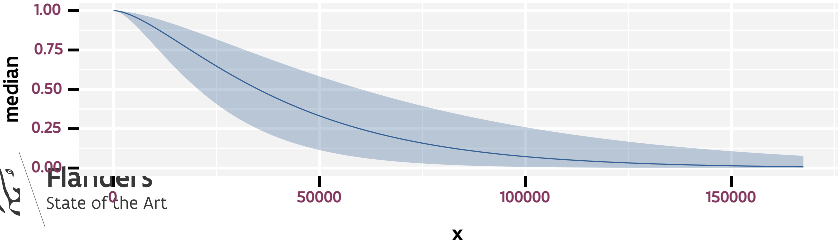
Comparing hyperparameters



Correlation structure

```
spde.posterior(bru_spde1, "site", what = "matern.covariance") -> covplot
spde.posterior(bru_spde1, "site", what = "matern.correlation") -> corplot
multiplot(plot(covplot), plot(corplot))
```





Calculate Pearson residuals

```
dataset %>%
  mutate(
    mu = model_spde1$summary.fitted.values$mean,
    sigma2 = mu ^ 2 / model_spde1$summary.hyperpar[1, "mean"],
    Pearson_iid = (Rain - mu) / sqrt(sigma2)
  ) -> dataset
```

```
## Error: Column `mu` must be length 528 (the number of rows) or one, not 1664
```

Using the stack index

```
si <- inla.stack.index(stack1, "estimation")$data
dataset %>%
  mutate(
    mu = model_spde1$summary.fitted.values$mean[si],
    sigma2 = mu ^ 2 / model_spde1$summary.hyperpar[1, "mean"],
    Pearson_spde = (Rain - mu) / sqrt(sigma2)
  ) -> dataset
```


Flanders
State of the Art

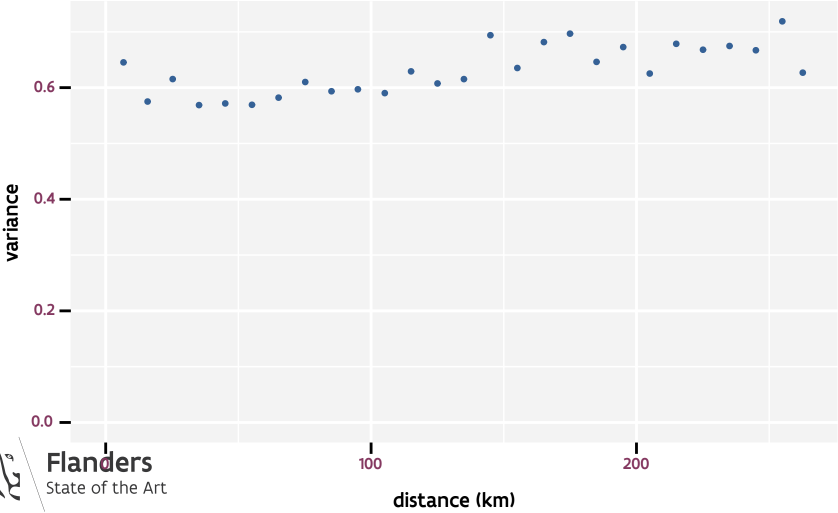
Using inlabru

```
fit <- predict(bru_spde1, as_Spatial(dataset), ~exp(Intercept + Xc + Yc + site))
dataset %>%
  mutate(
    mu = fit$mean,
    sigma2 = mu ^ 2 / model_spde1$summary.hyperpar[1, "mean"],
    Pearson_spde = (Rain - mu) / sqrt(sigma2)
  ) -> dataset
```


Flanders
State of the Art

Variogram

```
vg_fit <- variogram(Pearson_spde ~ 1, cressie = TRUE,
                    data = as_Spatial(dataset), width = 10e3)
```

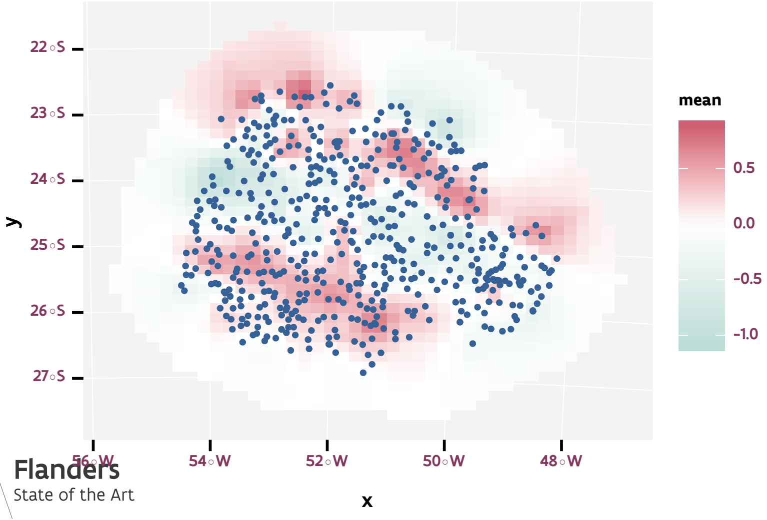


Interpolate GMRF

```r
A1.grid <- inla.mesh.projector(mesh1, dims = c(41, 41))
inla.mesh.project(A1.grid, model_spde1$summary.random$site) %>%
  as.matrix() %>%
  as.data.frame() %>%
  bind_cols(
    expand.grid(x = A1.grid$x, y = A1.grid$y)
  ) %>%
  filter(!is.na(ID)) -> eta_spde
```

Plot GMRF

```
ggplot(dataset) + geom_tile(data = eta_spde, aes(x = x, y = y, fill = mean)) +
  geom_sf() + scale_fill_gradient2()
```



Flanders
State of the Art

RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

Fitting the model

Predictions

www.INBO.be

Prediction stack for SPDE grid + fixed effects

```
expand.grid(X = A1.grid$x, Y = A1.grid$y) %>%
  mutate(Intercept = 1, Xc = X / 1e5 - 53, Yc = Y / 1e5 - 71) -> grid_data
stack1_grid <- inla.stack(
  tag = "grid", ## tag
  data = list(Rain = NA), ## response
  A = list(A1.grid$proj$A, 1), ## projector matrices (SPDE and fixed effects)
  effects = list(
    list(site = seq_len(spde1$n.spde)), ## random field index
    grid_data  ## covariates at grid locations
  )
)
```

Refit the model with the combinated stack

```
stack_all <- inla.stack(stack1, stack1_grid)
model_grid <- inla(Rain ~ 0 + Intercept + Xc + Yc + f(site, model = spde1),
                   family = "gamma", data = inla.stack.data(stack_all),
                   control.predictor = list(A = inla.stack.A(stack_all),
                                            link = 1),
                   control.compute = list(waic = TRUE),
                   control.mode = list(theta = model_spde1$mode$theta,
                                       restart = FALSE),
                   control.results = list(return.marginals.random = FALSE,
                                          return.marginals.predictor = FALSE)
                   )
```

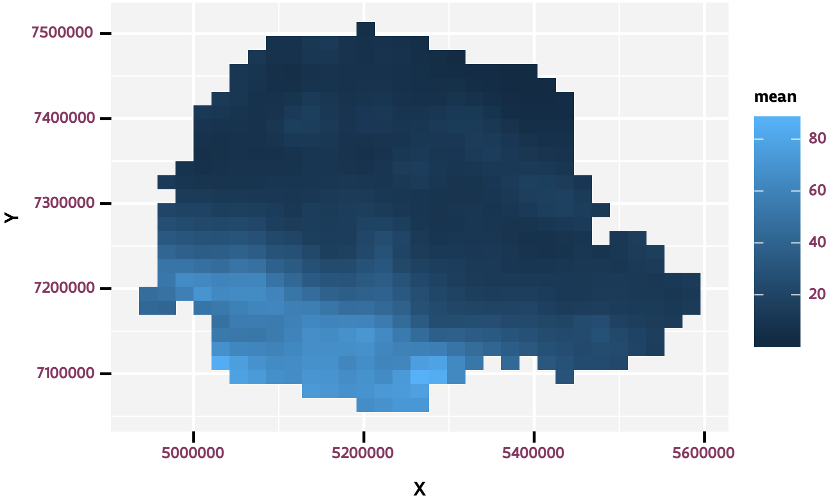
www.INBO.be

Flanders
State of the Art

Plot grid I

```r
si <- inla.stack.index(stack_all, "grid")$data
grid_data %>%
  bind_cols(model_grid$summary.fitted.values[si, ]) %>%
  `coordinates<-`(~X + Y) %>%
  `proj4string<-`(CRS("+init=epsg:5880")) -> gd
gd[!is.na(over(gd, boundary)), ] %>%
  as.data.frame() %>%
  ggplot() + geom_tile(aes(x = X, y = Y, fill = mean)) + coord_fixed()
```

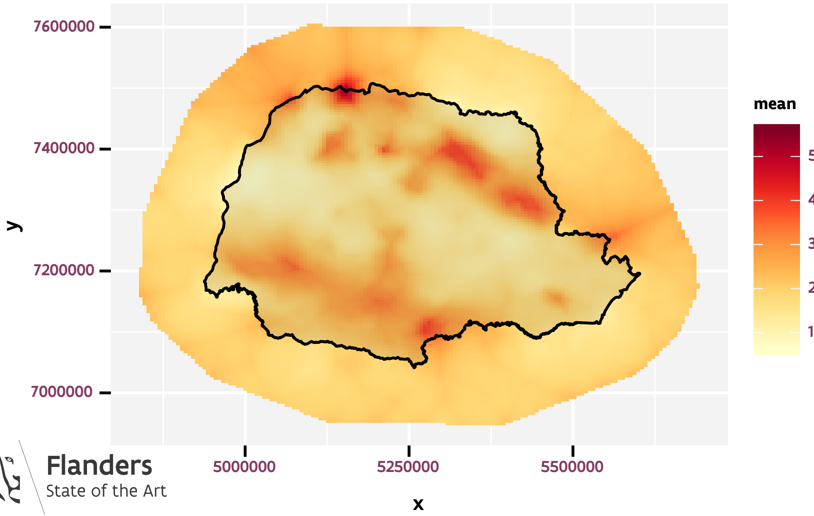
Plot grid II



Flanders
State of the Art

Using inlabru

```r
pred_mesh <- predict(bru_spde1, pixels(mesh1), ~exp(Intercept + Xc + Yc + site))
ggplot() + gg(pred_mesh) + gg(boundary)
```



Challenge 5

- Fit the model using the SPDE
- Plot a map of the GMRF
- Plot a map of the predictions and their credible interval