Spatial Modelling with INLA

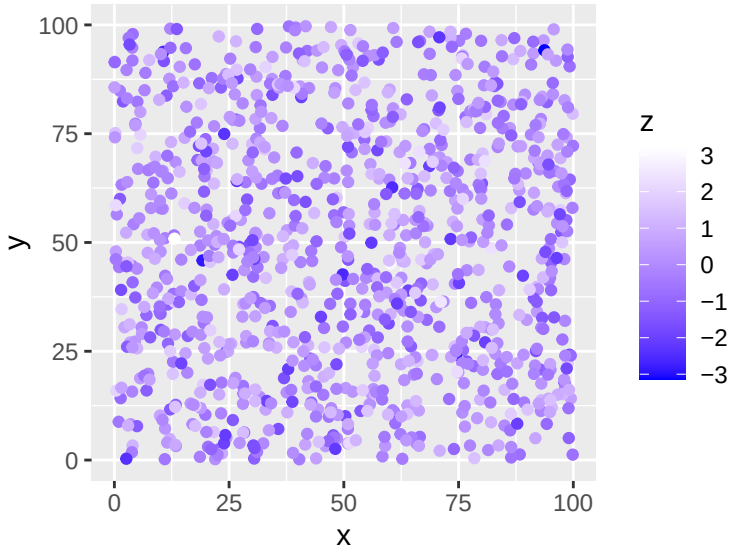
Theory and applications

Joaquin Cavieres G.

Simulated data

```
library(ggplot2)
library(INLA)

# generate some points
set.seed(1234)
n = 1000
x.coord <- runif(1000,0,100)
y.coord <- runif(1000,0,100)
points <- cbind(x.coord,y.coord)

# A random variable with Gaussian distribution
z = rnorm(n,0,1)
```

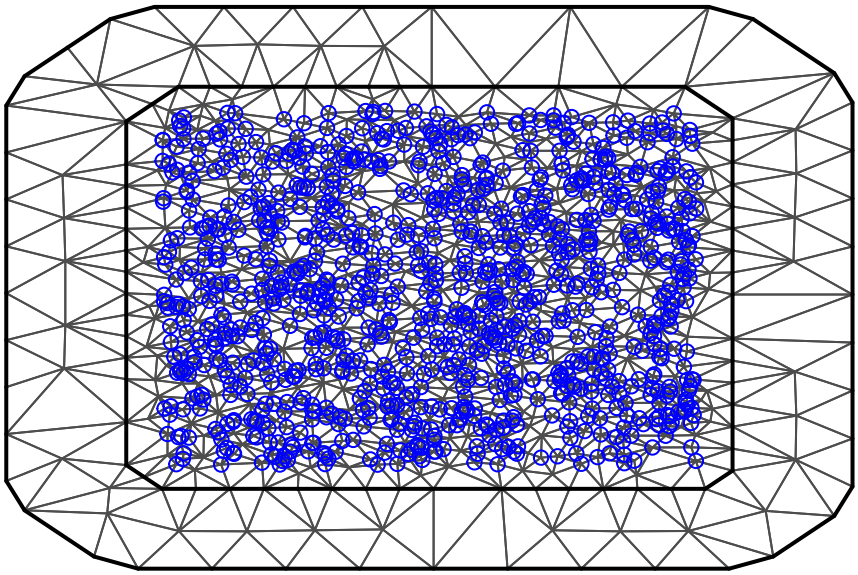
```
library(ggplot2)
library(INLA)
require(ggplot2)

# generate some points
set.seed(1234)
n = 1000
x.coord <- runif(1000,0,100)
y.coord <- runif(1000,0,100)
points <- cbind(x.coord,y.coord)

# A random variable with Gaussian distribution
z = rnorm(n,0,1)

df = data.frame(x = x.coord, y = y.coord, z = z)

ggplot(df, aes(x = x, y = y, col = z)) +
       geom_point() +
       scale_colour_gradient(low="blue", high="white")
```



Now we generating the `mesh` with the coordinates simulated and created the discretized spatial domain with the SPDE method.

```
# Coordinates
coo = cbind(x.coord, y.coord)

# Grid
mesh = inla.mesh.2d(loc = coo, cutoff = 1, max.edge = c(30, 60))
plot(mesh)
points(coo, col = "blue")
```

Constrained refined Delaunay triangulation



```
# SPDE method
spde = inla.spde2.matern(mesh = mesh, alpha = 2, constr = TRUE)

# Spatial index
indexs <- inla.spde.make.index("s", spde$n.spde)

# A matrix
A <- inla.spde.make.A(mesh = mesh, loc = coo)
dim(A)
```

```
## [1] 1000 1281
```

```
nrow(coo)
```

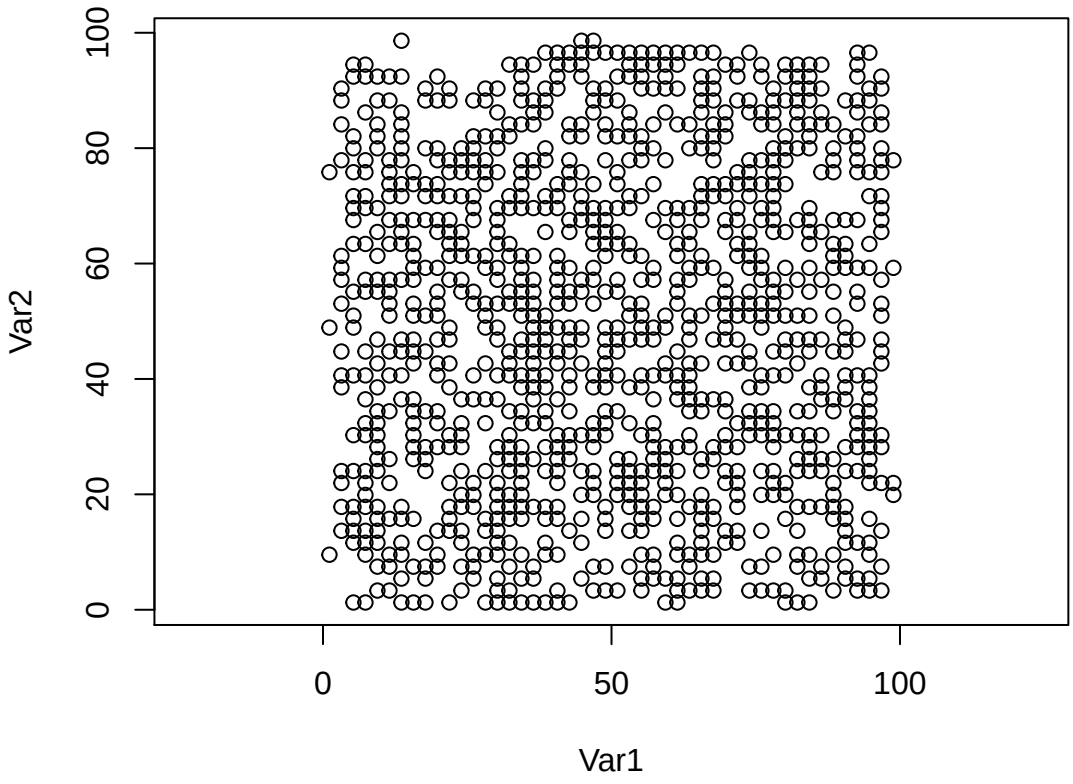
```
## [1] 1000
```

```
mesh$n
```

```
## [1] 1281
```

Borders of the spatial domian and proyection matrix A

```
# Borders of the zone
bb <- bbox(coo)
x <- seq(bb[1, "min"] - 1, bb[1, "max"] + 1, length.out = 50)
y <- seq(bb[2, "min"] - 1, bb[2, "max"] + 1, length.out = 50)
coop <- as.matrix(expand.grid(x, y))

ind <- point.in.polygon(coop[, 1], coop[, 2],
                        coo[, 1], coo[, 2])
coop <- coop[which(ind == 1), ]
plot(coop, asp = 1)
```




```
Ap <- inla.spde.make.A(mesh = mesh, loc = coop)
dim(Ap)
```

```
## [1] 1040 1281
```

`data.stack` for our `data` and prediction.

```r
# stack for estimation stk.e
stk.e <- inla.stack(tag = "est",
                    data = list(y = df$z),
                    A = list(1, A),
                    effects = list(data.frame(b0 = rep(1, nrow(coo))), s = indexs))

# stack for prediction stk.p
stk.p <- inla.stack(tag = "pred",
                    data = list(y = NA),
                    A = list(1, Ap),
                    effects = list(data.frame(b0 = rep(1, nrow(coop))), s = indexs))

# stk.full has stk.e and stk.p
stk.full <- inla.stack(stk.e, stk.p)
```

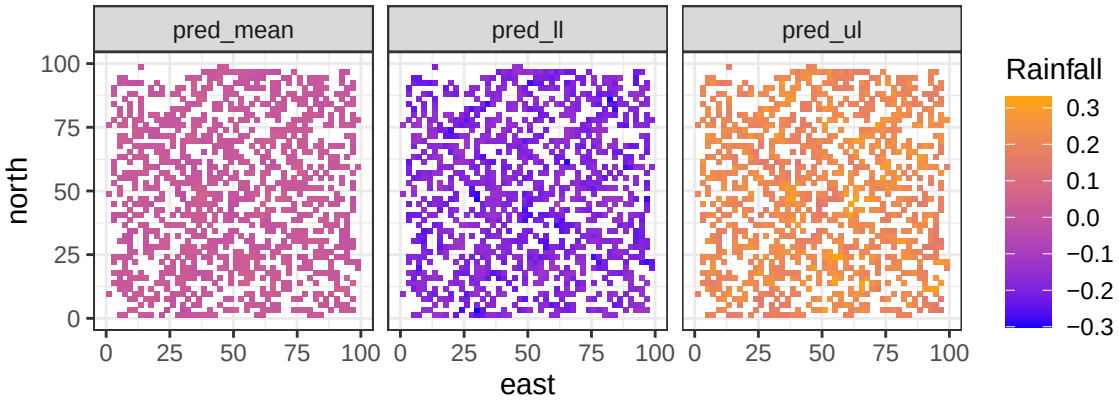
Spatial modelling

```r
# Formula
formula = y ~ 0 + b0 + f(s, model = spde)

# Model
res = inla(formula,
            data = inla.stack.data(stk.full),
            control.predictor = list(compute = TRUE,
            A = inla.stack.A(stk.full)),
            verbose=TRUE)
```

By default we are fitting a Gaussian distribution but can use other likelihood functions (`names(inla.models()$likelihood)`)

```r
# Results
index = inla.stack.index(stk.full, tag = "pred")$data
pred_mean = res$summary.fitted.values[index, "mean"]
pred_ll = res$summary.fitted.values[index, "0.025quant"]
pred_ul = res$summary.fitted.values[index, "0.975quant"]

dpm = rbind(data.frame(east = coop[, 1], north = coop[, 2],
             value = pred_mean, variable = "pred_mean"),
             data.frame(east = coop[, 1], north = coop[, 2],
             value = pred_ll, variable = "pred_ll"),
             data.frame(east = coop[, 1], north = coop[, 2],
                        value = pred_ul, variable = "pred_ul"))
dpm$variable = as.factor(dpm$variable)


ggplot(dpm) + geom_tile(aes(east, north, fill = value)) +
```

```
  facet_wrap(~variable, nrow = 1) +
  coord_fixed(ratio = 1) +
  scale_fill_gradient(
  name = "Rainfall",
  low = "blue", high = "orange") +
  theme_bw()
```



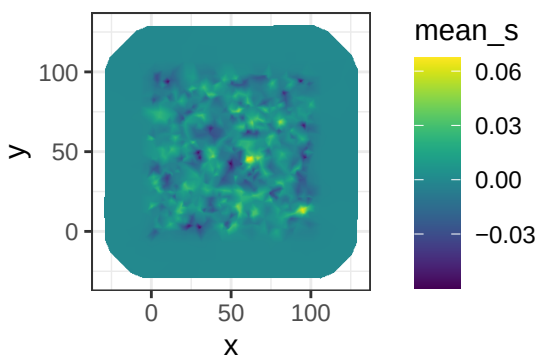
Additional results and visualization.

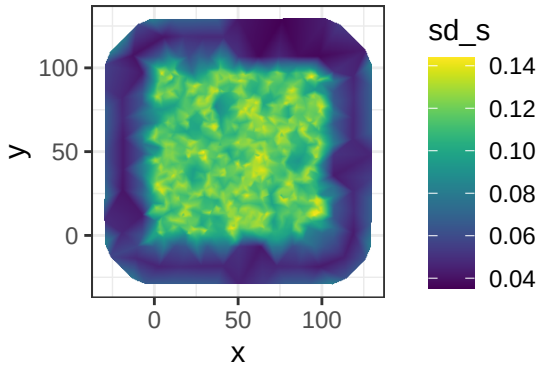
```
# Spatial random field projection
newloc <- cbind(c(219, 678, 818), c(20, 20, 160))
Aproj <- inla.spde.make.A(mesh, loc = newloc)
Aproj %*% res$summary.random$s$mean
```

```
## 3 x 1 Matrix of class "dgeMatrix"
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
```

```
rang <- apply(mesh$loc[, c(1, 2)], 2, range)
proj <- inla.mesh.projector(mesh,
                            xlim = rang[, 1], ylim = rang[, 2],
                            dims = c(300, 300))

mean_s <- inla.mesh.project(proj, res$summary.random$s$mean)
sd_s <- inla.mesh.project(proj, res$summary.random$s$sd)

df <- expand.grid(x = proj$x, y = proj$y)
df$mean_s <- as.vector(mean_s)
df$sd_s <- as.vector(sd_s)
```

```
library(viridis)
library(cowplot)

gmean <- ggplot(df, aes(x = x, y = y, fill = mean_s)) +
  geom_raster() +
  scale_fill_viridis(na.value = "transparent") +
  coord_fixed(ratio = 1) + theme_bw()

gsd <- ggplot(df, aes(x = x, y = y, fill = sd_s)) +
  geom_raster() +
  scale_fill_viridis(na.value = "transparent") +
  coord_fixed(ratio = 1) + theme_bw()

plot_grid(gmean, gsd)
```


mean_s
0.06
0.03
0.00
−0.03
y
100
50
0
0
50
100
x


sd_s
0.14
0.12
0.10
0.08
0.06
0.04
y
100
50
0
0
50
100
x