

Workshop preparation

F. E. Bachl and F. Lindgren

8 June 2017

Packages

The following packages are required to do workshop exercises. Note that even though you might already have them on your system `inlabru` may require the latest version.

From CRAN

Various packages (some already installed packages may be automatically upgraded by the `devtools` commands for non-CRAN packages below):

```
notinstalled <- setdiff(  
  c("devtools", "sp", "ggplot2", "rgdal", "rgeos", "testthat",  
    "ggmap", "rgl", "sphereplot", "raster", "Matrix",  
    "maptools", "mgcv", "shiny", "spatstat", "RColorBrewer"),  
  installed.packages()[,1])  
if (length(notinstalled) > 0) install.packages(notinstalled)
```

Others

The INLA package (Note: you need the latest testing version)

```
install.packages("INLA", repos = "https://inla.r-inla-download.org/R/testing",  
  dependencies = TRUE)
```

The fmesher package

```
devtools::install_bitbucket("finnlindgren/fmesher@default", dependencies = TRUE)
```

The inlabru package (Note: the `devel` branch is required)

```
devtools::install_github("fbachl/inlabru", ref="devel", dependencies = TRUE)
```

Testing your inlabru installation

INLA

A simple call to `inla` to make sure it is installed properly:

```
library("INLA")  
df <- data.frame(y = rnorm(10), x = 1)  
fit <- inla(y ~ x, "gaussian", data = df)  
summary(fit)
```

```
##
## Call:
## "inla(formula = y ~ x, family = \"gaussian\", data = df)"
##
## Time used:
## Pre-processing Running inla Post-processing Total
## 0.2781 0.0751 0.0461 0.3993
##
## Fixed effects:
## mean sd 0.025quant 0.5quant 0.975quant mode kld
## (Intercept) -0.2211 31.6256 -62.3128 -0.2220 61.8187 -0.2211 0
## x 0.0000 31.6228 -62.0862 -0.0009 62.0344 0.0000 0
##
## The model has no random effects
##
## Model hyperparameters:
## mean sd 0.025quant 0.5quant
## Precision for the Gaussian observations 0.6993 0.2915 0.2448 0.6596
## 0.975quant mode
## Precision for the Gaussian observations 1.375 0.5755
##
## Expected number of effective parameters(std dev): 1.00(0.00)
## Number of equivalent replicates : 10.00
##
## Marginal log-Likelihood: -26.70
```

Mesh and SPDE related commands (there is no output but if you receive an error you are in trouble.)

```
loc <- matrix(runif(10*2),10,2)
mesh <- inla.mesh.2d(loc,
                    boundary=list(inla.nonconvex.hull(loc, 0.1, 0.15),
                                  inla.nonconvex.hull(loc, 0.2, 0.2)),
                    max.edge=c(0.05, 0.1))
matern <- inla.spde2.pcmatern(mesh, prior.sigma = c(0.1, 0.01), prior.range = c(5, 0.01))
```

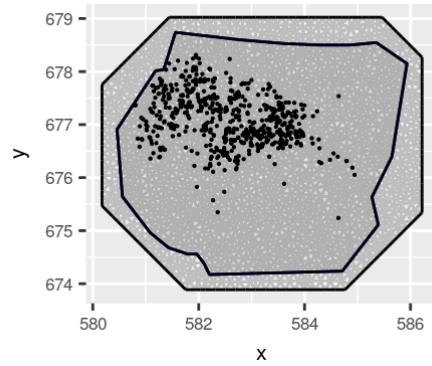
Plotting

Load inlabru

```
library("inlabru")
```

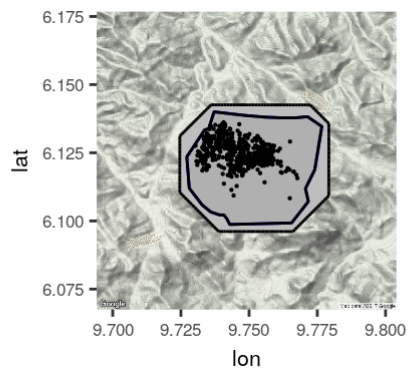
Plot gorilla data using gg

```
data(gorillas, package = "inlabru")
ggplot() +
  gg(gorillas$mesh) +
  gg(gorillas$boundary) +
  gg(gorillas$nest, size = 0.1) +
  coord_equal()
```



Plot using ggmap (requires internet connection)

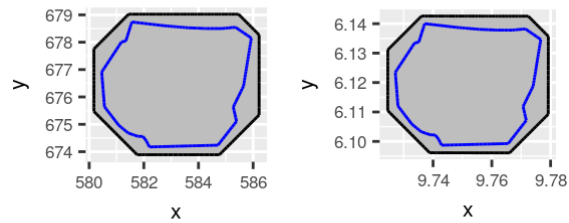
```
library("ggmap")
gmap(gorillas$nest) +
  gm(gorillas$mesh) +
  gm(gorillas$boundary, alpha = 0.1) +
  gm(gorillas$nest, size = 0.1) +
  coord_equal()
```



Coordinate transformations

This checks if rgdal does its job (Notes about unidentified transformation shapes are fine.)

```
tmesh <- sttransform(gorillas$mesh, crs = CRS("+proj=longlat"))
multiplot(ggplot() + gg(gorillas$mesh) + coord_equal(),
  ggplot() + gg(tmesh) + coord_equal(),
  cols = 2)
```



Integration

This checks if `rgeos` does its job

```
ips <- ipoints(gorillas$boundary, gorillas$mesh)
head(ips)
```

```
##              coordinates      weight
## tmp      (584.712, 674.2371) 0.005652837
## tmp.1    (585.3919, 675.11) 0.008666571
## tmp.2    (585.2724, 675.6337) 0.006668490
## tmp.3    (585.6492, 676.3963) 0.006819439
## tmp.4    (585.934, 678.1512) 0.007656232
## tmp.5    (585.3551, 678.5463) 0.029286922
```

Inference

A minimalistic data analysis checking if `inlabru`'s `lgcp` function works (Possible notes about “method with signature ‘sparseMatrix#matrix’ chosen for function” etc are harmless):

```
init.tutorial()

matern <- inla.spde2.pcmatern(gorillas$mesh,
                             prior.sigma = c(0.1, 0.01),
                             prior.range = c(5, 0.01))

cmp <- coordinates ~ mySPDE(map = coordinates,
                             model = matern) +
      Intercept

r <- lgcp(cmp,
          gorillas$nest,
          samplers = gorillas$boundary)

df <- pixels(gorillas$mesh)
pr <- predict(r, df, ~ exp(Intercept + mySPDE))
ggplot() + gg(pr)
```

