mvn_mvt_inlaTMB

Overview

This markdown file will implement the INLA / TMB approach to the spatial models. We'll leave this as a separate document for now so that all of the other models don't have to be run in JAGS or STAN. These code blocks are copied over from JAGS/STAN markdown file.

The only major differences are that TMB and INLA both have to be installed. INLA can be installed by running the line source("http://www.math.ntnu.no/inla/givemeINLA.R"). TMB can be installed from GitHub (install_github("kaskr/adcomp/TMB")) or from CRAN ("TMB").

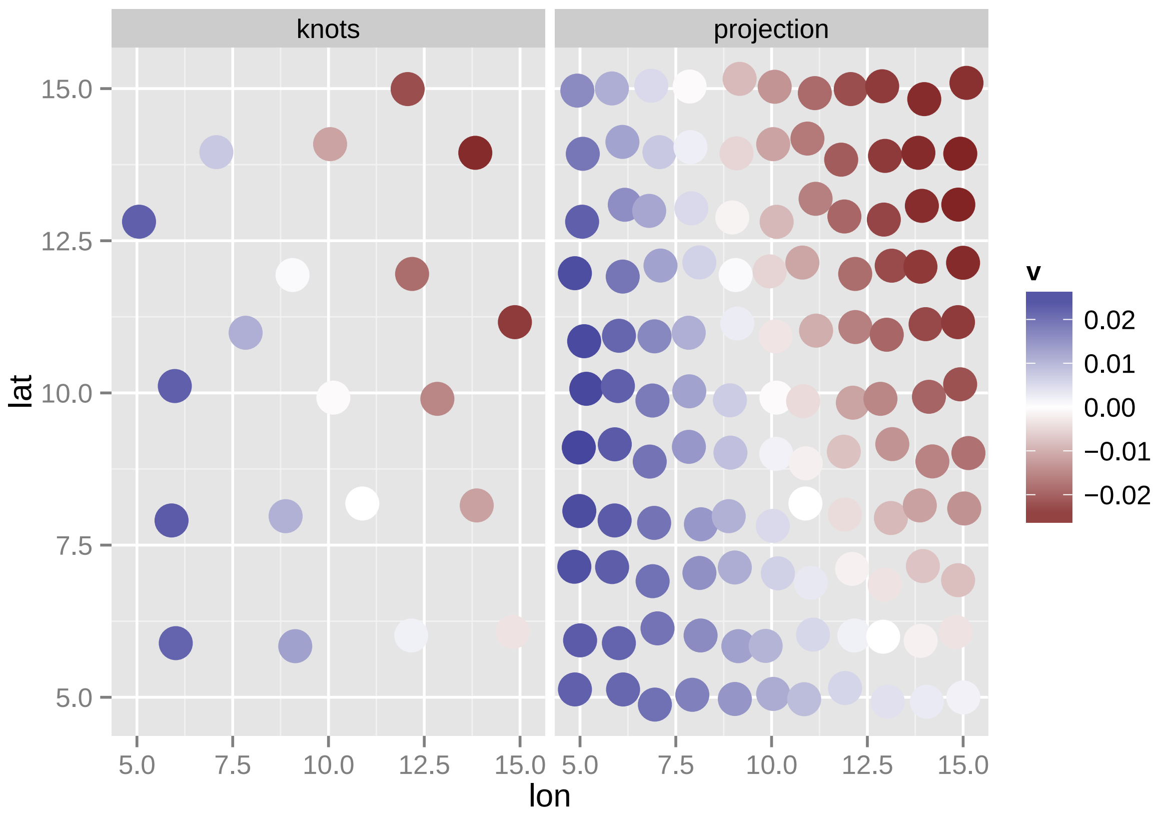
Spatial data simulation

We'll start and set the seed,

```
set.seed(3)
```

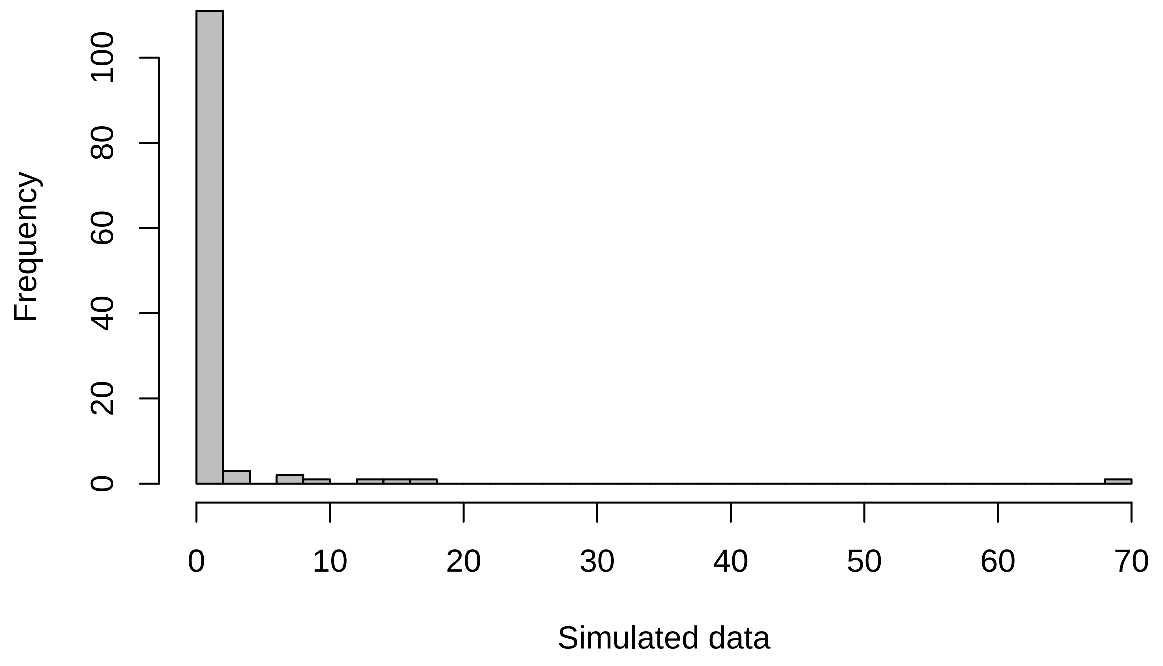
```
# Simulate data on grid
grid = as.matrix(expand.grid(lon = seq(5, 15, 1), lat = seq(5, 15, 1)))
grid[, 1] = jitter(grid[, 1])
grid[, 2] = jitter(grid[, 2])
nLocs = dim(grid)[1]

nKnots = 20  # Dimension of random effects
knots = pam(grid, nKnots)$medoids
distKnots = as.matrix(dist(knots))
distKnotsSq = distKnots^2  # squared distances
# note: shape parameter scaled to distance matrix
gp_scale = 0.01
sigma.norm = 0.01
corKnots = exp(-gp_scale * distKnotsSq)
Sigma.normal = corKnots * sigma.norm * sigma.norm
invSigmaKnots.norm = solve(Sigma.normal)
# Calculate distance from knots to grid
distAll = as.matrix(dist(rbind(grid, knots)))^2
distKnots21Sq = t(distAll[-c(1:nLocs), -c((nLocs + 1):ncol(distAll))])
Sigma21.normal = exp(-gp_scale * distKnots21Sq) * sigma.norm * sigma.norm
# Generate vector of random effects
re.norm = rmvt(1, sigma = Sigma.normal, df = 2)
re.norm = re.norm - mean(re.norm)  # Scale

# Project random effects to locations of the data
proj.norm = t((Sigma21.normal %*% invSigmaKnots.norm) %*% t(re.norm))

diagKnots = diag(nKnots)
nPoints = length(proj.norm)
muZeros = rep(0, nKnots)
indices = seq(1, nPoints)
```

```r
k <- data.frame(knots, v = 1 * t(re.norm), type = "knots")
p <- data.frame(grid, v = t(proj.norm), type = "projection")
d <- rbind(k, p)
ggplot(d, aes(lon, lat, colour = v)) + facet_wrap(~type) + geom_point(size = 6) +
    scale_colour_gradient2()
```



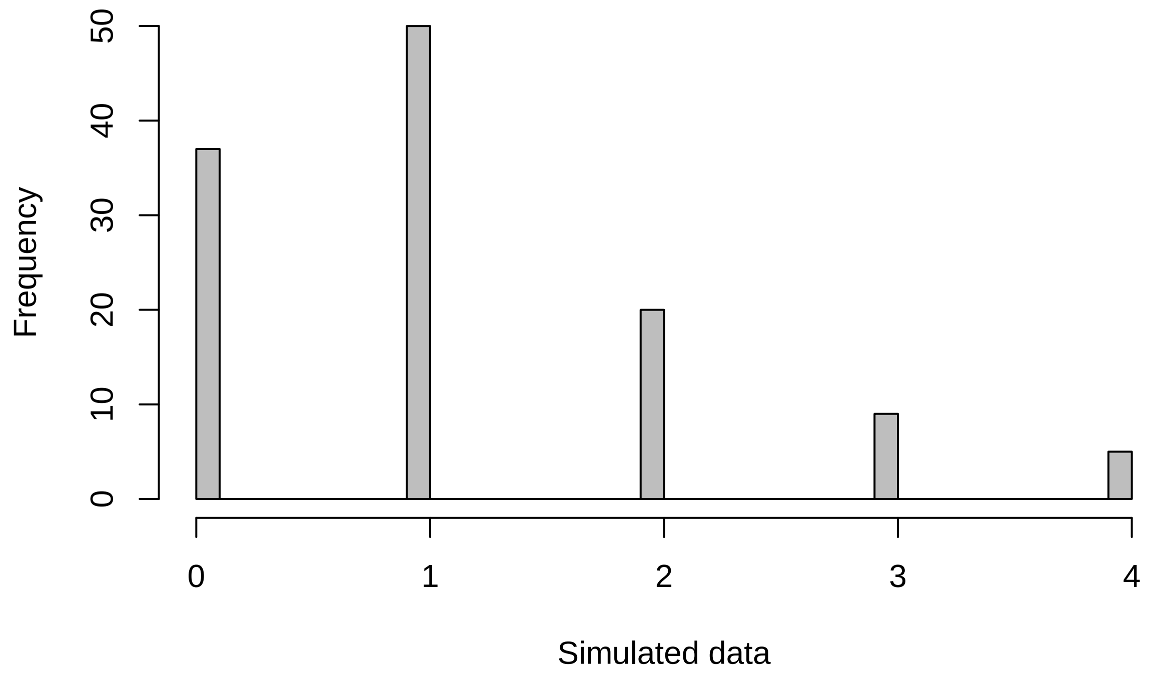
Simulating data with Gamma observation model

```r
# Include observation error Use same gamma parameterization as JAGS
gamma.a = 0.03
gamma.b = gamma.a/exp(proj.norm)
# simulate observed data on grid
y.gamma = rgamma(length(proj.norm), shape = gamma.a, rate = gamma.b)

hist(y.gamma, 40, col = "grey", xlab = "Simulated data", main = "")
```



Simulating data with Poisson observation model

```r
# Include observation error simulate observed data on grid
y.poisson = rpois(length(proj.norm), exp(proj.norm))

hist(y.poisson, 40, col = "grey", xlab = "Simulated data", main = "")
```



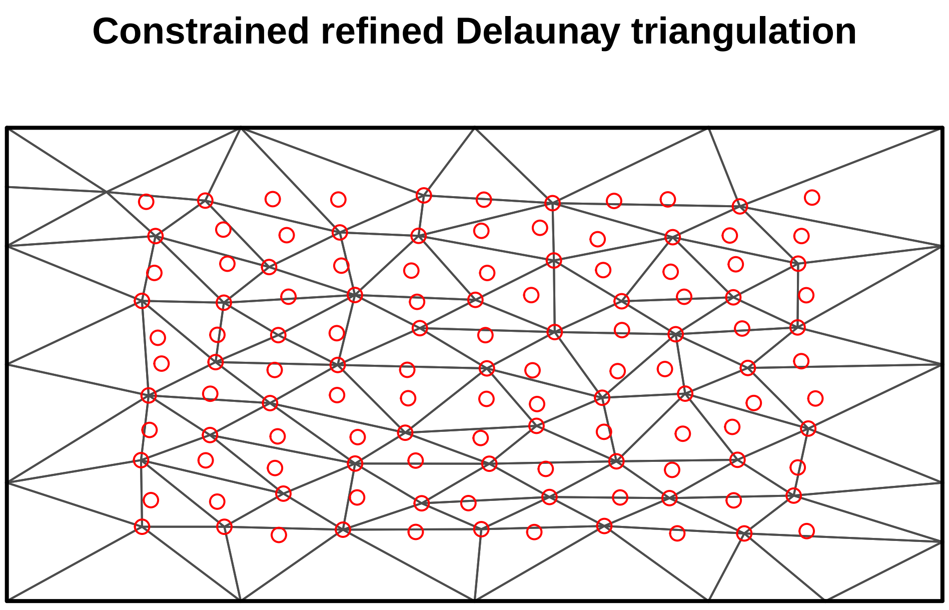
Using INLA to calculate matrices

We'll make this example slighty more complicated than before by including data from several years.

```r
n_years = 3
n_stations = nLocs
n_data = n_stations * n_years
Ymat = matrix(0, nLocs, n_years)
for (yr in 1:n_years) {
    Ymat[, yr] = rgamma(nLocs, shape = gamma.a, rate = gamma.b)
}

Site = as.vector(row(Ymat))
Year = as.vector(col(Ymat))
# simulate some NAs
y.gamma = c(Ymat)
y.gamma[sample(seq(1, length(y.gamma)), size = 40, replace = F)] = NA
NAindicator = as.integer(ifelse(is.na(y.gamma), 1, 0))
```

Build SPDE object using INLA

```r
# These are just some default options. Very specific to this simulation
mesh = inla.mesh.create(grid, plot.delay = NULL, extend = list(n = 4, offset = -0.2),
    refine = list(min.angle = 26), cutoff = 1.2)
plot(mesh)
points(grid, col = "red")
```



```r
print(mesh$n)  # number of knots
```

```
## [1] 70
```

Create the SPDE/GMRF model

Matern covariance function

```r
spde = inla.spde2.matern(mesh, alpha = 2)
```