DSTSOM: Dynamic Spatio-Temporal Sequential Ordinal Models

Geoffrey R. Hosack, Wen-Hsi Yang and Kyana N. Pike

31 August 2025

1 DSTSOM R package

Please cite the R package "DSTSOM: Dynamic Spatio-Temporal Sequential Ordinal Models" using the following citation:

```
citation("DSTSOM")

## To cite package 'DSTSOM' in publications use:
##
##   Hosack GR, Yang W, Pike KN (2025). _DSTSOM: Dynamic Spatio-Temporal
##   Sequential Ordinal Models_. R package version 0.1.0,
##   <https://github.com/csiro/DSTSOM>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {DSTSOM: Dynamic Spatio-Temporal Sequential Ordinal Models},
##     author = {Geoffrey R. Hosack and Wen-Hsi Yang and Kyana N. Pike},
##     year = {2025},
##     note = {R package version 0.1.0},
##     url = {https://github.com/csiro/DSTSOM},
##   }
```

2 Ordinal Data

Load package and data for analysis:

```
library(DSTSOM)
library(sf)

## Linking to GEOS 3.12.0, GDAL 3.7.2, PROJ 9.3.0; sf_use_s2() is TRUE

library(terra)

## terra 1.7.78
##
## Attaching package:  'terra'
```

```
## The following object is masked from 'package:knitr':
##
##     spin

library(lattice)

# dat: floristics, control and covariate data
data(ACT_FloristicSurveys, package = "DSTSOM")
dat <- ACT_FloristicSurveys

# transform to kms from metres for numerical stability
dat$easting <- dat$easting/1000
dat$northing <- dat$northing/1000
dat$dist2road_track_trail <- dat$dist2road_track_trail/1000

# act.mga: ACT boundary
data(ACT_Border, package = "DSTSOM")
# transform to kms from metres for numerical stability
act.mga <- st_transform(ACT_Border, gsub("units=m", "units=km",
                                         st_crs(ACT_Border)$proj4string))
act.mga

## Simple feature collection with 1 feature and 0 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 659.8908 ymin: 6022.933 xmax: 718.1073 ymax: 6111.102
## Projected CRS: +proj=utm +zone=55 +south +ellps=GRS80 +units=km +no_defs
##                          geometry
## 1 POLYGON ((708.8498 6087.599...

# covariate raster
ff <- system.file("extdata/ACT_Covariates.tif", package="DSTSOM")
covars <- terra::rast(ff)
# transform to kms from metres for consistency and stability
covars <- project(covars, gsub("units=m", "units=km",
                               crs(covars, proj = TRUE)))
covars$dist2road_track_trail <- covars$dist2road_track_trail/1000
covars

## class       : SpatRaster
## dimensions  : 882, 583, 2  (nrow, ncol, nlyr)
## resolution  : 0.09993254, 0.09993254  (x, y)
## extent      : 659.8908, 718.1514, 6022.962, 6111.102  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=55 +south +ellps=GRS80 +units=km +no_defs
## source(s)   : memory
## names       : ForestNGrass, dist2road_track_trail
## min values  :       Forest,           0.005085415
## max values  :       Others,           4.094660645
```

The ACT boundary data and the covariate raster have been projected to the coordinate reference system of GDA2020 / MGA zone 55 (i.e. EPSG:7855). In the floristics data, the subplot coordinates based on EPSG:7855 are in the columns 'easting' and 'northing'.

```
plot(st_geometry(act.mga), axes=T)
points(dat$easting, dat$northing, pch=20,
       col=palette.colors(alpha=0.3)[3])
```

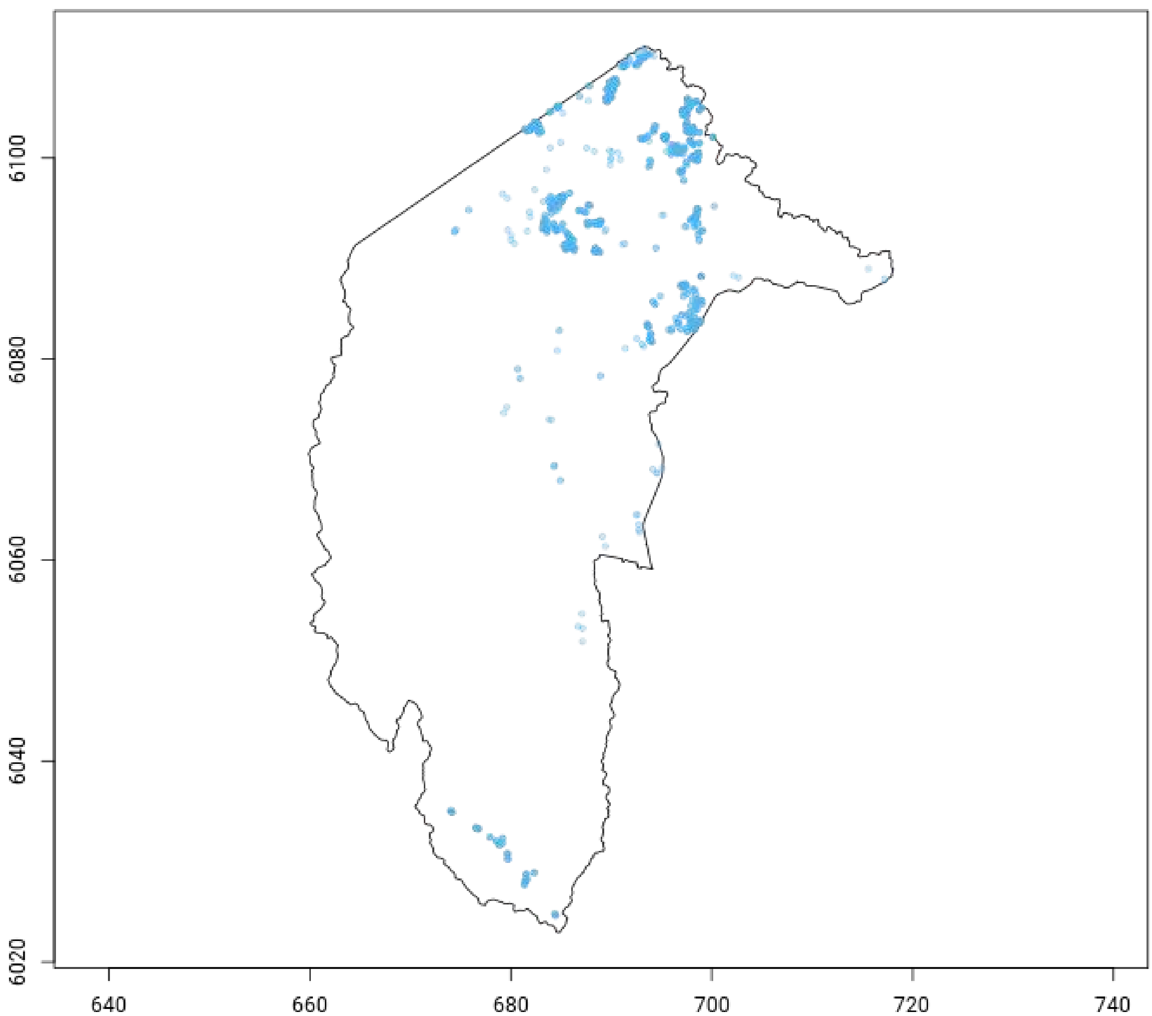


Figure 1: The subplot locations in the floristic survey data. Based on data provided by the ACT Government (see license).

The number of observation by year. These observations typically occurred at unique subplot locations with very few return visits to the same subplot in a year.

Table 1: The modified Braun-Blanquet score.

Score	Explanation
1	$<5\%$ cover and solitary (<4 individuals)
2	$<5\%$ cover and few (4-15 individuals)
3	$<5\%$ cover and numerous/scattered (>15 individuals)
4	5% - $<25\%$ cover
5	25%- $<50\%$ cover
6	50%- $<75\%$ cover
7	75% cover and greater.

```
# number of observations by year
table(dat$year)

##
## 2019 2020 2021 2022 2023
##  334  323  181  208  253

# number of observations at unique locations by year
tapply(paste(dat$easting, dat$northing), list(dat$year), function(x) length(unique(x)))

## 2019 2020 2021 2022 2023
##  327  322  172  203  252
```

The data are scored for each species according to a modified Braun-Blanquet score (Table 1). However, this scoring system was augmented by occasional scores of zero, whereas other observations are missing. For example, the proportion of recorded zero observations for *Nasella trichotoma* (serrated tussock) was 0.78, whereas a proportion of 0 of observations was recorded as missing.

If a recorded modified Braun-Blanquet score of "1" is regarded as an observed presence, while zeroes and missing observations are both assigned to an additional "0" category that corresponds to an observed absence, then the observed aggregate scores for *N. trichotoma* are

```
table(dat$nassella_trichotoma_0)

##
##    0    1    2    3    4    5    7
## 1013  160   70   46    7    2    1
```

Less than ten observations occur for any score of "4" or above. For analysis, an aggregated modified Braun-Blanquet score is considered, where score "4" will be assigned to observations with scores "4" or greater. These levels correspond to observations with greater than 5% cover (Table 1). One is then added to this scoring system for consistency with ordinal analysis, where ordinal models typically begin enumeration of arbitrary categories from "1". The response variable y has categories $k = 1, \ldots, 5$:

```
# serrated tussock
dat$y_St <- dat$nassella_trichotoma_0
dat$y_St <- ifelse(dat$y_St >= 4, 4, dat$y_St)
```

```
dat$y_St <- dat$y_St + 1
table(dat$y_St)

##
##    1    2    3    4    5
## 1013  160   70   46   10

# also build observations for a few other species:
 #  AGC: African love grass
 #  CNG: Chilean needle grass
 #  StJW: St Johns wort

dat$y_AGC <- dat$eragrostis_curvula_0
dat$y_AGC <- ifelse(dat$y_AGC >= 4, 4, dat$y_AGC)
dat$y_AGC <- dat$y_AGC + 1
table(dat$y_AGC)

##
##    1    2    3    4    5
## 1058   74   72   55   40

dat$y_CNG <- dat$nassella_neesiana_0
dat$y_CNG <- ifelse(dat$y_CNG >= 4, 4, dat$y_CNG)
dat$y_CNG <- dat$y_CNG + 1
table(dat$y_CNG)

##
##    1    2    3    4    5
## 1122   38   36   72   31

dat$y_StJW <- dat$hypericum_perforatum_0
dat$y_StJW <- ifelse(dat$y_StJW >= 4, 4, dat$y_StJW)
dat$y_StJW <- dat$y_StJW + 1
table(dat$y_StJW)

##
##   1   2   3   4   5
## 501 134 183 421  60
```

3 Covariates

Vegetation structure data provided by the ACT Government were modified to identify the combination of all woodland and forest vegetation communities, henceforth denoted as “forest” habitat, grassland habitat, or other habitats. This new vegetation type and three disturbances distance to nearest road, trail or track are rasterised in a 100 m x 100 m resolution.

3.1 Vegetation type

```
with(dat, table(veg_structure, ForestNGrass, useNA="ifany"))

##                                    ForestNGrass
## veg_structure                        Forest Grassland Others
##   Exotic Forest                           3         0      0
##   Exotic Grassland                        0        91      0
##   Grassland                               0       435      0
##   Grassland or Secondary Woodland         0       310      0
##   Open Forest                            52         0      0
##   Plantation                              9         0      0
##   Planting                               31         0      0
##   Urban                                   0         0     32
##   Woodland                              336         0      0
```

```
plot(covars, "ForestNGrass", col=palette.colors(alpha=0.8)[c(4,5,7)])
plot(sf::st_geometry(act.mga), add=T)
points(dat$easting, dat$northing, pch=20,
       col=palette.colors(alpha=0.2)[1])
```

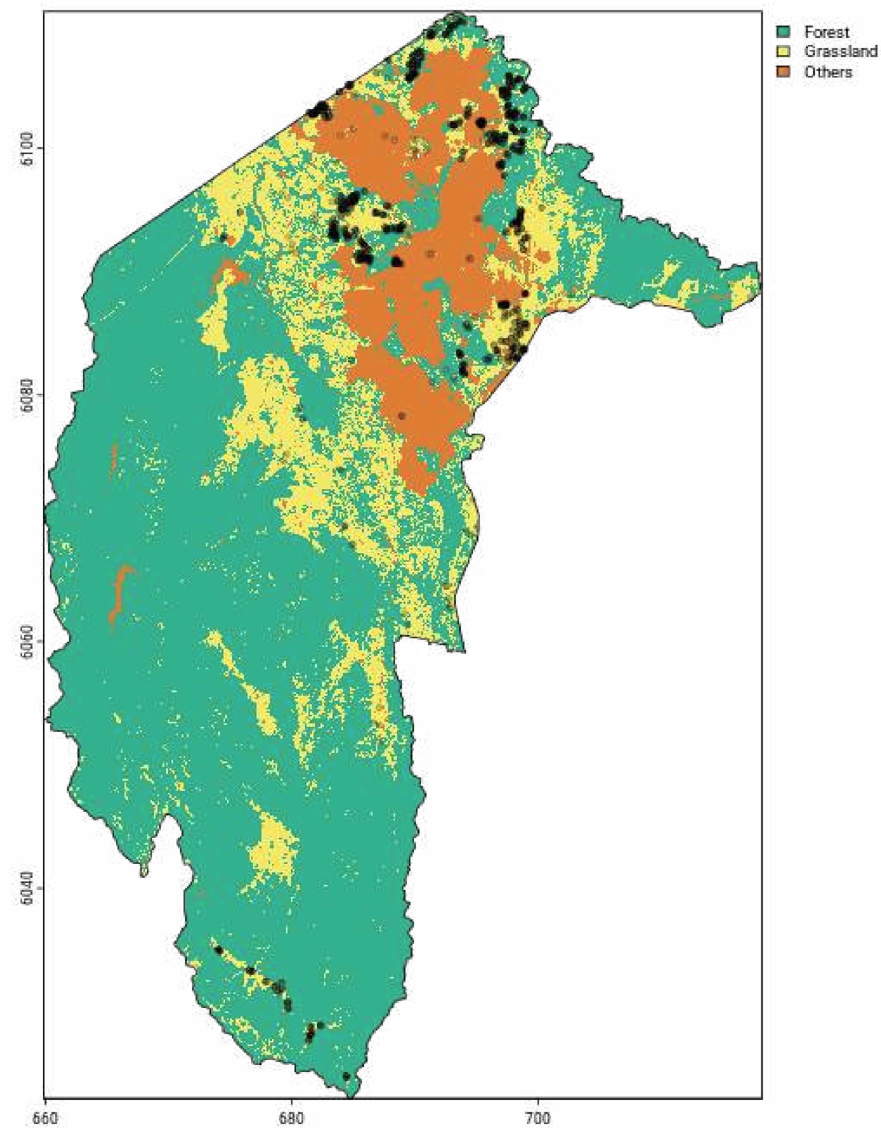


Figure 2: Distribution of forest, grassland and others as well and subplot locations. Based on data provided by the ACT Government (see license).

The “Others” category corresponds with “Urban” habitat. These unique sites are located near the urban boundary and are dropped from the analysis: too few data points for reliable prediction or estimation in urban habitat. The remaining habitats are scored zero for grassland and one for forest. Sites without habitat classifications are dropped from the analysis.

```
plot(covars, "ForestNGrass", col=palette.colors(alpha=0.8)[c(4,5,7)])
plot(sf::st_geometry(act.mga), add=T)
o.r <- extract(covars, cbind(dat$easting, dat$northing))
points(dat$easting[o.r$ForestNGrass == "Others"],
       dat$northing[o.r$ForestNGrass == "Others"]
)
```

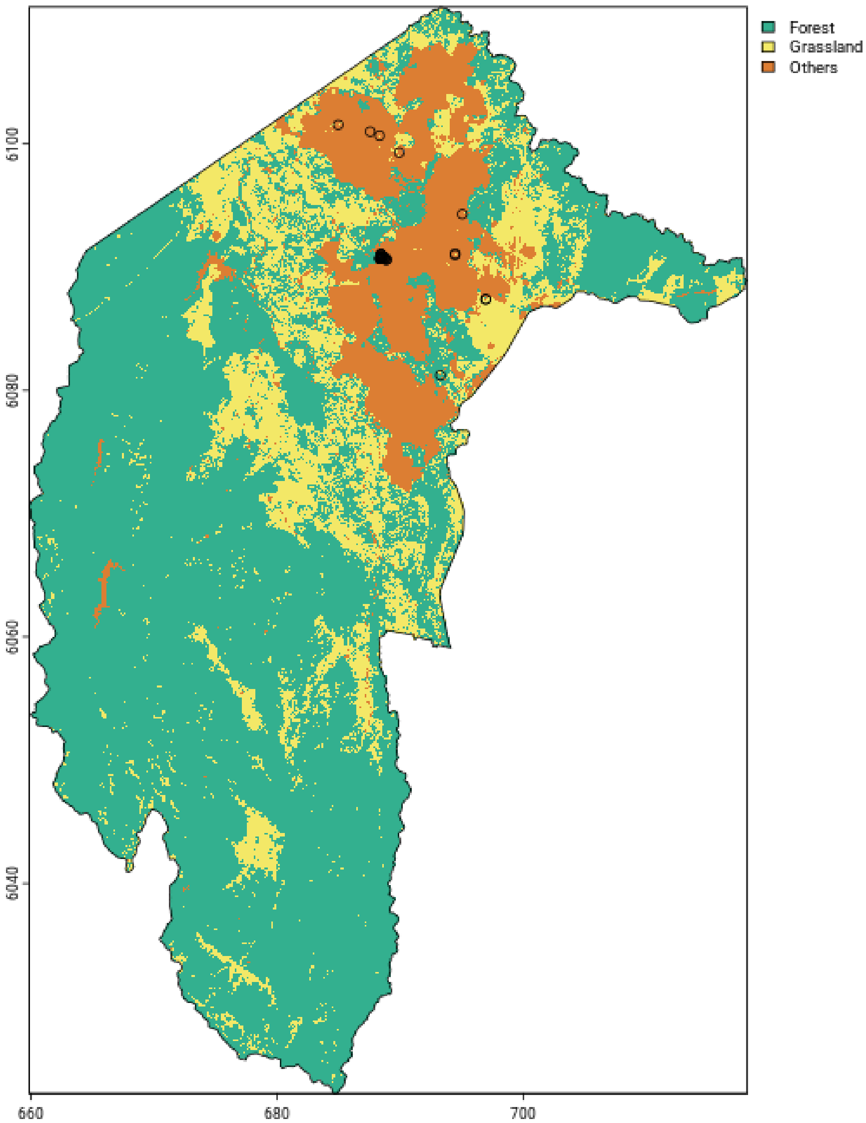


Figure 3: Distribution of forest, grassland and points in urban landscape as well and subplot locations. Based on data provided by the ACT Government (see license).

```
dat <- dat[dat$ForestNGrass != "Others" & !is.na(dat$ForestNGrass), ]
dat$forest <- ifelse(dat$ForestNGrass == "Forest", 1, 0)

colSums(with(dat, table(veg_structure, ForestNGrass, useNA="ifany")))

##    Forest Grassland
##       431       836

sum(dat$forest, na.rm = TRUE)

## [1] 431
```

```r
sum(dat$forest == 0, na.rm = TRUE)

## [1] 836
```

3.2 Minimum distance to a road, trail or track

```r
# transformed to km
with(dat, summary(dist2road_track_trail))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0010  0.0590  0.1068  0.1778  0.2098  1.9607
```

```r
plot(covars, "dist2road_track_trail", col= hcl.colors(32, palette = "Greens 3",alpha=0.9)[32:1
plot(sf::st_geometry(act.mga), add=T)
points(dat$easting, dat$northing, pch=20,
       col=palette.colors(alpha=0.2)[3])
```

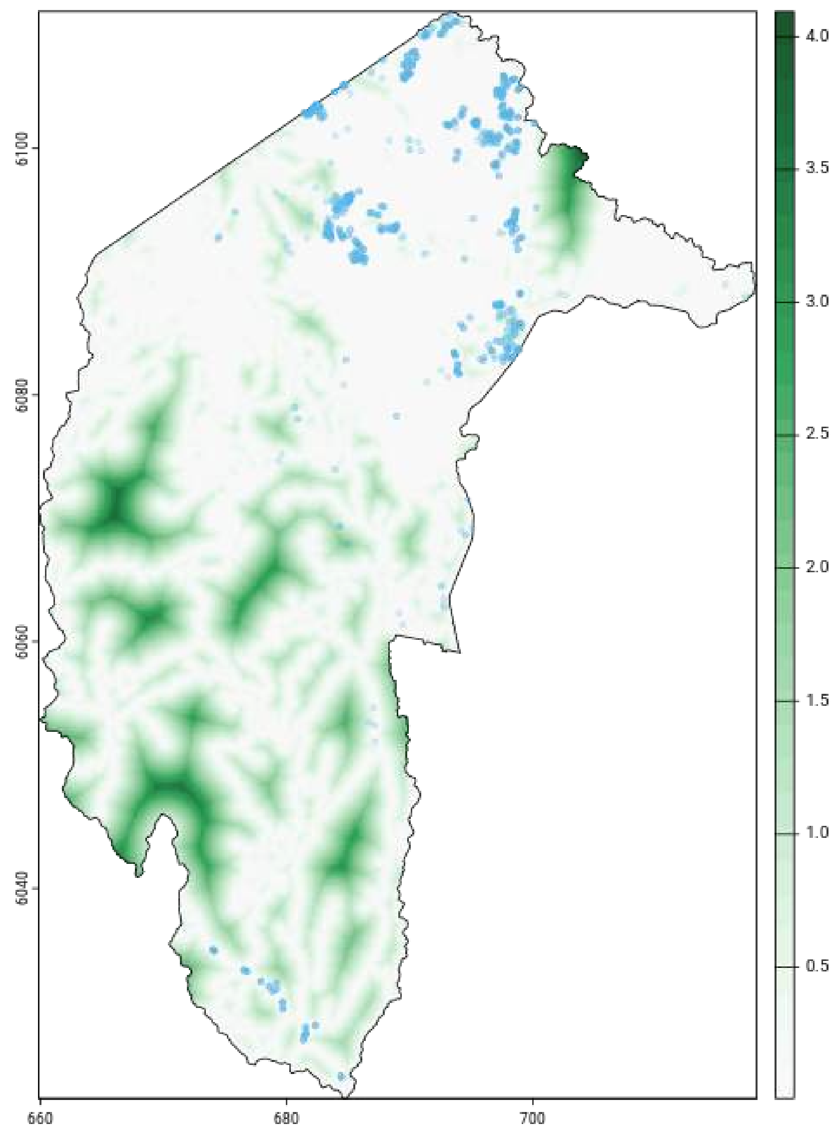


Figure 4: Map of minimum distance to a road, trail or track in kilometres and subplot locations. Based on data provided by the ACT Government (see license).

4 Sequential Ordinal Model and Binary Likelihood Example

For analysis of the ordinal data, we choose a sequential ordinal model that yields unconstrained threshold parameters θ (cut points) for the boundaries between ordinal categories. The following example reproduces the analysis of Example 9.6 in Tutz (2012) using the binary likelihood construction. The example uses a sequential logit model.

```
library(VGAM)

## Loading required package:  stats4
```

```
## Loading required package:  splines
```

```
library(catdata)
```

```
## Loading required package:  MASS
##
## Attaching package:  'MASS'
## The following object is masked from 'package:terra':
##
##     area
```

```
data("retinopathy")

# see catdata vignette: Retinopathy - Sequential Logit Models
seqm4 <- vglm(RET ~ SM + DIAB + GH + BP,
              family = sratio (link="logitlink",
                      parallel=FALSE ~ SM), data = retinopathy)

# binary model
r <- retinopathy
r$RET <- r$RET + 1
rbin <- dfBinary(r, "RET")

sl4 <- glm(cbind(rbin$y_binary, 1 - rbin$y_binary) ~ rbin$cut_1 + rbin$cut_2 +
            rbin$cut_1:rbin$SM + rbin$cut_2:rbin$SM  +
             rbin$DIAB + rbin$GH + rbin$BP - 1,
          family = binomial("logit"))

summary(seqm4)
```

```
## Call:
## vglm(formula = RET ~ SM + DIAB + GH + BP, family = sratio(link = "logitlink",
##     parallel = FALSE ~ SM), data = retinopathy)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept):1 11.12783    1.16861   9.522  < 2e-16 ***
## (Intercept):2 10.91554    1.21342   8.996  < 2e-16 ***
## SM:1          -0.37755    0.20248  -1.865   0.0622 .
## SM:2           0.49077    0.31285   1.569   0.1167
## DIAB          -0.12823    0.01229 -10.430  < 2e-16 ***
## GH            -0.42480    0.06730  -6.312 2.76e-10 ***
## BP            -0.06227    0.01220  -5.104 3.33e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y=1|Y>=1]), logitlink(P[Y=2|Y>=2])
##
## Residual deviance: 891.9767 on 1219 degrees of freedom
```

```
##
## Log-likelihood: -445.9884 on 1219 degrees of freedom
##
## Number of Fisher scoring iterations: 6
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):1'
##
##
## Exponentiated coefficients:
##      SM:1      SM:2      DIAB        GH        BP
## 0.6855376 1.6335785 0.8796526 0.6539010 0.9396286
```

```
summary(sl4)
```

```
##
## Call:
## glm(formula = cbind(rbin$y_binary, 1 - rbin$y_binary) ~ rbin$cut_1 +
##     rbin$cut_2 + rbin$cut_1:rbin$SM + rbin$cut_2:rbin$SM + rbin$DIAB +
##     rbin$GH + rbin$BP - 1, family = binomial("logit"))
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## rbin$cut_1         11.12781    1.16547   9.548  < 2e-16 ***
## rbin$cut_2         10.91552    1.21135   9.011  < 2e-16 ***
## rbin$DIAB          -0.12823    0.01271 -10.086  < 2e-16 ***
## rbin$GH            -0.42480    0.06833  -6.217 5.07e-10 ***
## rbin$BP            -0.06227    0.01228  -5.072 3.94e-07 ***
## rbin$cut_1:rbin$SM -0.37755    0.20277  -1.862   0.0626 .
## rbin$cut_2:rbin$SM  0.49077    0.31259   1.570   0.1164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1161.71  on 838  degrees of freedom
## Residual deviance:  891.98  on 831  degrees of freedom
## AIC: 905.98
##
## Number of Fisher Scoring iterations: 4
```

For completeness, the above construction of the binary likelihood is compared against a similar presentation in the literature. Berridge and Whitehead (1991) were the first to propose the binary regression trick for sequential ordinal models in the context of a cohort study. This formulation is implemented in function `cr.setup` of package **rms** (Harrell Jr, 2023).

```
# package rms required
if (require("rms")) {
  y <- r$RET
  y.cr <- rms::cr.setup(y)
  all.equal(y.cr$y, rbin$y_binary)
}

## Loading required package:  rms
## Loading required package:  Hmisc
##
## Attaching package:  'Hmisc'
## The following objects are masked from 'package:terra':
##
##     describe, mask, units, zoom
## The following objects are masked from 'package:base':
##
##     format.pval, units
##
## Attaching package:  'rms'
## The following objects are masked from 'package:VGAM':
##
##     calibrate, lrtest

## [1] TRUE
```

5 INLA syntax

Care must be taken with the INLA implementation given the shared random effect (spatial and/or temporal) across all binary "pseudo-observations" that share the same value of `index`.

The ordinal model is effectively a truncated multinomial model that suggests a multivariate implementation in INLA (sensu Palmi-Perales et al., 2022). Alternatively, the binary likelihood trick reduces the problem to a univariate binary regression. In a spatial model, the same random effect will be shared across multiple pseudo-observations. Thus it may be appropriate to consider approaches such as `copy` (Krainski et al., 2019) or `replicate` (Gómez-Rubio, 2020). Below, identical random effects are shared across multiple pseudo-observations using the projection matrix in INLA.

6 Synthetic Data Example

Simulate hypothetical dataset and fit models. First, construct ordinal observations with known spatio-temporal dependence and fixed effects.

```
library(INLA)

## Loading required package:  Matrix
## Loading required package:  sp
```

```
## This is INLA_24.09.14 built 2024-09-14 13:49:48 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
##  - List available models/likelihoods/etc with inla.list.models()
##  - Use inla.doc(<NAME>) to access documentation
```

```
set.seed(999)
n.sites <- 1000
sites <- matrix(runif(n.sites*2)*10000, nrow = n.sites, ncol = 2)

# separable spatio-temporal random field parameters
s.range <- 500 # Matern spatial range
s.stdev <- 0.25   # Matern st dev
rho.t <- 0.90     # temporal autocorrelation

# "fixed" parameters
parms <- c(-5, -2, 0, 1, -1, 0.33, 0.4, -0.5, 0.25)
names(parms) <- c(
  paste("cut", 1:4, sep = "_"), "ctrl", "d", "year", "forest", "log_access"
)
parms[5:length(parms)] <- -parms[5:length(parms)]

# spatial covariance
Sigma <- inla.matern.cov(
  nu = 1,
  kappa = sqrt(8*1)/s.range,
  x = as.matrix(dist(sites)),
  d = 2,
  corr = TRUE
)*s.stdev^2

# over 5 years
sim.df <- expand.grid(
  sites = 1:n.sites,
  year = 1:5
)
# coordinates
sim.df$easting <- sites[ , 1]
sim.df$northing <- sites[ , 2]
# spatial covariates
sim.df$log_access <- rep(rlnorm(n.sites, log(4.5), 1), times = 5)
sim.df$forest <- rep(rbinom(n.sites, 1, prob = 1/3), times = 5)

# simulated spatio-temporal control effort
# day of control events relative to start of observation period
# control events occur with equal prob over ten year period that begins
# two years prior to first observation
year.control <- sample(-2:7, n.sites, replace = TRUE)
# time since last control event
```

```
year.since.control <- sim.df$year - rep(year.control, times = 5)
# control effort is observed if it occurs in prior year
# otherwise no control (yet)
obs.year.since.control <- ifelse(year.since.control > 0, year.since.control, 0)
# control covariate: binary score for occurrence or not
sim.df$ctrl <- ifelse(obs.year.since.control > 0, 1, 0)
# duration since control effort in years
sim.df$d <- ifelse(obs.year.since.control > 0, obs.year.since.control, 0)

# simulate ordinal observations
sim.df$y_sim <- NA

# cut levels in global model
cuts <- matrix(parms[paste("cut", 1:4, sep = "_")],
               nrow = n.sites, ncol = 4, byrow = TRUE)

# year one
# spatial random effects
w <- t(chol(Sigma))%*%rnorm(n.sites)
# covariate fixed effects
fix.lp <- as.matrix(
    sim.df[sim.df$year == 1, c("year", "log_access", "forest", "ctrl", "d")]
  )%*%parms[c("year", "log_access", "forest", "ctrl", "d")]
# add cut levels, global fixed effects and spatial random effects
lp <- cuts - c(fix.lp) - c(w)
# inverse cloglog
seq.probs <- 1 - exp(-exp(lp))
# sequential realisations
seq.sim <- matrix(
  rbinom(prod(dim(seq.probs)), 1, prob = seq.probs),
  nrow = n.sites
)
ind.seq.sim <- apply(cbind(seq.sim, 1), 1,
                     function(x) min(which(x == 1, arr.ind = TRUE)))
sim.df[sim.df$year == 1, "y_sim"] <- ind.seq.sim

# successive years
for (yy in 2:5) {
  # spatiotemporal random effects
  w <- rho.t*w + t(chol(Sigma))%*%rnorm(n.sites)
  # covariate fixed effects
  fix.lp <- as.matrix(
      sim.df[sim.df$year == yy, c("year", "log_access", "forest", "ctrl", "d")]
    )%*%parms[c("year", "log_access", "forest", "ctrl", "d")]
  # add cut levels, global fixed effects and spatial random effects
  lp <- cuts - c(fix.lp) - c(w)
  # inverse cloglog
```

```
  seq.probs <- 1 - exp(-exp(lp))
  # sequential realisations
  seq.sim <- matrix(
    rbinom(prod(dim(seq.probs)), 1, prob = seq.probs),
    nrow = n.sites
  )
  ind.seq.sim <- apply(cbind(seq.sim, 1), 1,
                       function(x) min(which(x == 1, arr.ind = TRUE)))
  sim.df[sim.df$year == yy, "y_sim"] <- ind.seq.sim
}
```

Visualise synthetic data in each of the five simulated years from left to right.

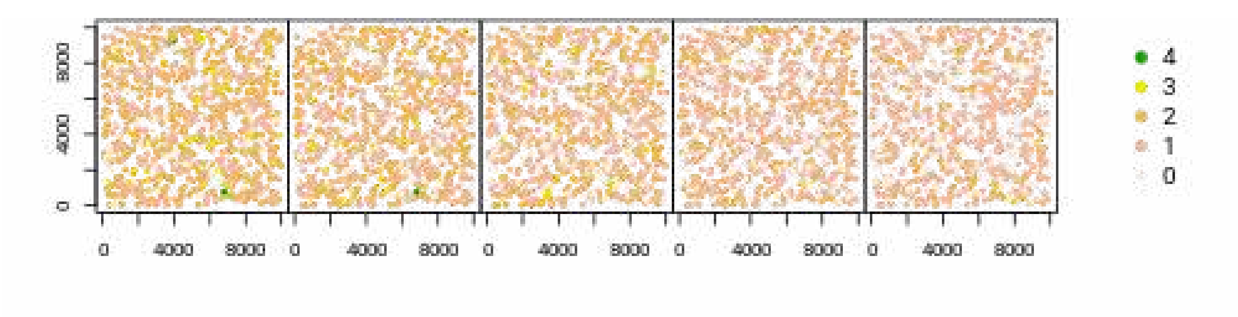
```
par(mfrow = c(1, 6), oma = c(5, 4, 0.1, 0.1), mar = rep(0, 4))

cols <- terrain.colors(5, rev = TRUE)

plot(range(sim.df$easting), range(sim.df$northing), type = 'n',
     xlab = "Easting", ylab = "Northing")
points(sim.df$easting[sim.df$year == 1], sim.df$northing[sim.df$year == 1],
       col = cols[sim.df$y_sim[sim.df$year == 1]], pch = 19, cex = 0.6)

for (yy in 2:5) {
  plot(range(sim.df$easting), range(sim.df$northing), type = 'n',
     xlab = "Easting", ylab = "Northing", axes = FALSE)
  points(sim.df$easting[sim.df$year == yy], sim.df$northing[sim.df$year == yy],
       col = cols[sim.df$y_sim[sim.df$year == yy]], pch = 19, cex = 0.6)
  axis(1)
  box()
}

plot(0:1, 0:1, type = 'n', axes = FALSE, xlab = NA, ylab = NA)
legend("center", bty = "n", col = rev(cols), border = NA, pch = 19,
       legend = 4:0, cex = 1.25)
```



Coerce multivariate ordinal observations into pseudo binary observations.

```r
sim_bin <- dfBinary(sim.df, "y_sim")

# reverse sign of global effect covariates
sim_bin[ , c("year", "log_access", "forest", "ctrl", "d")] <-
  -sim_bin[ , c("year", "log_access", "forest", "ctrl", "d")]
```

Three models are fit to the synthetic data: $M1$ is a GLM, $M2$ includes spatial random effects and $M3$ includes spatio-temporal random effects. The fixed effects are the same for all three models. The first two are misspecified. Although even correctly specified spatial models may not have consistent estimation under infill asymptotics (Zhang, 2004), the 95% central credible intervals for $M3$ include the known parameter values used to generate the synthetic dataset.

6.0.1 M1

```r
# INLA glm
sim_inla <- inla(y_binary ~ - 1 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl + d + year +
                    forest + log_access,
                    data = sim_bin,
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute = list(dic = TRUE)
                 )
summary(sim_inla)

## Time used:
##     Pre = 2.33, Running = 3.11, Post = 0.0487, Total = 5.49
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1      -4.631 0.095     -4.817   -4.631     -4.446 -4.631   0
## cut_2      -1.781 0.061     -1.900   -1.781     -1.662 -1.781   0
## cut_3       0.098 0.060     -0.019    0.098      0.216  0.098   0
## cut_4       0.899 0.188      0.531    0.899      1.267  0.899   0
## ctrl        1.005 0.063      0.882    1.005      1.129  1.005   0
## d          -0.310 0.018     -0.344   -0.310     -0.275 -0.310   0
## year       -0.345 0.016     -0.375   -0.345     -0.315 -0.345   0
## forest      0.527 0.040      0.448    0.527      0.606  0.527   0
## log_access -0.235 0.005     -0.246   -0.235     -0.224 -0.235   0
##
## Deviance Information Criterion (DIC) ...............: 7149.31
## Deviance Information Criterion (DIC, saturated) ....: 7141.14
## Effective number of parameters .....................: 9.00
##
## Marginal log-Likelihood:  -3628.56
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

```
parms
```

```
##      cut_1      cut_2      cut_3      cut_4       ctrl          d       year
##      -5.00      -2.00       0.00       1.00       1.00      -0.33      -0.40
##     forest log_access
##       0.50      -0.25
```

6.0.2 M2

```
# mesh
mesh.sim <- inla.mesh.2d(sites,
    max.edge = 0.05*100000,
    cutoff = 0.01*100000
)
# spatial prior
spde.sim <- inla.spde2.pcmatern(
  # Mesh and smoothness parameter
  mesh = mesh.sim, alpha = 2,
  # P(prior.range < 100 m) = 0.05
  prior.range = c(100, 0.05),
  # P(sigma > 0.25) = 0.05
  prior.sigma = c(0.25, 0.05)
)

# Projection matrix: A
A.sim <- inla.spde.make.A(mesh.sim,
                          loc = as.matrix(sim_bin[, c("easting", "northing")])
                          )

stk.sim <- inla.stack(
  data = list(y_binary = sim_bin$y_binary),
  A = list(A.sim, 1),
  effects = list(
    knot = 1:spde.sim$n.spde,
     data.frame(sim_bin[ , c("cut_1","cut_2","cut_3","cut_4",
       "ctrl", "d", "year", "forest", "log_access")])
     ),
  tag = 'est'
)

sim_inlaS <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl + d + year +
                    + forest + log_access +
                    f(knot, model = spde.sim),
                    data = inla.stack.data(stk.sim),
                    control.predictor = list(A = inla.stack.A(stk.sim)),
```

```r
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute = list(dic = TRUE))

summary(sim_inlaS)

## Time used:
##     Pre = 0.552, Running = 12.5, Post = 0.0866, Total = 13.1
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1      -4.719 0.106     -4.927   -4.719     -4.513 -4.719   0
## cut_2      -1.813 0.074     -1.960   -1.813     -1.668 -1.813   0
## cut_3       0.114 0.074     -0.030    0.114      0.259  0.114   0
## cut_4       1.022 0.206      0.618    1.022      1.428  1.022   0
## ctrl        1.033 0.064      0.907    1.033      1.159  1.033   0
## d          -0.315 0.018     -0.350   -0.315     -0.279 -0.315   0
## year       -0.357 0.016     -0.388   -0.357     -0.326 -0.357   0
## forest      0.556 0.042      0.474    0.556      0.639  0.556   0
## log_access -0.239 0.006     -0.250   -0.239     -0.228 -0.239   0
##
## Random effects:
##   Name   Model
##     knot SPDE2 model
##
## Model hyperparameters:
##                     mean      sd 0.025quant 0.5quant 0.975quant    mode
## Range for knot 1014.870 404.427     429.08  946.809   1994.012 822.141
## Stdev for knot    0.361   0.106       0.20    0.345      0.612   0.315
##
## Deviance Information Criterion (DIC) ...............: 7083.76
## Deviance Information Criterion (DIC, saturated) ....: 7075.59
## Effective number of parameters .....................: 47.06
##
## Marginal log-Likelihood:  -3611.24
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

parms; s.range; s.stdev

##      cut_1      cut_2      cut_3      cut_4       ctrl          d       year
##      -5.00      -2.00       0.00       1.00       1.00      -0.33      -0.40
##     forest log_access
##       0.50      -0.25
## [1] 500
## [1] 0.25

# Verify if the same predicted random effects are shared by the input data at
```

```r
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=sim_inlaS,
                            Aproj=A.sim,
                            DataLocs=as.matrix(sim_bin[, c("easting", "northing")]),
                            spdeName = "knot")

## [1] TRUE
```

6.0.3 M3

```r
# prior for temporal autoregressive parameter
h.sim <- list(rho = list(prior = 'pc.cor1', param = c(0, 0.9)))

# number of years
nyear.sim <- length(unique(sim_bin$year))

# year index
sim_bin$yearIdx <- sim_bin$year - min(sim_bin$year) + 1

# SPDE model index set
indexs.sim <- inla.spde.make.index('s', n.spde = spde.sim$n.spde, n.group = nyear.sim)

# Projection matrix by year
A.sim.st <- inla.spde.make.A(mesh = mesh.sim,
                         loc = as.matrix(sim_bin[, c("easting", "northing")]),
                         group = sim_bin$yearIdx)

# stack
stk.sim.st <- inla.stack(data = list(y_binary = sim_bin$y_binary),
                      A = list(A.sim.st, 1),
                      effects = list(indexs.sim,
                       data.frame(sim_bin[ , c("cut_1","cut_2","cut_3","cut_4",
                                  "ctrl", "d", "year",
                                  "forest", "log_access")])),
                      tag = 'est_st')

sim_inlaST <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                          ctrl + d + year +
                          forest + log_access +
                          f(s, model = spde.sim, group = s.group,
                          control.group = list(model = 'ar1', hyper = h.sim)),
                    data = inla.stack.data(stk.sim.st),
                    control.predictor = list(A = inla.stack.A(stk.sim.st)),
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
```

```
                    control.compute=list(config = TRUE, dic = TRUE)
                   )
summary(sim_inlaST)

## Time used:
##     Pre = 0.514, Running = 16.1, Post = 0.146, Total = 16.7
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1      -4.730 0.111     -4.950   -4.729     -4.513 -4.729   0
## cut_2      -1.811 0.081     -1.971   -1.811     -1.652 -1.811   0
## cut_3       0.123 0.081     -0.034    0.123      0.284  0.123   0
## cut_4       1.032 0.208      0.624    1.032      1.441  1.032   0
## ctrl        1.040 0.065      0.913    1.040      1.168  1.040   0
## d          -0.317 0.018     -0.352   -0.317     -0.281 -0.317   0
## year       -0.357 0.019     -0.395   -0.357     -0.320 -0.357   0
## forest      0.558 0.042      0.476    0.558      0.641  0.558   0
## log_access -0.240 0.006     -0.251   -0.240     -0.229 -0.240   0
##
## Random effects:
##   Name   Model
##     s SPDE2 model
##
## Model hyperparameters:
##                     mean      sd 0.025quant 0.5quant 0.975quant    mode
## Range for s    1162.570 478.412    475.275 1080.392   2324.368 929.799
## Stdev for s       0.363   0.100      0.209    0.349      0.598   0.322
## GroupRho for s    0.900   0.076      0.703    0.920      0.987   0.961
##
## Deviance Information Criterion (DIC) ...............: 7082.33
## Deviance Information Criterion (DIC, saturated) ....: 7074.17
## Effective number of parameters .....................: 63.58
##
## Marginal log-Likelihood:  -3611.51
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

parms; s.range; s.stdev; rho.t

##      cut_1      cut_2      cut_3      cut_4       ctrl          d       year
##      -5.00      -2.00       0.00       1.00       1.00      -0.33      -0.40
##     forest log_access
##       0.50      -0.25
## [1] 500
## [1] 0.25
## [1] 0.9

# Verify if the same predicted random effects are shared by the input data at
```

```
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=sim_inlaST,
                            Aproj=A.sim.st,
                            DataLocs=as.matrix(sim_bin[, c("easting", "northing", "yearIdx")]),
                            spdeName = "s")

## [1] TRUE

# DIC
matrix(
  c(sim_inla$dic$dic, sim_inlaS$dic$dic, sim_inlaST$dic$dic),
  nrow = 3, ncol = 1,
  dimnames = list(
    c("M1", "M2", "M3"),
    c("DIC")
  )
)

##         DIC
## M1 7149.307
## M2 7083.755
## M3 7082.334
```

7 ACT Data Example

The ACT floristics data provide the invasive weed density using the modified Braun-Blanquet abundance/cover system.

```
# for days since control
for (i in 1:ncol(dat)) {
  if (grepl("daySinceControl", names(dat)[i])) {
    # convert NAs to 0
    d <- dat[, i]
    d[is.na(d)] <- 0
    d <- ifelse(d < 0, 0, d)
    # units in years
    d <- d/365
    # binary control level variable
    if (names(dat)[i] == "daySinceControl") {
      dat$d_any <- d
      dat$ctrl_any <- ifelse(d == 0, d, 1)
    } else if (names(dat)[i] == "ALG_daySinceControl") { # ALG vs AGC
      dat$d_AGC <- d
      dat$ctrl_AGC <- ifelse(d == 0, d, 1)
    } else if (names(dat)[i] == "ST_daySinceControl") { # ST vs St
      dat$d_St <- d
```

```
      dat$ctrl_St <- ifelse(d == 0, d, 1)
    } else if (names(dat)[i] == "CNG_daySinceControl") {
      dat$d_CNG <- d
      dat$ctrl_CNG <- ifelse(d == 0, d, 1)
    } else if (names(dat)[i] == "StJW_daySinceControl") {
      dat$d_StJW <- d
      dat$ctrl_StJW <- ifelse(d == 0, d, 1)
    }
  }
}

# years since first year of observation period
dat$year <- dat$year - min(dat$year)

# log transform distance to nearest road, track or trail
# as an overall measure of accessibility
dat$log_access <- log(dat$dist2road_track_trail)
```

```
# Convert the ordinal scores into the binary format
AGC_bin <- dfBinary(dat, "y_AGC")
CNG_bin <- dfBinary(dat, "y_CNG")
St_bin <- dfBinary(dat, "y_St")
StJW_bin <- dfBinary(dat, "y_StJW")

# sign reversal convention for covariates with global effects
AGC_bin[ , c("year", "log_access", "forest", "ctrl_AGC", "d_AGC")] <-
  -AGC_bin[ , c("year", "log_access", "forest", "ctrl_AGC", "d_AGC")]
CNG_bin[ , c("year", "log_access", "forest", "ctrl_CNG", "d_CNG")] <-
  -CNG_bin[ , c("year", "log_access", "forest", "ctrl_CNG", "d_CNG")]
St_bin[ , c("year", "log_access", "forest", "ctrl_St", "d_St")] <-
  -St_bin[ , c("year", "log_access", "forest", "ctrl_St", "d_St")]
StJW_bin[ , c("year", "log_access", "forest", "ctrl_StJW", "d_StJW")] <-
  -StJW_bin[ , c("year", "log_access", "forest", "ctrl_StJW", "d_StJW")]
```

The spatial range κ is defined as the distance where the spatial correlation is about 0.13 (Lindgren and Rue, 2015). A penalised complexity prior (Fuglstad et al., 2019) sets $P(\kappa < 10km) = 0.05$ for a Matern covariance function with fractional order operator $\alpha = 2$. The 10 km bound is about 10% of the maximum distance between observed locations. The spatial standard deviation also uses a penalised complexity prior specified such that $P(\sigma > 0.25) = 0.05$. For spatio-temporal models, a penalised complexity prior for the temporal correlation parameter ρ specifies that $P(\rho > 0.90) = 0.90$.

The choice of mesh is linked to both the locations of the observations and the spatial prior specification. The cutoff that limits the size of the smallest triangles in the mesh is set to 500 metres, which is 1/20 of the 10 km probabilistic bound used in the above penalised complexity prior for the spatial range. The inner maximum edge argument is increased two-fold relative to the cutoff to 1 km and the outer maximum edge argument is increased another ten-fold to 10 km.

```r
library(fmesher)

# unique locations with observations (in kms)
locs <- unique(dat[ , c("easting", "northing")])

# max distance between locations:
max(dist(locs))

## [1] 86.72276

max.edge <- c(1, 10)
cutoff <- 0.5

meshACT <- fmesher::fm_mesh_2d(
  loc = locs,
  boundary = fmesher::fm_extensions(act.mga),
  max.edge = max.edge,
  cutoff = cutoff
)
```

```r
# spatial prior
spde1 <- inla.spde2.pcmatern(
  # Mesh and smoothness parameter
  mesh = meshACT, alpha = 2,
  # P(prior.range < 10 km) = 0.05
  prior.range = c(10, 0.05),
  # P(sigma > 0.25) = 0.05
  prior.sigma = c(0.25, 0.05))

# prior for temporal autoregressive parameter
# P(rho > u) = alpha
# param = c(u, alpha)
h.spec <- list(rho = list(prior = 'pc.cor1', param = c(0.5, 2/3)))
```

The constructed mesh is plotted.

```r
plot(meshACT)
plot(st_geometry(act.mga), add = TRUE, col = NA, border = 'red')
points(locs, pch = 19, col = 'cyan', cex = 0.4)
```

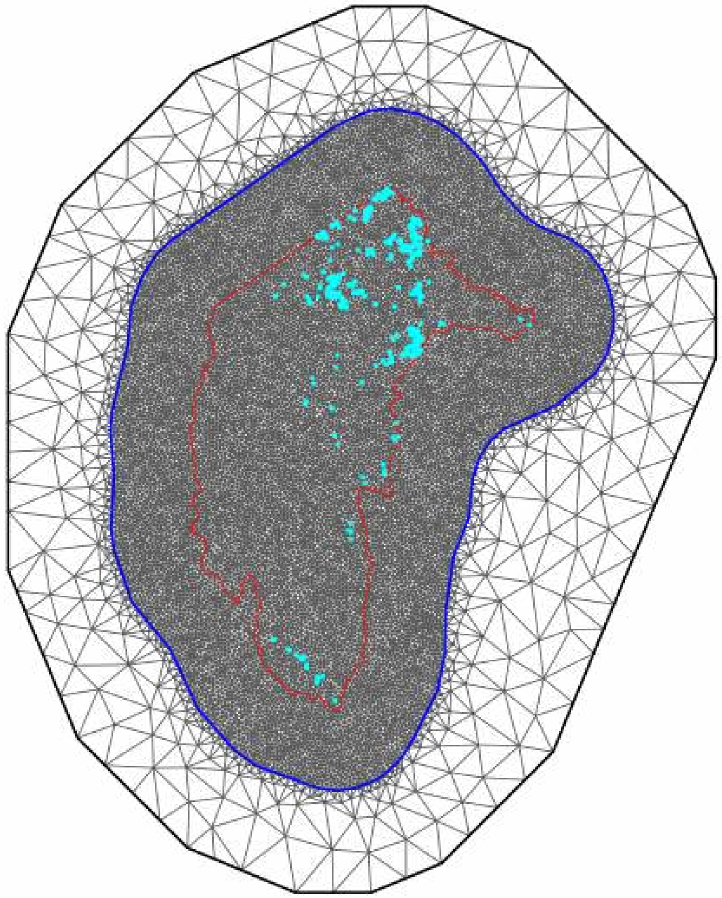


Figure 5: The mesh plot. Based on data provided by the ACT Government (see license).

8 Results

The data are fitted to the three sequential ordinal models using INLA: M1 not accounting for spatial and temporal effects; M2 with spatial effects; and M3 accounting spatial and temporal effects.

The spatial model (M2) is fitted using a mesh constructed using the locations of the data. Note that the created spatial domain cannot cover the whole territory of the ACT.

M2 takes advantage of the SPDE (stochastic partial differential equation) approach to estimate spatial effects, and the estimated effect of a location is shared among the binary samples at the location for the sequential ordinal model. The mesh extent is based on the ACT polygon. The mesh construction uses the **fmesher** package.

M3 uses the mesh as the same as the one used by M2 for an annual population growth model.

8.1 African Love Grass

8.1.1 M1

```r
# INLA glm
AGC_inla <- inla(y_binary ~ - 1 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl_AGC + d_AGC + year +
                    forest + log_access,
                    data = AGC_bin,
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute = list(dic = TRUE))
summary(AGC_inla)

## Time used:
##     Pre = 0.373, Running = 1.21, Post = 0.0231, Total = 1.6
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1       0.644 0.092      0.464    0.644      0.824  0.644   0
## cut_2      -0.681 0.144     -0.964   -0.681     -0.399 -0.681   0
## cut_3      -0.127 0.148     -0.418   -0.127      0.163 -0.127   0
## cut_4       0.164 0.184     -0.197    0.164      0.525  0.164   0
## ctrl_AGC    0.668 0.117      0.438    0.668      0.898  0.668   0
## d_AGC      -0.045 0.029     -0.102   -0.045      0.012 -0.045   0
## year        0.044 0.022      0.001    0.044      0.088  0.044   0
## forest     -0.308 0.074     -0.453   -0.308     -0.163 -0.308   0
## log_access  0.033 0.031     -0.028    0.033      0.095  0.033   0
##
## Deviance Information Criterion (DIC) ...............: 1631.11
## Deviance Information Criterion (DIC, saturated) ....: 1630.41
## Effective number of parameters .....................: 8.97
##
## Marginal log-Likelihood:  -863.09
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

8.1.2 M2

```r
# Projection matrix: A
locs2 <- as.matrix(AGC_bin[, c("easting", "northing")])
A1 <- inla.spde.make.A(meshACT, loc = locs2)

stk1 <- inla.stack(
  data = list(y_binary = AGC_bin$y_binary),
  A = list(A1, 1),
```

```
  effects = list(
    knot = 1:spde1$n.spde,
     data.frame(AGC_bin[ , c("cut_1","cut_2","cut_3","cut_4",
       "ctrl_AGC", "d_AGC","year", "forest", "log_access")])
     ),
  tag = 'est')

AGC_inlaS <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl_AGC + d_AGC + year +
                    + forest + log_access +
                    f(knot, model = spde1),
                    data = inla.stack.data(stk1),
                    control.predictor = list(A = inla.stack.A(stk1)),
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute = list(dic = TRUE))

summary(AGC_inlaS)

## Time used:
##     Pre = 0.538, Running = 116, Post = 0.961, Total = 118
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1        1.181 0.312      0.587    1.175      1.815  1.175   0
## cut_2        0.626 0.346     -0.033    0.619      1.328  0.619   0
## cut_3        1.516 0.361      0.828    1.508      2.250  1.508   0
## cut_4        2.205 0.400      1.442    2.196      3.017  2.197   0
## ctrl_AGC    -0.019 0.178     -0.369   -0.019      0.330 -0.019   0
## d_AGC        0.059 0.044     -0.027    0.059      0.145  0.059   0
## year         0.214 0.038      0.140    0.213      0.289  0.213   0
## forest      -0.280 0.144     -0.563   -0.279      0.001 -0.279   0
## log_access -0.006 0.065     -0.133   -0.006      0.121 -0.006   0
##
## Random effects:
##   Name   Model
##     knot SPDE2 model
##
## Model hyperparameters:
##                 mean    sd 0.025quant 0.5quant 0.975quant mode
## Range for knot 3.86 0.701       2.68     3.79       5.43 3.65
## Stdev for knot 1.42 0.182       1.10     1.41       1.81 1.39
##
## Deviance Information Criterion (DIC) ...............: 1258.94
## Deviance Information Criterion (DIC, saturated) ....: 1258.24
## Effective number of parameters .....................: 116.71
##
## Marginal log-Likelihood:  -730.64
##  is computed
```

```
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

```r
AGC_inlaS$summary.fixed
```

```
##                     mean         sd  0.025quant     0.5quant  0.975quant
## cut_1        1.181443604 0.31194850  0.58691825  1.174655811 1.815260352
## cut_2        0.626234043 0.34594304 -0.03268453  0.618884791 1.328110474
## cut_3        1.515517570 0.36148193  0.82817038  1.507526813 2.250086921
## cut_4        2.204521052 0.40021120  1.44203622  2.196289878 3.016701269
## ctrl_AGC    -0.018762305 0.17804567 -0.36853111 -0.018566796 0.329900356
## d_AGC        0.058721448 0.04387243 -0.02706982  0.058628954 0.145034835
## year         0.213788979 0.03789925  0.14044483  0.213437404 0.289167234
## forest      -0.279606648 0.14377817 -0.56301970 -0.279156358 0.001238552
## log_access -0.006117345 0.06474931 -0.13269037 -0.006296153 0.121461770
##                     mode          kld
## cut_1        1.174923889 8.486640e-08
## cut_2        0.619212253 9.614969e-08
## cut_3        1.507974132 1.088262e-07
## cut_4        2.196967088 1.053902e-07
## ctrl_AGC    -0.018564215 3.243166e-10
## d_AGC        0.058627379 1.126867e-09
## year         0.213444638 2.284639e-08
## forest      -0.279157641 3.180138e-09
## log_access -0.006299343 2.970495e-09
```

```r
# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=AGC_inlaS,
                            Aproj=A1,
                            DataLocs=as.matrix(AGC_bin[, c("easting", "northing")]),
                            spdeName = "knot")
```

```
## [1] TRUE
```

8.1.3 M3

```r
# number of years
nyear <- length(unique(AGC_bin$year))

# year index
AGC_bin$yearIdx <- AGC_bin$year - min(AGC_bin$year) + 1

# SPDE model index set
indexs <- inla.spde.make.index('s', n.spde = spde1$n.spde, n.group = nyear)
```

```r
# Projection matrix by year
A_st <- inla.spde.make.A(mesh = meshACT, loc = locs2,
                         group = AGC_bin$yearIdx)

# stack
stkc_st <- inla.stack(data = list(y_binary = AGC_bin$y_binary),
                      A = list(A_st, 1),
                      effects = list(indexs,
                       data.frame(AGC_bin[ , c("cut_1","cut_2","cut_3","cut_4",
                                      "ctrl_AGC", "d_AGC", "year",
                                      "forest", "log_access")])),
                      tag = 'est_st')

#(INLA::inla.getOption("inla.mode"))
AGC_inlaST <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                           ctrl_AGC + d_AGC  + year +
                           forest + log_access +
                           f(s, model = spde1, group = s.group,
                           control.group = list(model = 'ar1', hyper = h.spec)),
                    data = inla.stack.data(stkc_st),
                    control.predictor = list(A = inla.stack.A(stkc_st)),
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute=list(config = TRUE, # Store internal GMRF approximations (
                                        dic = TRUE)
                   )
summary(AGC_inlaST)

## Time used:
##     Pre = 0.659, Running = 1805, Post = 8.72, Total = 1814
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1        1.174 0.315      0.576    1.167      1.816  1.167   0
## cut_2        0.626 0.350     -0.039    0.617      1.339  0.617   0
## cut_3        1.521 0.367      0.827    1.511      2.271  1.511   0
## cut_4        2.219 0.409      1.447    2.208      3.053  2.207   0
## ctrl_AGC   -0.031 0.182     -0.389   -0.030      0.325 -0.030   0
## d_AGC        0.063 0.045     -0.025    0.063      0.151  0.063   0
## year         0.212 0.044      0.126    0.211      0.301  0.211   0
## forest     -0.282 0.144     -0.566   -0.282     -0.002 -0.282   0
## log_access -0.006 0.065     -0.132   -0.006      0.122 -0.006   0
##
## Random effects:
##   Name   Model
##     s SPDE2 model
##
## Model hyperparameters:
```

```
##                  mean    sd 0.025quant 0.5quant 0.975quant mode
## Range for s    3.839 0.694      2.664    3.774       5.39 3.64
## Stdev for s    1.419 0.181      1.094    1.408       1.81 1.39
## GroupRho for s 0.996 0.006      0.981    0.998       1.00 1.00
##
## Deviance Information Criterion (DIC) ...............: 1262.01
## Deviance Information Criterion (DIC, saturated) ....: 1261.31
## Effective number of parameters .....................: 121.83
##
## Marginal log-Likelihood:  -733.12
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

```
# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=AGC_inlaST,
                            Aproj=A_st,
                            DataLocs=as.matrix(AGC_bin[, c("easting", "northing", "yearIdx")]),
                            spdeName = "s")
```

```
## [1] TRUE
```

```
# output summary table
out.df <- data.frame(
  cbind(
    c("\\multirow{12}{*}{African love grass}", rep("", 11)),
    gsub("_", "\\\\_", c(row.names(AGC_inlaST$summary.fixed),
      row.names(AGC_inlaST$summary.hyperpar)
    )),
    rbind(
      AGC_inlaST$summary.fixed[ , c("0.025quant", "0.5quant", "0.975quant")],
      AGC_inlaST$summary.hyperpar[ , c("0.025quant", "0.5quant", "0.975quant")]
    )
  )
)
names(out.df) <- c("Species", "Parameter", "q0.025", "q0.50", "q0.975")
print(xtable::xtable(out.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "AGCtab.txt")

# DIC
dic.df <- data.frame(
  cbind(
    c("\\multirow{3}{*}{African lovegrass}", rep("", 2)),
    c("M1", "M2", "M3"),
```

```r
    round(c(AGC_inla$dic$dic, AGC_inlaS$dic$dic, AGC_inlaST$dic$dic))
  )
)
names(dic.df) <- c("Species", "Model", "DIC")
print(xtable::xtable(dic.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "AGCdic.txt")
```

8.1.4 Prediction for M3

Use the INLA MC sampler to predict posterior marginal probabilities of the aggregated modified Braun-Blanquet score by year.

```r
# Data for prediction
## Raster data
# get the raster cells (or indices) and locs for prediction
aa <- terra::values(covars$ForestNGrass)
table(aa, useNA="ifany") # forest=1, Grassland=2, others=NA

## aa
##      1      2      3    NaN
## 161477  44917  29912 277900

# Forest and Grassland
FnGcells <- which(!is.na(aa[,1]) & (aa[,1] == 1 | aa[,1] == 2))
FnG <- aa[FnGcells,1]
FnG[FnG==2] <- 0 # Index grassland as 0

# locs
locs4pred <- terra::xyFromCell(covars, FnGcells)

# log(dist2road_track_trail)
log_accessTmp <- log(terra::values(covars$dist2road_track_trail)[FnGcells,])
summary(log_accessTmp) # Six NAs

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -5.25214 -2.33379 -1.05818 -1.25080 -0.08805  1.40968        6

# spatio-temporal control effort
ALGcontrol.t <- system.file("extdata/ALG_DaysSinceControl.tif", package="DSTSOM")
ALGcontrol.r <- terra::rast(ALGcontrol.t)/365 # duration since last control in years
# metres to kms
ALGcontrol.r <- project(ALGcontrol.r, gsub("units=m", "units=km",
                               crs(ALGcontrol.r, proj = TRUE)))
ALGcontrol <- terra::values(ALGcontrol.r)[FnGcells, ]
summary(ALGcontrol) # 104 NAs in each year
```

```
##  DaySinceCon_Ref20190701 DaySinceCon_Ref20200701 DaySinceCon_Ref20210701
##  Min.   :0.0000          Min.   :0.00000         Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.00000         1st Qu.:0.0000
##  Median :0.0000          Median :0.00000         Median :0.0000
##  Mean   :0.0671          Mean   :0.08893         Mean   :0.1085
##  3rd Qu.:0.0000          3rd Qu.:0.00000         3rd Qu.:0.0000
##  Max.   :4.3468          Max.   :5.33316         Max.   :6.3288
##  NA's   :104             NA's   :104             NA's   :104
##  DaySinceCon_Ref20220701 DaySinceCon_Ref20230701
##  Min.   :0.0000          Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.0000
##  Median :0.0000          Median :0.0000
##  Mean   :0.1342          Mean   :0.1648
##  3rd Qu.:0.0000          3rd Qu.:0.0000
##  Max.   :7.3288          Max.   :8.3288
##  NA's   :104             NA's   :104
```

```
# data.frame of spatial data for prediction
data4pred01 <- data.frame(easting = locs4pred[,"x"],
                          northing = locs4pred[,"y"],
                          forest = FnG,
                          log_access = log_accessTmp,
                          PredCell = FnGcells
                          )

summary(data4pred01)
```

```
##     easting         northing        forest          log_access
##  Min.   :659.9   Min.   :6023   Min.   :0.0000   Min.   :-5.25214
##  1st Qu.:672.2   1st Qu.:6052   1st Qu.:1.0000   1st Qu.:-2.33379
##  Median :679.5   Median :6069   Median :1.0000   Median :-1.05818
##  Mean   :680.5   Mean   :6068   Mean   :0.7824   Mean   :-1.25080
##  3rd Qu.:686.7   3rd Qu.:6086   3rd Qu.:1.0000   3rd Qu.:-0.08805
##  Max.   :718.1   Max.   :6111   Max.   :1.0000   Max.   : 1.40968
##                                                  NA's   :6
##     PredCell
##  Min.   :   335
##  1st Qu.:146106
##  Median :247320
##  Mean   :251202
##  3rd Qu.:346395
##  Max.   :513874
##
```

```
## take out NAs
allNAs <- unique(c(which(is.na(data4pred01$log_access)),
                  which(is.na(ALGcontrol[ , 1]))))
data4pred01 <- data4pred01[-allNAs, ]
```

```r
ALGcontrol <- ALGcontrol[-allNAs, ]

stAGC.ls <- list(
  ctrl_AGC = ifelse(ALGcontrol > 0, 1, 0),
  d_AGC = ALGcontrol
)

rm(aa, FnG, locs4pred, log_accessTmp, ALGcontrol)
invisible(gc())

# the projection matrix of the pred locs (spatial only because the locs are the same)
A_s_Pred01 <- inla.spde.make.A(mesh = meshACT,
                     loc = as.matrix(data4pred01[,c("easting", "northing")]))

# form spatio-temporal array
AGCarr <- array(NA, dim = c(nrow(data4pred01), 5, 5),
                dimnames = list(
                  sites = 1:nrow(data4pred01),
                  predictors = c("ctrl_AGC", "d_AGC", "year", "forest", "log_access"),
                  year = 1:5
                ))

# spatio-temporal covariates
AGCarr[ , "ctrl_AGC", ] <- stAGC.ls[["ctrl_AGC"]]
AGCarr[ , "d_AGC", ] <- stAGC.ls[["d_AGC"]]

# spatial covariates
for (i in 1:5) {
  AGCarr[ , "forest", i] <- data4pred01[ , "forest"]
  AGCarr[ , "log_access", i] <- data4pred01[ , "log_access"]
}

# temporal covariates
for (i in 1:5) {
  AGCarr[ , "year", i] <- i - 0.5
}

# sign reversal convention for fixed effects
AGCarr <- -AGCarr

# Prediction
pt <- proc.time()
fm_pred <- predict_DSTSOM(
                             fm_inla = AGC_inlaST,
                             nSample = 10000,
                             arrPred = AGCarr,
                             A_s_Pred = A_s_Pred01
```

```
                             )
proc.time() - pt

##     user   system  elapsed 
## 3052.165 1321.563 4563.176

str(fm_pred)

## List of 2
##  $ MCsummary: num [1:206289, 1:5, 1:3, 1:5] 0.844 0.834 0.927 0.93 0.933 ...
##   ..- attr(*, "dimnames")=List of 4
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "cut_1" "cut_2" "cut_3" "cut_4" ...
##   .. ..$ : chr [1:3] "quant05" "median" "quant95"
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
##  $ maxDev   : num [1:206289, 1:5] 2.22e-16 2.22e-16 2.22e-16 2.22e-16 2.22e-16 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
```

```
rasterVis::levelplot(ALGcontrol.r,
                     names.att = paste0("Year ", 1:5), layout = c(5, 1),
                     par.settings =
                       latticeExtra::custom.theme(
                         region = c("white", RColorBrewer::brewer.pal(9, "Oranges")[2:9])
                       ),
                     panel = function(...) {
                       panel.levelplot(...)
                       sp::sp.polygons(as(sf::st_geometry(act.mga), "Spatial"))
                     }
                     )
```

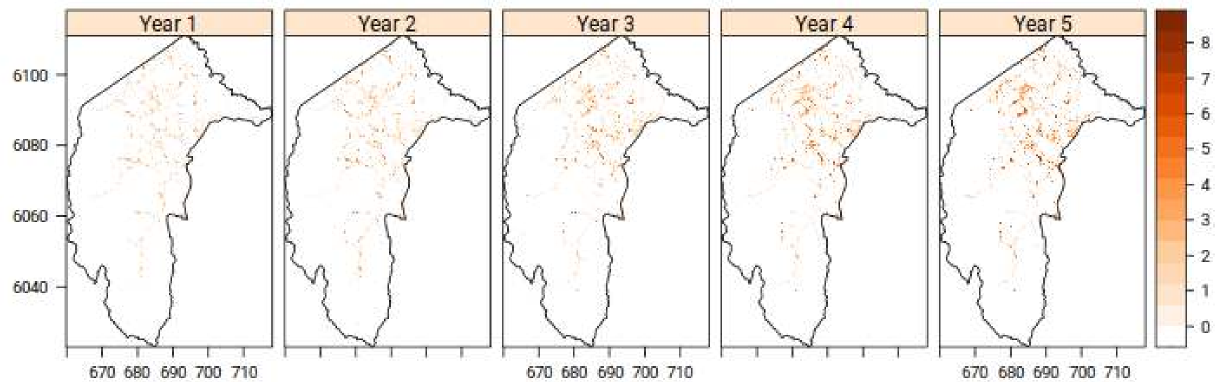


Figure 6: Map of duration since last control effort (colour key is in units of years) on 1 July of each year for African love grass. Based on data provided by the ACT Government (see license).

```
# predictive posterior marginal probabilities by modified BB score and year
# create raster
tempRast <- rast(covars,
                 nlyrs = 5*5*3, # 5 BB levels by 5 years for median, q0.05 and q0.95
                 names = c(paste0(paste0("Y_", 1:5, "_Bmedian_", rep(0:4, each = 5))),
                           paste0(paste0("Y_", 1:5, "_Bq05_", rep(0:4, each = 5))),
                           paste0(paste0("Y_", 1:5, "_Bq95_", rep(0:4, each = 5)))
                 )
)
```

```r
# modified BB scores
for (bb in 0:4) {
  # years
  for (yy in 1:5) {
    tempRast[[paste0(paste0("Y_", yy), "_Bmedian_", bb)]][data4pred01$PredCell] <-
      fm_pred$MCsummary[ , bb + 1, "median", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq05_", bb)]][data4pred01$PredCell] <-
      fm_pred$MCsummary[ , bb + 1, "quant05", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq95_", bb)]][data4pred01$PredCell] <-
      fm_pred$MCsummary[ , bb + 1, "quant95", yy]
  }
}
```

```r
ats <- seq(from = 0, to = 1, length.out = 101)
cols <- terrain.colors(100, rev = TRUE)

# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bmedian", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

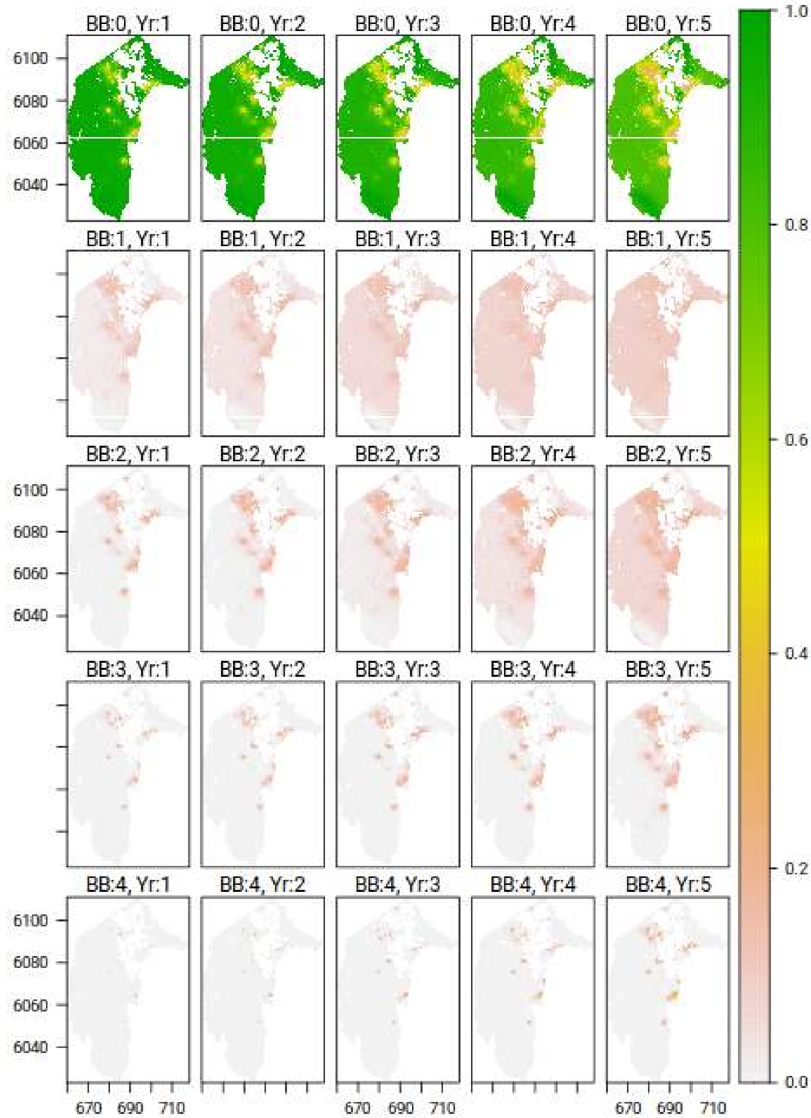


Figure 7: Map of the posterior median probability of the aggregated modified Braun-Blanquet scores by year for African love grass.

```
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq05", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

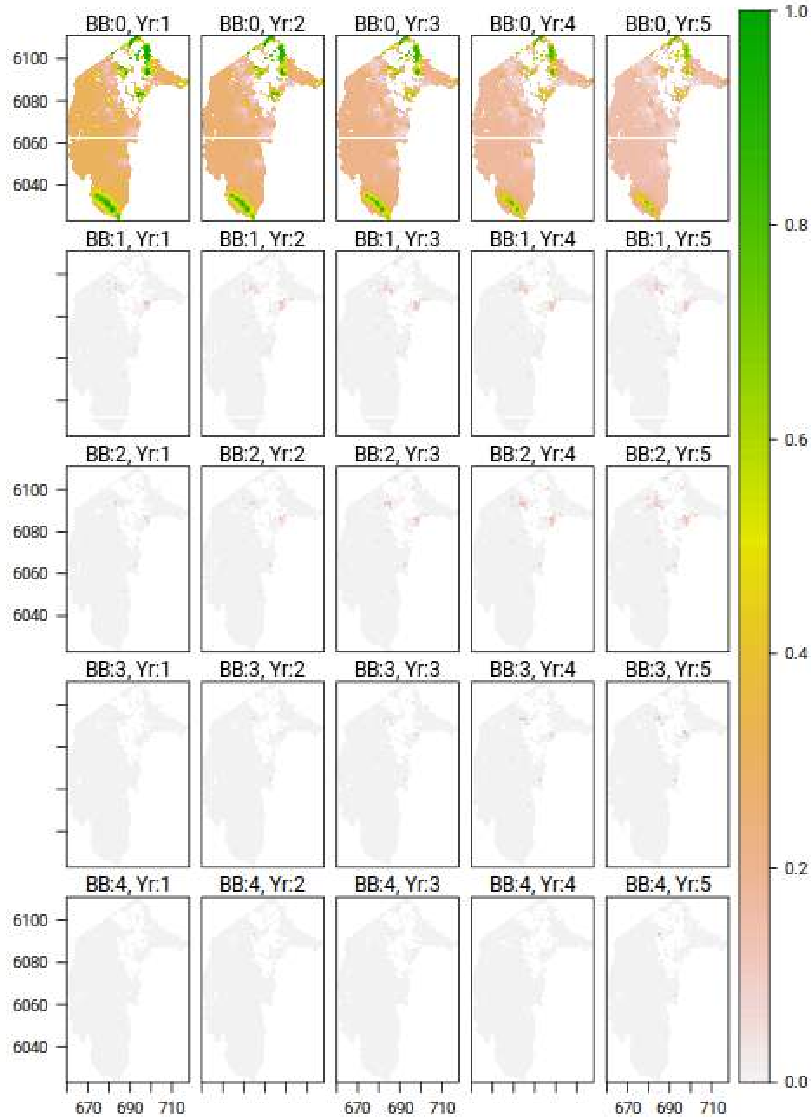


Figure 8: Map of the posterior 0.05 quantile of the aggregated modified Braun-Blanquet scores for African love grass. Based on data provided by the ACT Government (see license).

```
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq95", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

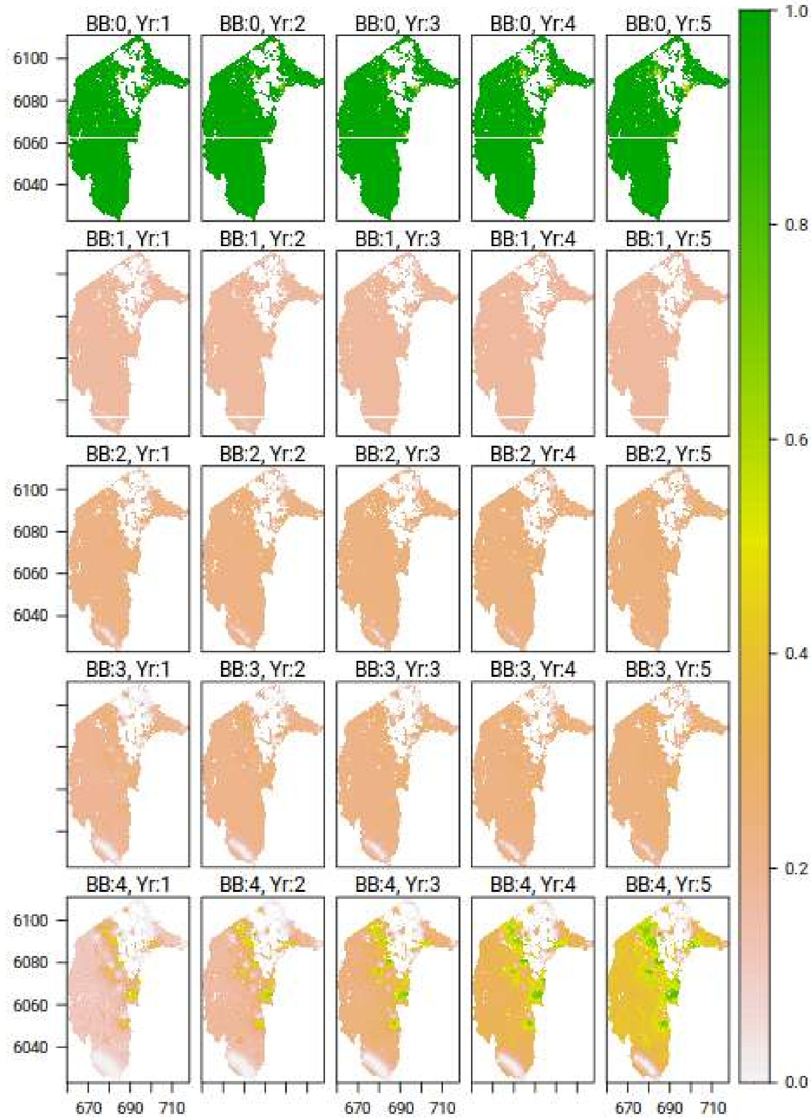


Figure 9: Map of the posterior 0.95 quantile of the aggregated modified Braun-Blanquet scores for African love grass. Based on data provided by the ACT Government (see license).

8.2 Chilean Needle Grass

8.2.1 M1

```
# INLA glm
CNG_inla <- inla(y_binary ~ - 1 + cut_1 + cut_2 + cut_3 + cut_4 +
                   ctrl_CNG + d_CNG + year + forest + log_access,
                   data = CNG_bin,
                   family = "binomial", Ntrials = 1,
                   control.family = list(link = "cloglog"),
```

```
                    control.compute = list(dic = TRUE))
summary(CNG_inla)

## Time used:
##     Pre = 0.432, Running = 1.38, Post = 0.0247, Total = 1.84
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1        0.740 0.093      0.558    0.740      0.922  0.740   0
## cut_2       -1.298 0.187     -1.665   -1.298     -0.931 -1.298   0
## cut_3       -1.178 0.200     -1.570   -1.178     -0.786 -1.178   0
## cut_4        0.313 0.159      0.001    0.313      0.625  0.313   0
## ctrl_CNG     0.765 0.253      0.269    0.765      1.262  0.765   0
## d_CNG       -0.042 0.049     -0.138   -0.042      0.054 -0.042   0
## year         0.029 0.022     -0.015    0.029      0.073  0.029   0
## forest      -0.328 0.076     -0.477   -0.328     -0.179 -0.328   0
## log_access   0.004 0.035     -0.065    0.004      0.072  0.004   0
##
## Deviance Information Criterion (DIC) ...............: 1348.01
## Deviance Information Criterion (DIC, saturated) ....: 1347.37
## Effective number of parameters .....................: 8.96
##
## Marginal log-Likelihood:  -719.93
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

8.2.2 M2

```
# Projection matrix: A
locs2 <- as.matrix(CNG_bin[, c("easting", "northing")])
A1 <- inla.spde.make.A(meshACT, loc = locs2)

stk1 <- inla.stack(
  data = list(y_binary = CNG_bin$y_binary),
  A = list(A1, 1),
  effects = list(
    knot = 1:spde1$n.spde,
     data.frame(CNG_bin[ , c("cut_1","cut_2","cut_3","cut_4",
    "ctrl_CNG", "d_CNG",  "year", "forest", "log_access")])
    ),
  tag = 'est')

CNG_inlaS <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl_CNG + d_CNG + year +
                    forest + log_access +
                    f(knot, model = spde1),
```

```r
                    data = inla.stack.data(stk1),
                    control.predictor = list(A = inla.stack.A(stk1)),
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute = list(dic = TRUE))

summary(CNG_inlaS)

## Time used:
##     Pre = 0.646, Running = 110, Post = 0.52, Total = 111
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1       1.854 0.350      1.222    1.834      2.605  1.783   0
## cut_2       0.671 0.408     -0.082    0.654      1.529  0.656   0
## cut_3       1.030 0.424      0.244    1.013      1.920  1.015   0
## cut_4       2.993 0.435      2.190    2.975      3.914  2.978   0
## ctrl_CNG    0.409 0.346     -0.271    0.410      1.086  0.410   0
## d_CNG      -0.034 0.062     -0.154   -0.034      0.087 -0.034   0
## year        0.064 0.034     -0.003    0.064      0.131  0.064   0
## forest     -0.170 0.144     -0.456   -0.169      0.111 -0.169   0
## log_access -0.091 0.066     -0.224   -0.090      0.034 -0.090   0
##
## Random effects:
##   Name   Model
##     knot SPDE2 model
##
## Model hyperparameters:
##                 mean    sd 0.025quant 0.5quant 0.975quant mode
## Range for knot 4.65 1.056      2.948     4.52       7.08 4.27
## Stdev for knot 1.17 0.177      0.864     1.16       1.56 1.14
##
## Deviance Information Criterion (DIC) ...............: 955.63
## Deviance Information Criterion (DIC, saturated) ....: 954.99
## Effective number of parameters .....................: 81.21
##
## Marginal log-Likelihood:  -559.79
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

CNG_inlaS$summary.fixed

##                   mean         sd   0.025quant    0.5quant 0.975quant
## cut_1       1.85362891 0.34972674  1.222167964  1.83410963 2.60473910
## cut_2       0.67100757 0.40788833 -0.082368265  0.65411511 1.52891620
## cut_3       1.02980832 0.42438274  0.243526953  1.01306724 1.92027872
## cut_4       2.99342198 0.43521111  2.190067607  2.97519820 3.91402828
## ctrl_CNG    0.40931032 0.34585569 -0.270609024  0.40987181 1.08603212
```

```
## d_CNG      -0.03352076 0.06150183 -0.154125626 -0.03352475 0.08710607
## year        0.06365119 0.03392832 -0.002532557  0.06352449 0.13055681
## forest     -0.17012225 0.14440032 -0.456332602 -0.16915308 0.11053238
## log_access -0.09113806 0.06568084 -0.224337728 -0.08968906 0.03369427
##                   mode          kld
## cut_1       1.78324036 2.506570e-06
## cut_2       0.65615533 5.633927e-07
## cut_3       1.01521927 4.398601e-07
## cut_4       2.97830740 5.214738e-07
## ctrl_CNG    0.40987357 6.799058e-10
## d_CNG      -0.03352499 1.178140e-12
## year        0.06352420 3.578584e-09
## forest     -0.16916541 1.284343e-08
## log_access -0.08971985 1.150100e-07

# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=CNG_inlaS,
                            Aproj=A1,
                            DataLocs=as.matrix(CNG_bin[, c("easting", "northing")]),
                            spdeName = "knot")

## [1] TRUE
```

8.2.3 M3

```
# number of years
nyear <- length(unique(CNG_bin$year))

# year index
CNG_bin$yearIdx <- CNG_bin$year - min(CNG_bin$year) + 1

# SPDE model index set
indexs <- inla.spde.make.index('s', n.spde = spde1$n.spde, n.group = nyear)

# Projection matrix by year
A_st <- inla.spde.make.A(mesh = meshACT, loc = locs2,
                         group = CNG_bin$yearIdx)

# stack
stkc_st <- inla.stack(data = list(y_binary = CNG_bin$y_binary),
                      A = list(A_st, 1),
                      effects = list(indexs,
                        data.frame(
                             CNG_bin[ , c("cut_1","cut_2","cut_3","cut_4",
```

```
                                       "ctrl_CNG", "d_CNG",  "year",
                                       "forest", "log_access")]
                             )
                           ),
                      tag = 'est_st')

(INLA::inla.getOption("inla.mode"))

## [1] "compact"

CNG_inlaST <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                           ctrl_CNG + d_CNG + year +
                           forest + log_access +
                           f(s, model = spde1, group = s.group,
                           control.group = list(model = 'ar1', hyper = h.spec)),
                    data = inla.stack.data(stkc_st),
                    control.predictor = list(A = inla.stack.A(stkc_st)),
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute=list(config = TRUE, # Store internal GMRF approximations (
                                         dic = TRUE)
                  )

summary(CNG_inlaST)

## Time used:
##     Pre = 0.572, Running = 1734, Post = 6.88, Total = 1741
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1        1.886 0.367      1.226    1.863      2.670  1.807   0
## cut_2        0.734 0.429     -0.052    0.712      1.635  0.646   0
## cut_3        1.105 0.449      0.284    1.084      2.044  1.014   0
## cut_4        3.071 0.463      2.232    3.046      4.044  2.974   0
## ctrl_CNG     0.380 0.358     -0.327    0.381      1.080  0.381   0
## d_CNG       -0.024 0.065     -0.152   -0.024      0.104 -0.024   0
## year         0.064 0.049     -0.034    0.063      0.164  0.063   0
## forest      -0.176 0.147     -0.468   -0.175      0.109 -0.175   0
## log_access -0.099 0.067     -0.237   -0.098      0.028 -0.098   0
##
## Random effects:
##   Name   Model
##     s SPDE2 model
##
## Model hyperparameters:
##                  mean    sd 0.025quant 0.5quant 0.975quant  mode
## Range for s    4.684 1.054      2.982     4.56      7.110 4.309
## Stdev for s    1.201 0.183      0.879     1.19      1.597 1.163
## GroupRho for s 0.985 0.015      0.944     0.99      0.999 0.998
```

```
##
## Deviance Information Criterion (DIC) ...............: 958.78
## Deviance Information Criterion (DIC, saturated) ....: 958.14
## Effective number of parameters .....................: 92.05
##
## Marginal log-Likelihood:  -561.59
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=CNG_inlaST,
                            Aproj=A_st,
                            DataLocs=as.matrix(CNG_bin[, c("easting", "northing", "yearIdx")]),
                            spdeName = "s")

## [1] TRUE

# output summary table
out.df <- data.frame(
  cbind(
    c("\\multirow{12}{*}{Chilean needle grass}", rep("", 11)),
    gsub("_", "\\\\_", c(row.names(CNG_inlaST$summary.fixed),
      row.names(CNG_inlaST$summary.hyperpar)
    )),
    rbind(
      CNG_inlaST$summary.fixed[ , c("0.025quant", "0.5quant", "0.975quant")],
      CNG_inlaST$summary.hyperpar[ , c("0.025quant", "0.5quant", "0.975quant")]
    )
  )
)
names(out.df) <- c("Species", "Parameter", "q0.025", "q0.50", "q0.975")
print(xtable::xtable(out.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "CNGtab.txt")

# DIC
dic.df <- data.frame(
  cbind(
    c("\\multirow{3}{*}{Chilean needle grass}", rep("", 2)),
    c("M1", "M2", "M3"),
    round(c(CNG_inla$dic$dic, CNG_inlaS$dic$dic, CNG_inlaST$dic$dic))
  )
)
names(dic.df) <- c("Species", "Model", "DIC")
```

```r
print(xtable::xtable(dic.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "CNGdic.txt")
```

8.2.4 Prediction for M3

Use the INLA MC sampler to predict posterior marginal probabilities of the aggregated modified Braun-Blanquet score by year.

```r
# spatio-temporal control effort
CNGcontrol.t <- system.file("extdata/CNG_DaysSinceControl.tif", package="DSTSOM")
CNGcontrol.r <- terra::rast(CNGcontrol.t)/365 # duration since last control in years
# metres to kms
CNGcontrol.r <- project(CNGcontrol.r, gsub("units=m", "units=km",
                              crs(CNGcontrol.r, proj = TRUE)))
CNGcontrol <- terra::values(CNGcontrol.r)[FnGcells, ]
summary(CNGcontrol) # 104 NAs in each year

##  DaySinceCon_Ref20190701 DaySinceCon_Ref20200701 DaySinceCon_Ref20210701
##  Min.   :0.00000         Min.   :0.00000         Min.   :0.0000
##  1st Qu.:0.00000         1st Qu.:0.00000         1st Qu.:0.0000
##  Median :0.00000         Median :0.00000         Median :0.0000
##  Mean   :0.01751         Mean   :0.02157         Mean   :0.0265
##  3rd Qu.:0.00000         3rd Qu.:0.00000         3rd Qu.:0.0000
##  Max.   :4.47671         Max.   :5.47945         Max.   :6.4794
##  NA's   :104             NA's   :104             NA's   :104
##  DaySinceCon_Ref20220701 DaySinceCon_Ref20230701
##  Min.   :0.00000         Min.   :0.0000
##  1st Qu.:0.00000         1st Qu.:0.0000
##  Median :0.00000         Median :0.0000
##  Mean   :0.03265         Mean   :0.0382
##  3rd Qu.:0.00000         3rd Qu.:0.0000
##  Max.   :7.47945         Max.   :8.4794
##  NA's   :104             NA's   :104

CNGcontrol <- CNGcontrol[-allNAs, ]
stCNG.ls <- list(
  ctrl_CNG = ifelse(CNGcontrol > 0, 1, 0),
  d_CNG = CNGcontrol
)
rm(CNGcontrol)

# form spatio-temporal array
CNGarr <- array(NA, dim = c(nrow(data4pred01), 5, 5),
                dimnames = list(
                  sites = 1:nrow(data4pred01),
```

```r
                  predictors = c("ctrl_CNG", "d_CNG", "year", "forest", "log_access"),
                  year = 1:5
                ))

# spatio-temporal covariates
CNGarr[ , "ctrl_CNG", ] <- stCNG.ls[["ctrl_CNG"]]
CNGarr[ , "d_CNG", ] <- stCNG.ls[["d_CNG"]]

# spatial covariates
for (i in 1:5) {
  CNGarr[ , "forest", i] <- data4pred01[ , "forest"]
  CNGarr[ , "log_access", i] <- data4pred01[ , "log_access"]
}

# temporal covariates
for (i in 1:5) {
  CNGarr[ , "year", i] <- i - 0.5
}

# sign reversal convention for fixed effects
CNGarr <- -CNGarr

# Prediction
pt <- proc.time()
cng_pred <- predict_DSTSOM(
                             fm_inla = CNG_inlaST,
                             nSample = 10000,
                             arrPred = CNGarr,
                             A_s_Pred = A_s_Pred01
                             )
proc.time() - pt

##     user   system  elapsed
## 3144.772 1874.541 5757.169

str(cng_pred)

## List of 2
##  $ MCsummary: num [1:206289, 1:5, 1:3, 1:5] 0.811 0.79 0.834 0.827 0.846 ...
##   ..- attr(*, "dimnames")=List of 4
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "cut_1" "cut_2" "cut_3" "cut_4" ...
##   .. ..$ : chr [1:3] "quant05" "median" "quant95"
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
##  $ maxDev   : num [1:206289, 1:5] 2.22e-16 2.22e-16 2.22e-16 2.22e-16 2.22e-16 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
```

```r
rasterVis::levelplot(CNGcontrol.r,
                     names.att = paste0("Year ", 1:5), layout = c(5, 1),
                     par.settings =
                       latticeExtra::custom.theme(
                         region = c("white", RColorBrewer::brewer.pal(9, "Oranges")[2:9])
                       ),
                     panel = function(...) {
                       panel.levelplot(...)
                       sp::sp.polygons(as(sf::st_geometry(act.mga), "Spatial"))
                     }
                     )
```

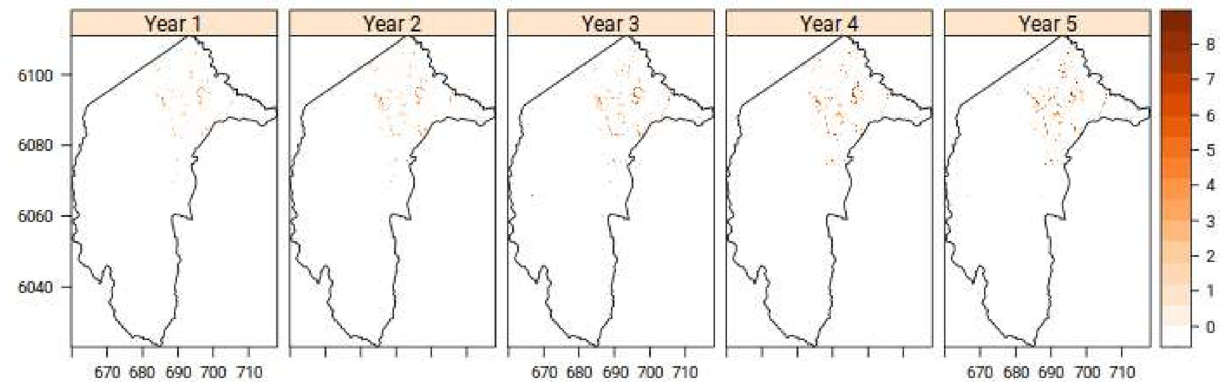


Figure 10: Map of duration since last control effort (colour key is in units of years) on 1 July of each year for Chilean needle grass. Based on data provided by the ACT Government (see license).

```
# predictive posterior marginal probabilities by modified BB score and year
# create raster
tempRast <- rast(covars,
                  nlyrs = 5*5*3, # 5 BB levels by 5 years for median, q0.05 and q0.95
                  names = c(paste0(paste0("Y_", 1:5, "_Bmedian_", rep(0:4, each = 5))),
                            paste0(paste0("Y_", 1:5, "_Bq05_", rep(0:4, each = 5))),
                            paste0(paste0("Y_", 1:5, "_Bq95_", rep(0:4, each = 5)))
                  )
)
```

```r
# modified BB scores
for (bb in 0:4) {
  # years
  for (yy in 1:5) {
    tempRast[[paste0(paste0("Y_", yy), "_Bmedian_", bb)]][data4pred01$PredCell] <-
      cng_pred$MCsummary[ , bb + 1, "median", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq05_", bb)]][data4pred01$PredCell] <-
      cng_pred$MCsummary[ , bb + 1, "quant05", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq95_", bb)]][data4pred01$PredCell] <-
      cng_pred$MCsummary[ , bb + 1, "quant95", yy]
  }
}
```

```r
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bmedian", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

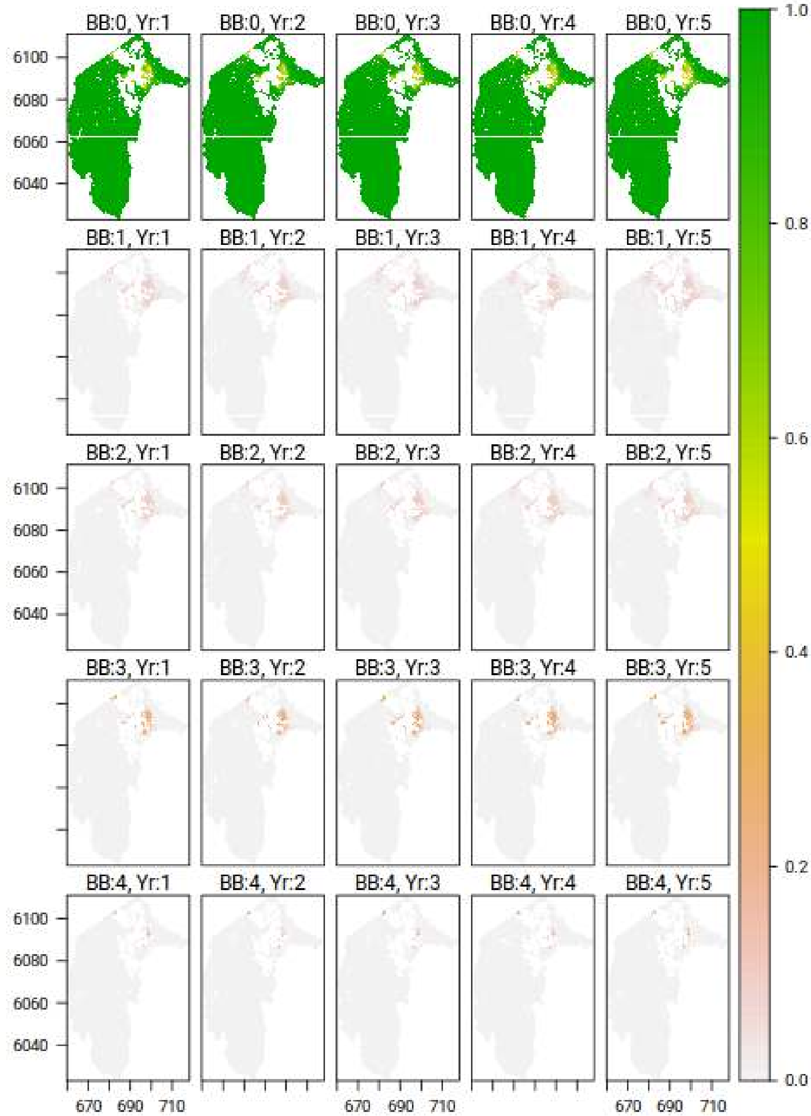


Figure 11: Map of the posterior median probability of the aggregated modified Braun-Blanquet scores for Chilean needle grass. Based on data provided by the ACT Government (see license).

```
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq05", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

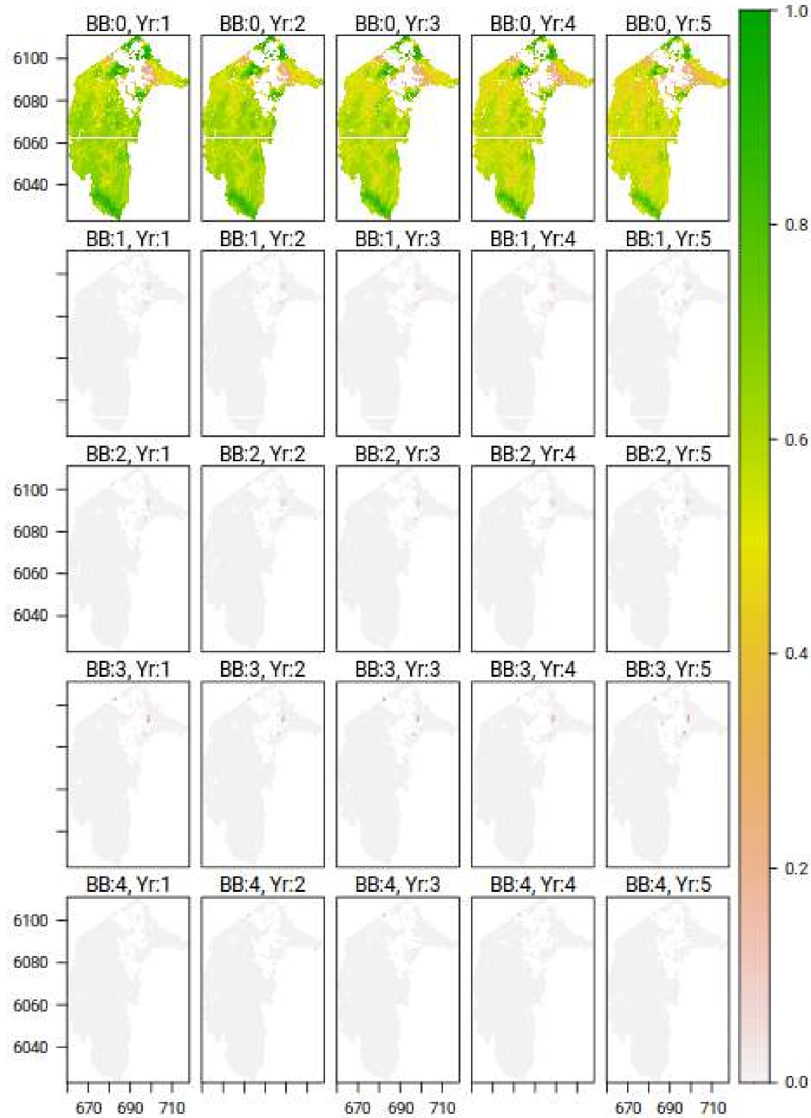


Figure 12: Map of the posterior 0.05 quantile of the aggregated modified Braun-Blanquet scores for Chilean needle grass. Based on data provided by the ACT Government (see license).

```
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq95", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

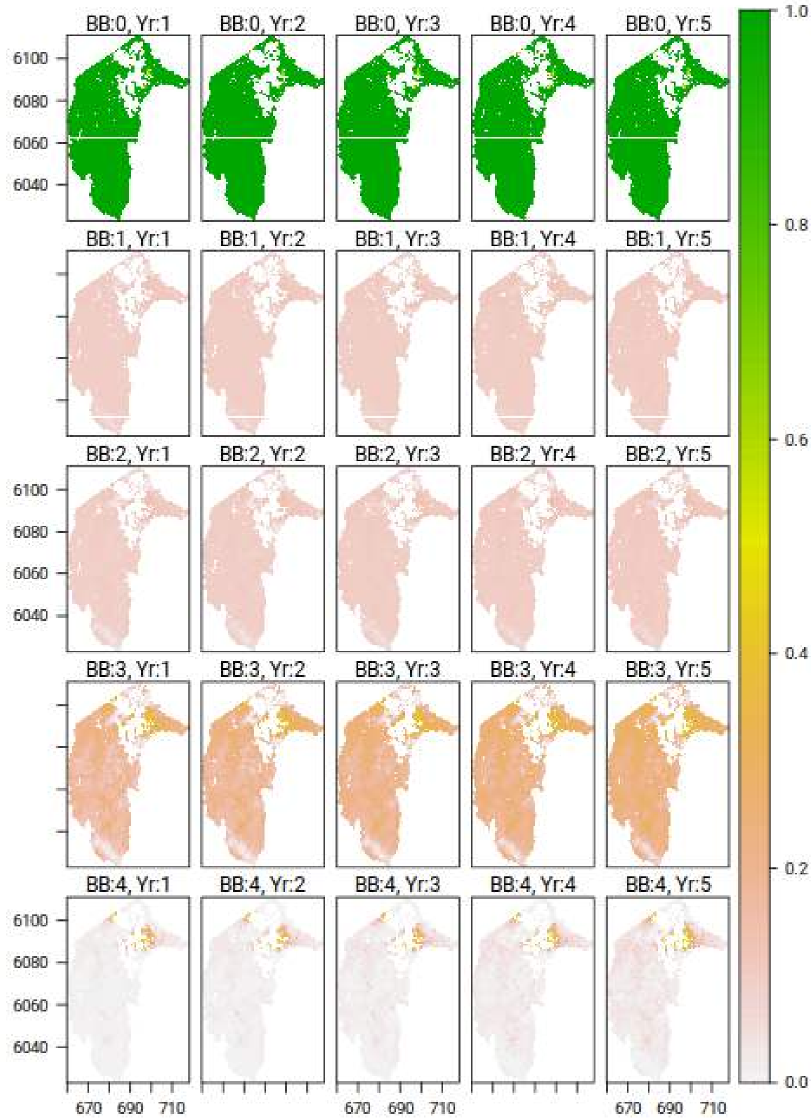


Figure 13: Map of the posterior 0.95 quantile of the aggregated modified Braun-Blanquet scores for Chilean needle grass.

8.3 St John's Wort

8.3.1 M1

```
# INLA glm
StJW_inla <- inla(y_binary ~ - 1 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl_StJW + d_StJW + year +
                    forest + log_access,
                    data = StJW_bin,
                    family = "binomial", Ntrials = 1,
```

```
                    control.family = list(link = "cloglog"),
                    control.compute = list(dic = TRUE))
summary(StJW_inla)

## Time used:
##     Pre = 0.467, Running = 1.73, Post = 0.0276, Total = 2.23
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1      -0.170 0.093     -0.352   -0.170      0.012 -0.170   0
## cut_2      -1.110 0.121     -1.348   -1.110     -0.872 -1.110   0
## cut_3      -0.550 0.114     -0.773   -0.550     -0.327 -0.550   0
## cut_4       1.370 0.109      1.156    1.370      1.583  1.370   0
## ctrl_StJW   0.083 0.093     -0.100    0.083      0.266  0.083   0
## d_StJW      0.043 0.024     -0.004    0.043      0.089  0.043   0
## year        0.080 0.021      0.040    0.080      0.121  0.080   0
## forest     -0.213 0.065     -0.341   -0.213     -0.085 -0.213   0
## log_access -0.166 0.033     -0.230   -0.166     -0.101 -0.166   0
##
## Deviance Information Criterion (DIC) ...............: 3443.92
## Deviance Information Criterion (DIC, saturated) ....: 3441.07
## Effective number of parameters .....................: 8.98
##
## Marginal log-Likelihood:  -1771.56
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

8.3.2 M2

```
# Projection matrix: A
locs2 <- as.matrix(StJW_bin[, c("easting", "northing")])
A1 <- inla.spde.make.A(meshACT, loc = locs2)

stk1 <- inla.stack(
  data = list(y_binary = StJW_bin$y_binary),
  A = list(A1, 1),
  effects = list(
    knot = 1:spde1$n.spde,
     data.frame(StJW_bin[ , c("cut_1","cut_2","cut_3","cut_4",
    "ctrl_StJW", "d_StJW",  "year", "forest", "log_access")])
    ),
  tag = 'est')

StJW_inlaS <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl_StJW + d_StJW + year +
                    forest + log_access +
```

```
                     f(knot, model = spde1),
                     data = inla.stack.data(stk1),
                     control.predictor = list(A = inla.stack.A(stk1)),
                     family = "binomial", Ntrials = 1,
                     control.family = list(link = "cloglog"),
                     control.compute = list(dic = TRUE))

summary(StJW_inlaS)

## Time used:
##     Pre = 0.64, Running = 144, Post = 0.546, Total = 145
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1      -0.125 0.310     -0.726   -0.128      0.495 -0.128   0
## cut_2      -0.508 0.320     -1.128   -0.511      0.133 -0.511   0
## cut_3       0.295 0.319     -0.322    0.291      0.933  0.291   0
## cut_4       2.841 0.326      2.212    2.836      3.495  2.837   0
## ctrl_StJW   0.085 0.148     -0.206    0.086      0.374  0.085   0
## d_StJW      0.073 0.036      0.003    0.073      0.144  0.073   0
## year        0.258 0.029      0.201    0.258      0.316  0.258   0
## forest     -0.308 0.106     -0.516   -0.308     -0.101 -0.308   0
## log_access -0.070 0.052     -0.171   -0.070      0.033 -0.070   0
##
## Random effects:
##   Name   Model
##     knot SPDE2 model
##
## Model hyperparameters:
##                  mean    sd 0.025quant 0.5quant 0.975quant mode
## Range for knot 4.66 0.817       3.27     4.58       6.47 4.43
## Stdev for knot 1.46 0.156       1.18     1.45       1.79 1.44
##
## Deviance Information Criterion (DIC) ...............: 2740.20
## Deviance Information Criterion (DIC, saturated) ....: 2737.36
## Effective number of parameters .....................: 127.96
##
## Marginal log-Likelihood:  -1507.20
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

StJW_inlaS$summary.fixed

##                   mean         sd   0.025quant    0.5quant  0.975quant
## cut_1      -0.12482930 0.30994110 -0.726142270 -0.12802368  0.49526782
## cut_2      -0.50776089 0.32014437 -1.127758110 -0.51133718  0.13305313
## cut_3       0.29466108 0.31868965 -0.321997657  0.29090291  0.93311788
## cut_4       2.84080007 0.32575250  2.212242730  2.83643059  3.49454987
```

```
## ctrl_StJW   0.08514985 0.14780790 -0.205764866  0.08550006  0.37406618
## d_StJW      0.07333117 0.03602239  0.002735358  0.07331400  0.14402439
## year        0.25790538 0.02929341  0.200720071  0.25781329  0.31561864
## forest     -0.30782255 0.10575572 -0.515545669 -0.30771930 -0.10068781
## log_access -0.06985039 0.05198030 -0.171009370 -0.07013130  0.03291041
##                   mode          kld
## cut_1      -0.12776144 2.527680e-08
## cut_2      -0.51112066 3.075377e-08
## cut_3       0.29110611 3.272600e-08
## cut_4       2.83660084 4.044356e-08
## ctrl_StJW   0.08549987 1.482600e-09
## d_StJW      0.07331383 5.975678e-11
## year        0.25781389 2.597014e-09
## forest     -0.30771914 2.741919e-10
## log_access -0.07013145 7.508093e-09

# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=StJW_inlaS,
                            Aproj=A1,
                            DataLocs=as.matrix(StJW_bin[, c("easting", "northing")]),
                            spdeName = "knot")

## [1] TRUE
```

8.3.3 M3

```
# number of years
nyear <- length(unique(StJW_bin$year))

# year index
StJW_bin$yearIdx <- StJW_bin$year - min(StJW_bin$year) + 1

# SPDE model index set
indexs <- inla.spde.make.index('s', n.spde = spde1$n.spde, n.group = nyear)

# Projection matrix by year
A_st <- inla.spde.make.A(mesh = meshACT, loc = locs2,
                         group = StJW_bin$yearIdx)

# stack
stkc_st <- inla.stack(data = list(y_binary = StJW_bin$y_binary),
                      A = list(A_st, 1),
                      effects = list(indexs,
                       data.frame(
```

```
                             StJW_bin[ , c("cut_1","cut_2","cut_3","cut_4",
                                      "ctrl_StJW", "d_StJW",  "year",
                                      "forest", "log_access")]
                             )
                           ),
                      tag = 'est_st')

(INLA::inla.getOption("inla.mode"))

## [1] "compact"

StJW_inlaST <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                           ctrl_StJW + d_StJW + year +
                           forest + log_access +
                           f(s, model = spde1, group = s.group,
                           control.group = list(model = 'ar1', hyper = h.spec)),
                    data = inla.stack.data(stkc_st),
                    control.predictor = list(A = inla.stack.A(stkc_st)),
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute=list(config = TRUE, # Store internal GMRF approximations (
                                        dic = TRUE)
                  )


summary(StJW_inlaST)

## Time used:
##     Pre = 0.581, Running = 1676, Post = 6.69, Total = 1683
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1       -0.231 0.440     -1.097   -0.232      0.642 -0.232   0
## cut_2       -0.508 0.448     -1.387   -0.509      0.383 -0.509   0
## cut_3        0.373 0.448     -0.506    0.371      1.263  0.371   0
## cut_4        3.195 0.460      2.298    3.191      4.116  3.191   0
## ctrl_StJW    0.348 0.169      0.016    0.348      0.681  0.348   0
## d_StJW       0.038 0.044     -0.049    0.038      0.125  0.038   0
## year         0.162 0.107     -0.057    0.165      0.366  0.165   0
## forest      -0.246 0.107     -0.456   -0.246     -0.038 -0.246   0
## log_access -0.096 0.053     -0.199   -0.096      0.009 -0.096   0
##
## Random effects:
##   Name    Model
##     s SPDE2 model
##
## Model hyperparameters:
##                   mean    sd 0.025quant 0.5quant 0.975quant  mode
## Range for s     6.500 1.071      4.658    6.410      8.864 6.227
```

```
## Stdev for s    1.792 0.193      1.444    1.781      2.202 1.759
## GroupRho for s 0.918 0.024      0.861    0.921      0.956 0.927
##
## Deviance Information Criterion (DIC) ...............: 2625.23
## Deviance Information Criterion (DIC, saturated) ....: 2622.39
## Effective number of parameters .....................: 183.06
##
## Marginal log-Likelihood:  -1470.38
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

```r
# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=StJW_inlaST,
                            Aproj=A_st,
                            DataLocs=as.matrix(StJW_bin[, c("easting", "northing", "yearIdx")]
                            spdeName = "s")
```

```
## [1] TRUE
```

```r
# output summary table
out.df <- data.frame(
  cbind(
    c("\\multirow{12}{*}{Saint John's wort}", rep("", 11)),
    gsub("_", "\\\\_", c(row.names(StJW_inlaST$summary.fixed),
      row.names(StJW_inlaST$summary.hyperpar)
    )),
    rbind(
      StJW_inlaST$summary.fixed[ , c("0.025quant", "0.5quant", "0.975quant")],
      StJW_inlaST$summary.hyperpar[ , c("0.025quant", "0.5quant", "0.975quant")]
    )
  )
)
names(out.df) <- c("Species", "Parameter", "q0.025", "q0.50", "q0.975")
print(xtable::xtable(out.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "StJWtab.txt")

# DIC
dic.df <- data.frame(
  cbind(
    c("\\multirow{3}{*}{Saint John's Wort}", rep("", 2)),
    c("M1", "M2", "M3"),
    round(c(StJW_inla$dic$dic, StJW_inlaS$dic$dic, StJW_inlaST$dic$dic))
  )
```

```
)
names(dic.df) <- c("Species", "Model", "DIC")
print(xtable::xtable(dic.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "StJWdic.txt")
```

8.3.4 Prediction for M3

Use the INLA MC sampler to predict posterior marginal probabilities of the aggregated modified Braun-Blanquet score by year.

```
# spatio-temporal control effort
StJWcontrol.t <- system.file("extdata/SJW_DaysSinceControl.tif", package="DSTSOM")
StJWcontrol.r <- terra::rast(StJWcontrol.t)/365 # duration since control in years
# metres to kms
StJWcontrol.r <- project(StJWcontrol.r, gsub("units=m", "units=km",
                             crs(StJWcontrol.r, proj = TRUE)))
StJWcontrol <- terra::values(StJWcontrol.r)[FnGcells, ]
summary(StJWcontrol) # 104 NAs in each year

##  DaySinceCon_Ref20190701 DaySinceCon_Ref20200701 DaySinceCon_Ref20210701
##  Min.   :0.0000          Min.   :0.0000          Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.0000          1st Qu.:0.0000
##  Median :0.0000          Median :0.0000          Median :0.0000
##  Mean   :0.1468          Mean   :0.2021          Mean   :0.2413
##  3rd Qu.:0.0000          3rd Qu.:0.0000          3rd Qu.:0.0000
##  Max.   :4.4110          Max.   :5.4137          Max.   :6.4137
##  NA's   :104             NA's   :104             NA's   :104
##  DaySinceCon_Ref20220701 DaySinceCon_Ref20230701
##  Min.   :0.0000          Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.0000
##  Median :0.0000          Median :0.0000
##  Mean   :0.2888          Mean   :0.3452
##  3rd Qu.:0.0000          3rd Qu.:0.0000
##  Max.   :7.4137          Max.   :8.4137
##  NA's   :104             NA's   :104

StJWcontrol <- StJWcontrol[-allNAs, ]
stStJW.ls <- list(
  ctrl_StJW = ifelse(StJWcontrol > 0, 1, 0),
  d_StJW = StJWcontrol
)
rm(StJWcontrol)

# form spatio-temporal array
StJWarr <- array(NA, dim = c(nrow(data4pred01), 5, 5),
```

```
                 dimnames = list(
                   sites = 1:nrow(data4pred01),
                   predictors = c("ctrl_StJW", "d_StJW", "year", "forest", "log_access"),
                   year = 1:5
                 ))

# spatio-temporal covariates
StJWarr[ , "ctrl_StJW", ] <- stStJW.ls[["ctrl_StJW"]]
StJWarr[ , "d_StJW", ] <- stStJW.ls[["d_StJW"]]

# spatial covariates
for (i in 1:5) {
  StJWarr[ , "forest", i] <- data4pred01[ , "forest"]
  StJWarr[ , "log_access", i] <- data4pred01[ , "log_access"]
}

# temporal covariates
for (i in 1:5) {
  StJWarr[ , "year", i] <- i - 0.5
}

# sign reversal convention for fixed effects
StJWarr <- -StJWarr

# Prediction
pt <- proc.time()
sjw_pred <- predict_DSTSOM(
                             fm_inla = StJW_inlaST,
                             nSample = 10000,
                             arrPred = StJWarr,
                             A_s_Pred = A_s_Pred01
                             )
proc.time() - pt

##     user   system  elapsed
## 3167.033 2184.757 5553.896

str(sjw_pred)

## List of 2
##  $ MCsummary: num [1:206289, 1:5, 1:3, 1:5] 0.671 0.655 0.743 0.736 0.754 ...
##   ..- attr(*, "dimnames")=List of 4
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "cut_1" "cut_2" "cut_3" "cut_4" ...
##   .. ..$ : chr [1:3] "quant05" "median" "quant95"
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
##  $ maxDev   : num [1:206289, 1:5] 2.22e-16 2.22e-16 2.22e-16 2.22e-16 2.22e-16 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
```

```
rasterVis::levelplot(StJWcontrol.r,
                     names.att = paste0("Year ", 1:5), layout = c(5, 1),
                     par.settings =
                       latticeExtra::custom.theme(
                         region = c("white", RColorBrewer::brewer.pal(9, "Oranges")[2:9])
                       ),
                     panel = function(...) {
                       panel.levelplot(...)
                       sp::sp.polygons(as(sf::st_geometry(act.mga), "Spatial"))
                     }
                     )
```

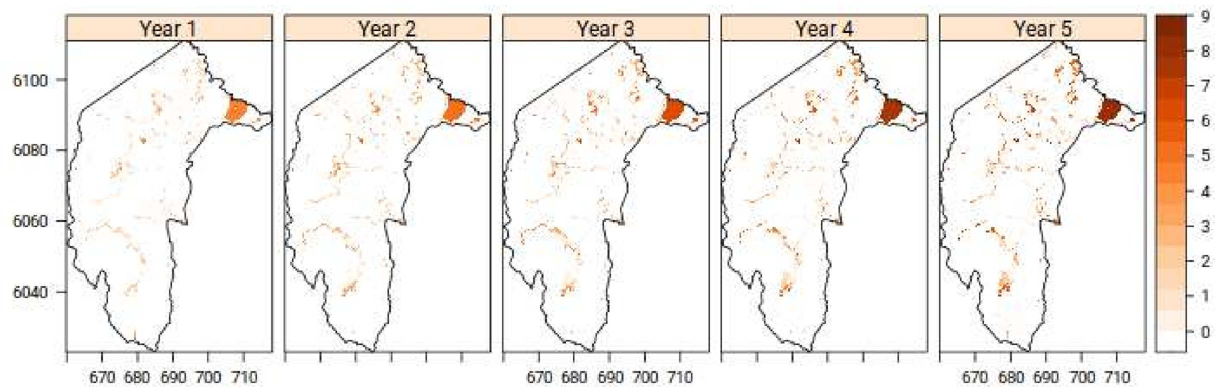


Figure 14: Map of duration since last control effort (colour key is in units of years) on 1 July of each year for St Johns Wort. Based on data provided by the ACT Government (see license).

```
# predictive posterior  marginal probabilities by modified BB score and year
# create raster
tempRast <- rast(covars,
                 nlyrs = 5*5*3, # 5 BB levels by 5 years for median, q0.05 and q0.95
                 names = c(paste0(paste0("Y_", 1:5, "_Bmedian_", rep(0:4, each = 5))),
                           paste0(paste0("Y_", 1:5, "_Bq05_", rep(0:4, each = 5))),
                           paste0(paste0("Y_", 1:5, "_Bq95_", rep(0:4, each = 5)))
                 )
)
```

```r
# modified BB scores
for (bb in 0:4) {
  # years
  for (yy in 1:5) {
    tempRast[[paste0(paste0("Y_", yy), "_Bmedian_", bb)]][data4pred01$PredCell] <-
      sjw_pred$MCsummary[ , bb + 1, "median", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq05_", bb)]][data4pred01$PredCell] <-
      sjw_pred$MCsummary[ , bb + 1, "quant05", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq95_", bb)]][data4pred01$PredCell] <-
      sjw_pred$MCsummary[ , bb + 1, "quant95", yy]
  }
}
```

```r
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bmedian", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

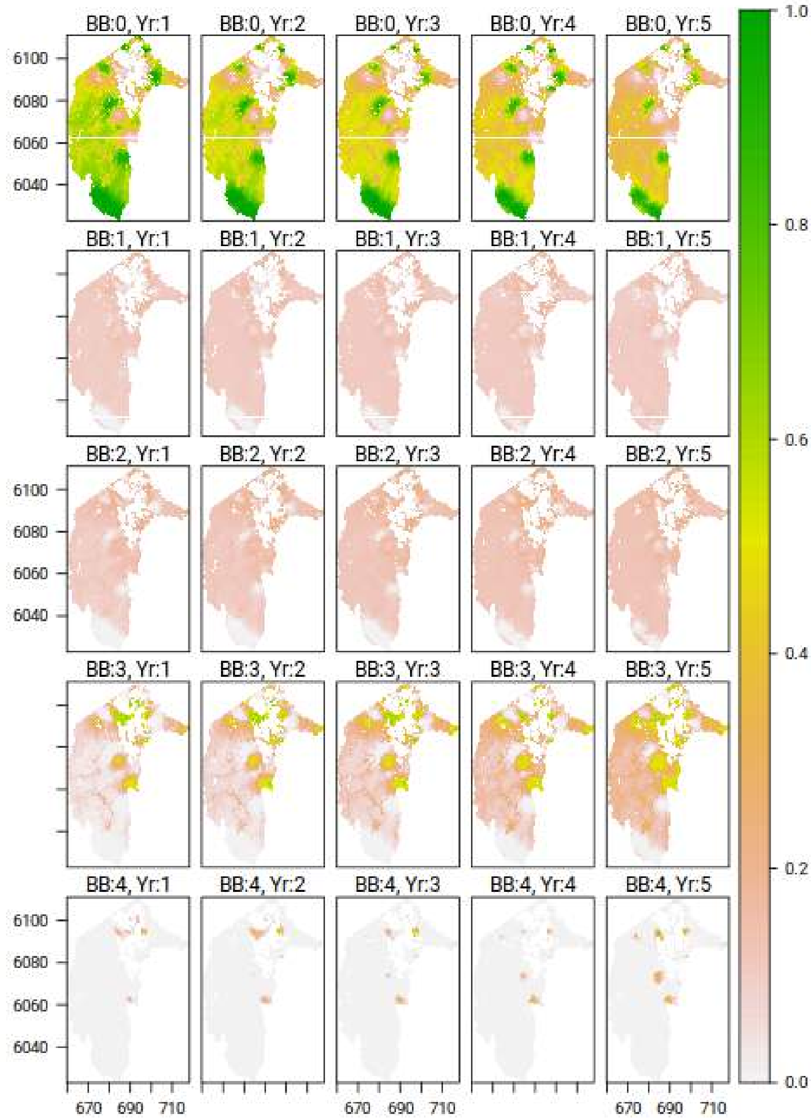


Figure 15: Map of the posterior median probability of the aggregated modified Braun-Blanquet scores for St Johns Wort. Based on data provided by the ACT Government (see license).

```
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq05", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

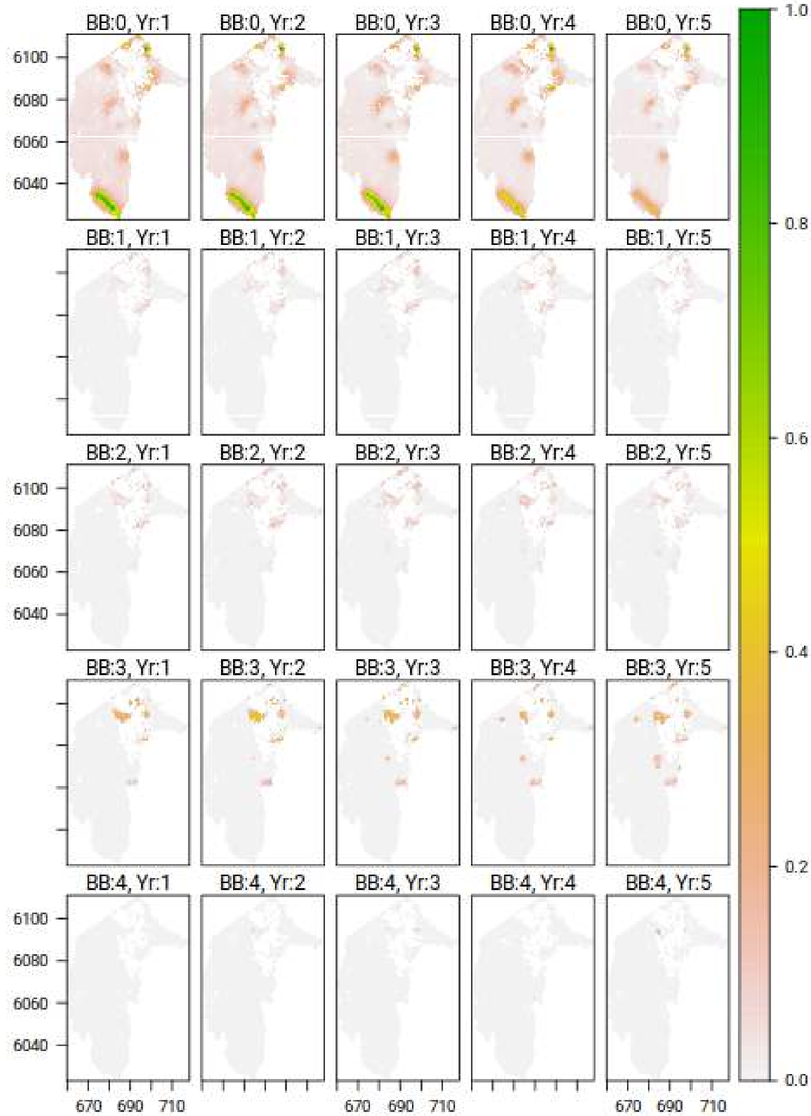


Figure 16: Map of the posterior 0.05 quantile of the aggregated modified Braun-Blanquet scores for St Johns Wort. Based on data provided by the ACT Government (see license).

```
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq95", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

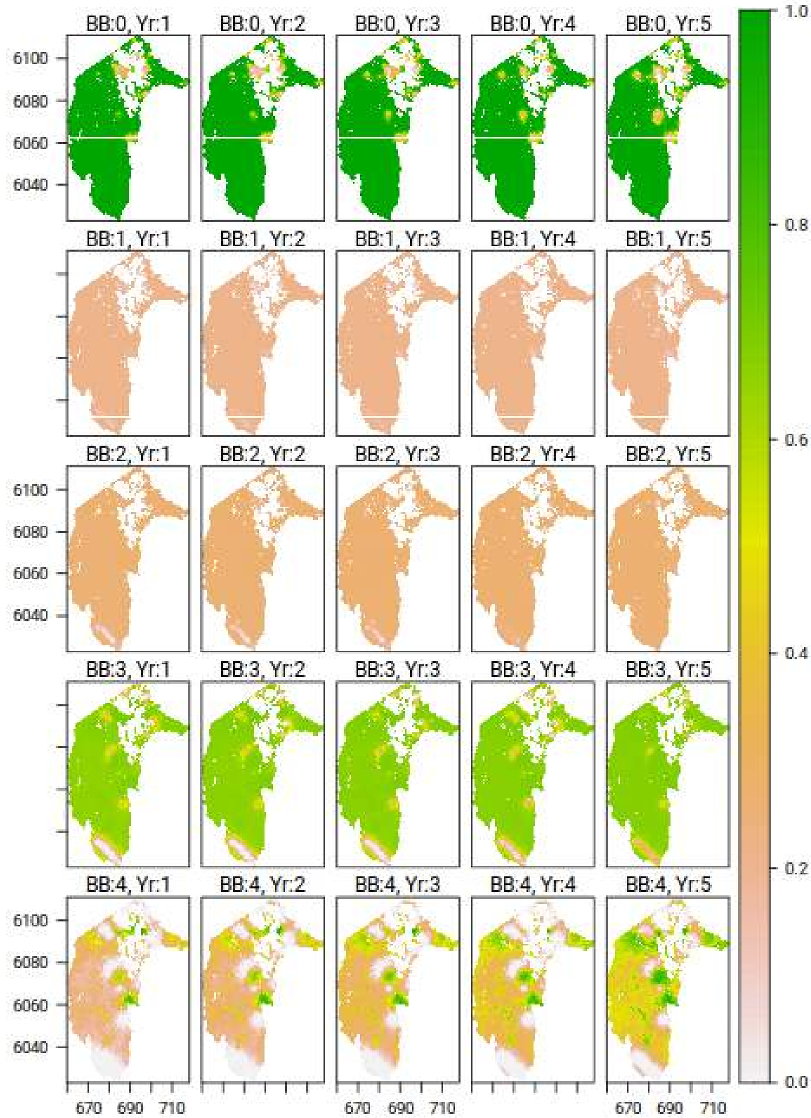


Figure 17: Map of the posterior 0.95 quantile of the aggregated modified Braun-Blanquet scores for St Johns Wort.

8.4 Serrated Tussock

8.4.1 M1

```
# INLA glm
St_inla <- inla(y_binary ~ - 1 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl_St + d_St + year +
                    forest + log_access,
                    data = St_bin,
                    family = "binomial", Ntrials = 1,
```

```
                    control.family = list(link = "cloglog"),
                    control.compute = list(dic = TRUE))
summary(St_inla)

## Time used:
##     Pre = 0.456, Running = 1.51, Post = 0.0273, Total = 1.99
## Fixed effects:
##              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1       0.927 0.094      0.743    0.927      1.111  0.927   0
## cut_2       0.552 0.123      0.311    0.552      0.794  0.552   0
## cut_3       0.637 0.157      0.329    0.637      0.946  0.637   0
## cut_4       1.485 0.200      1.094    1.485      1.877  1.485   0
## ctrl_St     1.126 0.094      0.941    1.126      1.311  1.126   0
## d_St       -0.177 0.024     -0.223   -0.177     -0.131 -0.177   0
## year        0.191 0.024      0.145    0.191      0.238  0.191   0
## forest     -0.043 0.069     -0.178   -0.043      0.092 -0.043   0
## log_access  0.077 0.032      0.015    0.077      0.139  0.077   0
##
## Deviance Information Criterion (DIC) ...............: 1784.53
## Deviance Information Criterion (DIC, saturated) ....: 1783.79
## Effective number of parameters .....................: 8.98
##
## Marginal log-Likelihood:  -940.48
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

8.4.2 M2

```
# Projection matrix: A
locs2 <- as.matrix(St_bin[, c("easting", "northing")])
A1 <- inla.spde.make.A(meshACT, loc = locs2)

stk1 <- inla.stack(
  data = list(y_binary = St_bin$y_binary),
  A = list(A1, 1),
  effects = list(
    knot = 1:spde1$n.spde,
     data.frame(St_bin[ , c("cut_1","cut_2","cut_3","cut_4",
    "ctrl_St", "d_St",  "year", "forest", "log_access")])
    ),
  tag = 'est')

St_inlaS <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                    ctrl_St + d_St + year +
                    forest + log_access +
```

```r
                     f(knot, model = spde1),
                     data = inla.stack.data(stk1),
                     control.predictor = list(A = inla.stack.A(stk1)),
                     family = "binomial", Ntrials = 1,
                     control.family = list(link = "cloglog"),
                     control.compute = list(dic = TRUE))

summary(St_inlaS)

## Time used:
##     Pre = 0.608, Running = 126, Post = 4.54, Total = 131
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1        1.290 0.231      0.850    1.285      1.760  1.285   0
## cut_2        1.437 0.259      0.944    1.432      1.963  1.432   0
## cut_3        1.835 0.289      1.282    1.831      2.418  1.831   0
## cut_4        2.992 0.342      2.335    2.987      3.677  2.987   0
## ctrl_St      0.637 0.155      0.333    0.637      0.941  0.637   0
## d_St        -0.082 0.034     -0.148   -0.082     -0.015 -0.082   0
## year         0.191 0.031      0.131    0.191      0.252  0.191   0
## forest      -0.120 0.111     -0.338   -0.121      0.097 -0.121   0
## log_access   0.003 0.051     -0.098    0.003      0.103  0.003   0
##
## Random effects:
##   Name   Model
##     knot SPDE2 model
##
## Model hyperparameters:
##                  mean    sd 0.025quant 0.5quant 0.975quant  mode
## Range for knot 3.908 0.917      2.446    3.793       6.04 3.560
## Stdev for knot 0.906 0.127      0.682    0.898       1.18 0.882
##
## Deviance Information Criterion (DIC) ...............: 1519.50
## Deviance Information Criterion (DIC, saturated) ....: 1518.76
## Effective number of parameters .....................: 89.95
##
## Marginal log-Likelihood:  -855.33
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

St_inlaS$summary.fixed

##                    mean         sd  0.025quant     0.5quant  0.975quant
## cut_1       1.290142866 0.23132200  0.84989878  1.284984639  1.76048123
## cut_2       1.437446336 0.25912953  0.94353751  1.432148230  1.96252884
## cut_3       1.835462200 0.28919841  1.28152841  1.830587242  2.41840589
## cut_4       2.991769382 0.34165099  2.33463445  2.987024493  3.67747657
```

```
## ctrl_St     0.637383208 0.15489134  0.33333648  0.637440436  0.94110672
## d_St       -0.081631971 0.03371935 -0.14763065 -0.081681203 -0.01535346
## year        0.191279268 0.03073759  0.13119179  0.191211713  0.25175168
## forest     -0.120456072 0.11086176 -0.33756965 -0.120590607  0.09741500
## log_access  0.002582242 0.05132360 -0.09816786  0.002585453  0.10331942
##                    mode          kld
## cut_1       1.285279064 7.777274e-08
## cut_2       1.432453417 8.158733e-08
## cut_3       1.830923379 6.765254e-08
## cut_4       2.987402970 5.442730e-08
## ctrl_St     0.637440842 3.935219e-10
## d_St       -0.081681480 5.815040e-10
## year        0.191211518 1.245861e-09
## forest     -0.120593307 6.214915e-10
## log_access  0.002586878 1.761933e-09

# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=St_inlaS,
                            Aproj=A1,
                            DataLocs=as.matrix(St_bin[, c("easting", "northing")]),
                            spdeName = "knot")

## [1] TRUE
```

8.4.3 M3

```
# number of years
nyear <- length(unique(St_bin$year))

# year index
St_bin$yearIdx <- St_bin$year - min(St_bin$year) + 1

# SPDE model index set
indexs <- inla.spde.make.index('s', n.spde = spde1$n.spde, n.group = nyear)

# Projection matrix by year
A_st <- inla.spde.make.A(mesh = meshACT, loc = locs2,
                         group = St_bin$yearIdx)

# stack
stkc_st <- inla.stack(data = list(y_binary = St_bin$y_binary),
                      A = list(A_st, 1),
                      effects = list(indexs,
                        data.frame(
```

```r
                              St_bin[ , c("cut_1","cut_2","cut_3","cut_4",
                                       "ctrl_St", "d_St", "year",
                                       "forest", "log_access")]
                              )
                            ),
                      tag = 'est_st')

(INLA::inla.getOption("inla.mode"))

## [1] "compact"

St_inlaST <- INLA::inla(y_binary ~ 0 + cut_1 + cut_2 + cut_3 + cut_4 +
                            ctrl_St + d_St + year +
                            forest + log_access +
                            f(s, model = spde1, group = s.group,
                            control.group = list(model = 'ar1', hyper = h.spec)),
                    data = inla.stack.data(stkc_st),
                    control.predictor = list(A = inla.stack.A(stkc_st)),
                    family = "binomial", Ntrials = 1,
                    control.family = list(link = "cloglog"),
                    control.compute=list(config = TRUE,  # Store internal GMRF approximations
                                         dic = TRUE)
                  )

summary(St_inlaST)

## Time used:
##     Pre = 0.763, Running = 1740, Post = 7.23, Total = 1748
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## cut_1        1.371 0.257      0.884    1.364      1.895  1.364   0
## cut_2        1.588 0.290      1.040    1.580      2.180  1.580   0
## cut_3        2.014 0.323      1.402    2.006      2.672  2.005   0
## cut_4        3.227 0.385      2.501    3.218      4.008  3.217   0
## ctrl_St      0.742 0.167      0.415    0.741      1.072  0.741   0
## d_St        -0.104 0.038     -0.179   -0.104     -0.029 -0.104   0
## year         0.212 0.051      0.114    0.211      0.315  0.211   0
## forest      -0.114 0.113     -0.336   -0.114      0.109 -0.114   0
## log_access -0.004 0.053     -0.107   -0.004      0.099 -0.004   0
##
## Random effects:
##   Name    Model
##     s SPDE2 model
##
## Model hyperparameters:
##                  mean    sd 0.025quant 0.5quant 0.975quant  mode
## Range for s     4.020 0.934      2.533    3.903      6.188 3.664
## Stdev for s     0.981 0.135      0.741    0.973      1.271 0.956
```

```
## GroupRho for s 0.950 0.026      0.884    0.955      0.985 0.965
##
## Deviance Information Criterion (DIC) ...............: 1506.71
## Deviance Information Criterion (DIC, saturated) ....: 1505.97
## Effective number of parameters .....................: 116.44
##
## Marginal log-Likelihood:  -852.93
##  is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

```
# Verify if the same predicted random effects are shared by the input data at
#  the same location.
# TRUE means the input data at the same location share the same random effect.
is_SpatialRandom_consistent(inlaObj=St_inlaST,
                            Aproj=A_st,
                            DataLocs=as.matrix(St_bin[, c("easting", "northing", "yearIdx")]),
                            spdeName = "s")
```

```
## [1] TRUE
```

```
# output summary table
out.df <- data.frame(
  cbind(
    c("\\multirow{12}{*}{Serrated tussock}", rep("", 11)),
    gsub("_", "\\\\_", c(row.names(St_inlaST$summary.fixed),
      row.names(St_inlaST$summary.hyperpar)
    )),
    rbind(
      St_inlaST$summary.fixed[ , c("0.025quant", "0.5quant", "0.975quant")],
      St_inlaST$summary.hyperpar[ , c("0.025quant", "0.5quant", "0.975quant")]
    )
  )
)
names(out.df) <- c("Species", "Parameter", "q0.025", "q0.50", "q0.975")
print(xtable::xtable(out.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "Sttab.txt")

# DIC
dic.df <- data.frame(
  cbind(
    c("\\multirow{3}{*}{Serrated tussock}", rep("", 2)),
    c("M1", "M2", "M3"),
    round(c(St_inla$dic$dic, St_inlaS$dic$dic, St_inlaST$dic$dic))
  )
)
```

```r
names(dic.df) <- c("Species", "Model", "DIC")
print(xtable::xtable(dic.df, digits = 3),
      include.rownames = FALSE,
      sanitize.text.function = force,
      file = "Stdic.txt")
```

8.4.4 Prediction for M3

Use the INLA MC sampler to predict posterior marginal probabilities of the aggregated modified Braun-Blanquet score by year.

```r
# spatio-temporal control effort
STcontrol.t <- system.file("extdata/ST_DaysSinceControl.tif", package="DSTSOM")
STcontrol.r <- terra::rast(STcontrol.t)/365 # duration since control in years
# metres to kms
STcontrol.r <- project(STcontrol.r, gsub("units=m", "units=km",
                              crs(STcontrol.r, proj = TRUE)))
STcontrol <- terra::values(STcontrol.r)[FnGcells, ]
summary(STcontrol) # 104 NAs in each year

##  DaySinceCon_Ref20190701 DaySinceCon_Ref20200701 DaySinceCon_Ref20210701
##  Min.   :0.0000          Min.   :0.000           Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.000           1st Qu.:0.0000
##  Median :0.0000          Median :0.000           Median :0.0000
##  Mean   :0.1903          Mean   :0.239           Mean   :0.3114
##  3rd Qu.:0.0000          3rd Qu.:0.000           3rd Qu.:0.0000
##  Max.   :4.4110          Max.   :5.414           Max.   :6.4137
##  NA's   :104             NA's   :104             NA's   :104
##  DaySinceCon_Ref20220701 DaySinceCon_Ref20230701
##  Min.   :0.0000          Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.0000
##  Median :0.0000          Median :0.0000
##  Mean   :0.3787          Mean   :0.4436
##  3rd Qu.:0.0000          3rd Qu.:0.0000
##  Max.   :7.4137          Max.   :8.4137
##  NA's   :104             NA's   :104

STcontrol <- STcontrol[-allNAs, ]
stSt.ls <- list(
  ctrl_St = ifelse(STcontrol > 0, 1, 0),
  d_St = STcontrol
)
rm(STcontrol)

# form spatio-temporal array
Starr <- array(NA, dim = c(nrow(data4pred01), 5, 5),
                dimnames = list(
```

```
                  sites = 1:nrow(data4pred01),
                  predictors = c("ctrl_St", "d_St", "year", "forest", "log_access"),
                  year = 1:5
                ))

# spatio-temporal covariates
Starr[ , "ctrl_St", ] <- stSt.ls[["ctrl_St"]]
Starr[ , "d_St", ] <- stSt.ls[["d_St"]]

# spatial covariates
for (i in 1:5) {
  Starr[ , "forest", i] <- data4pred01[ , "forest"]
  Starr[ , "log_access", i] <- data4pred01[ , "log_access"]
}

# temporal covariates
for (i in 1:5) {
  Starr[ , "year", i] <- i - 0.5
}

# sign reversal convention for fixed effects
Starr <- -Starr

# Prediction
pt <- proc.time()
st_pred <- predict_DSTSOM(
                             fm_inla = St_inlaST,
                             nSample = 10000,
                             arrPred = Starr,
                             A_s_Pred = A_s_Pred01
                             )
proc.time() - pt

##     user   system  elapsed
## 3120.692 2186.899 5878.105

str(st_pred)

## List of 2
##  $ MCsummary: num [1:206289, 1:5, 1:3, 1:5] 0.117 0.116 0.123 0.127 0.13 ...
##   ..- attr(*, "dimnames")=List of 4
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "cut_1" "cut_2" "cut_3" "cut_4" ...
##   .. ..$ : chr [1:3] "quant05" "median" "quant95"
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
##  $ maxDev   : num [1:206289, 1:5] 3.33e-16 3.33e-16 3.33e-16 2.22e-16 3.33e-16 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:206289] "1" "2" "3" "4" ...
```

```
##   .. ..$ : chr [1:5] "year1" "year2" "year3" "year4" ...
```

```
rasterVis::levelplot(STcontrol.r,
                     names.att = paste0("Year ", 1:5), layout = c(5, 1),
                     par.settings =
                       latticeExtra::custom.theme(
                         region = c("white", RColorBrewer::brewer.pal(9, "Oranges")[2:9])
                       ),
                     panel = function(...) {
                       panel.levelplot(...)
                       sp::sp.polygons(as(sf::st_geometry(act.mga), "Spatial"))
                     }
                     )
```

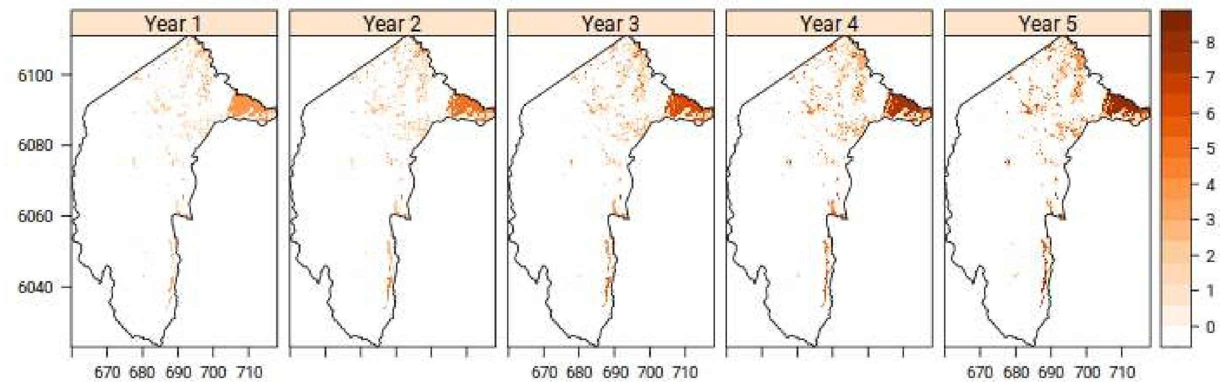


Figure 18: Map of duration since last control effort (colour key is in units of years) on 1 July of each year for serrated tussock. Based on data provided by the ACT Government (see license).

```
# predictive posterior marginal probabilities by modified BB score and year
# create raster
tempRast <- rast(covars,
                  nlyrs = 5*5*3, # 5 BB levels by 5 years for median, q0.05 and q0.95
                  names = c(paste0(paste0("Y_", 1:5, "_Bmedian_", rep(0:4, each = 5))),
                            paste0(paste0("Y_", 1:5, "_Bq05_", rep(0:4, each = 5))),
                            paste0(paste0("Y_", 1:5, "_Bq95_", rep(0:4, each = 5)))
                  )
)
```

```r
# modified BB scores
for (bb in 0:4) {
  # years
  for (yy in 1:5) {
    tempRast[[paste0(paste0("Y_", yy), "_Bmedian_", bb)]][data4pred01$PredCell] <-
      st_pred$MCsummary[ , bb + 1, "median", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq05_", bb)]][data4pred01$PredCell] <-
      st_pred$MCsummary[ , bb + 1, "quant05", yy]
    tempRast[[paste0(paste0("Y_", yy), "_Bq95_", bb)]][data4pred01$PredCell] <-
      st_pred$MCsummary[ , bb + 1, "quant95", yy]
  }
}
```

```r
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bmedian", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

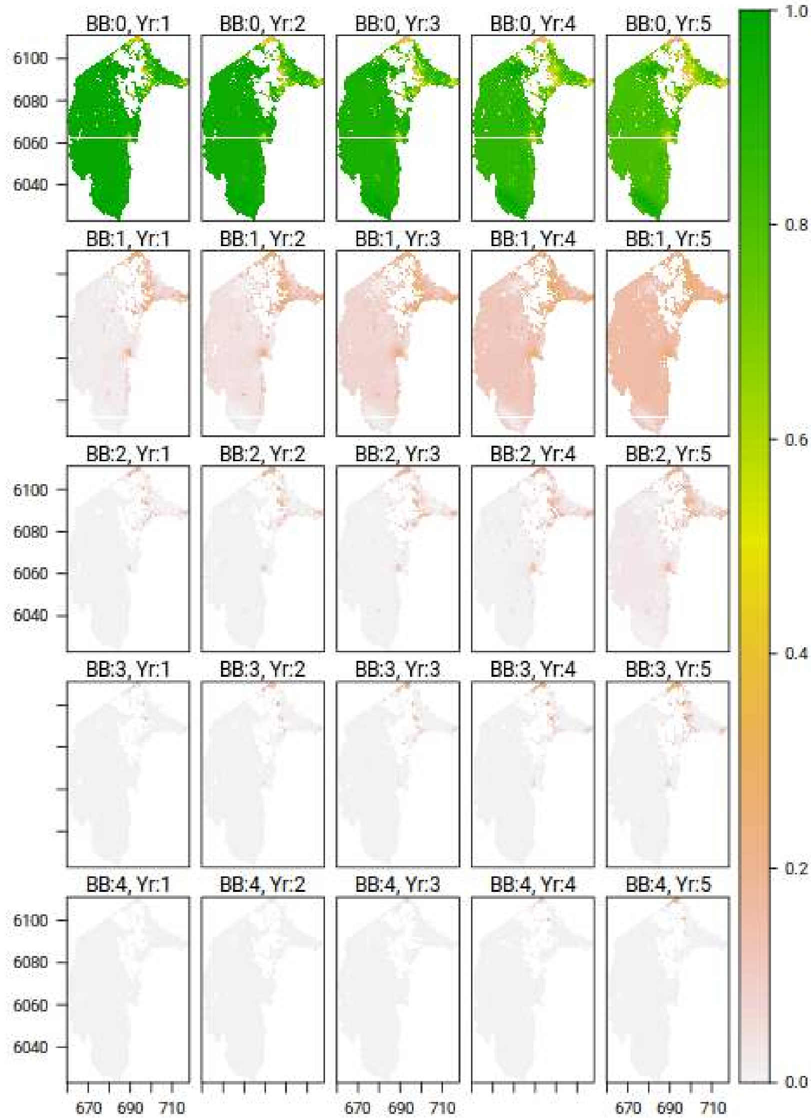


Figure 19: Map of the posterior median probability of the aggregated modified Braun-Blanquet scores for serrated tussock. Based on data provided by the ACT Government (see license).

```
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq05", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

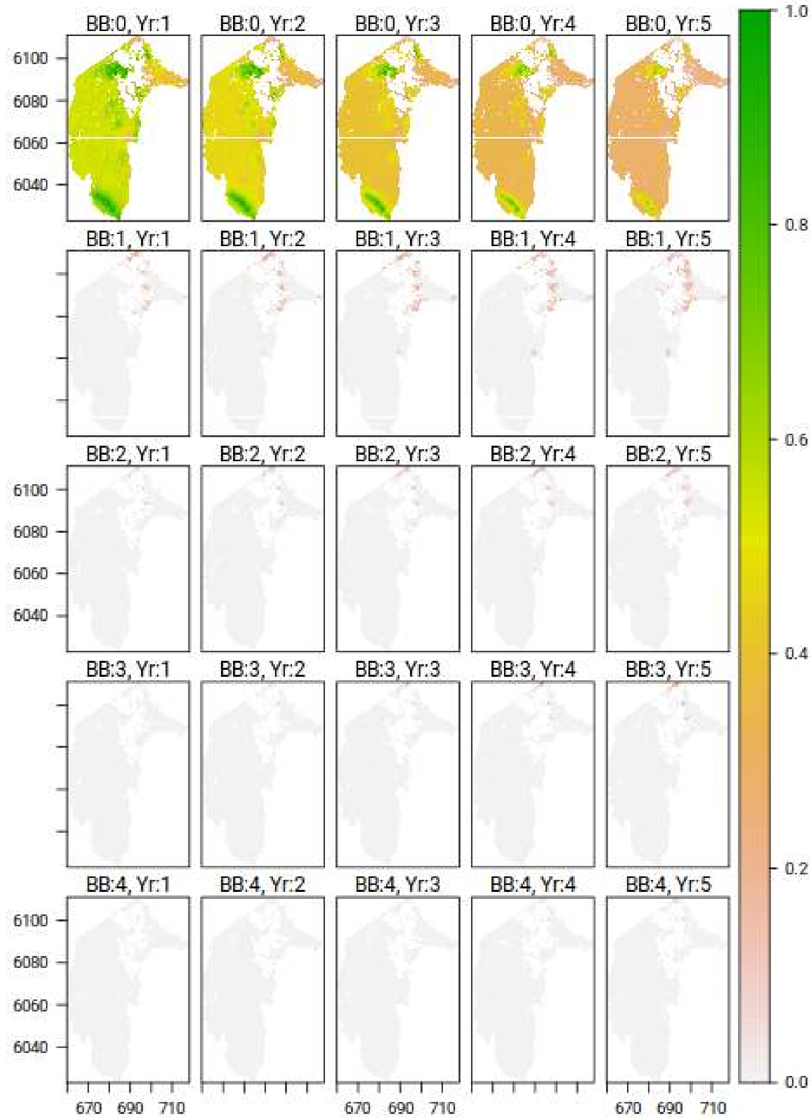


Figure 20: Map of the posterior 0.05 quantile of the aggregated modified Braun-Blanquet scores for serrated tussock. Based on data provided by the ACT Government (see license).

```r
# Map of the aggregated modified Braun-Blanquet score
rasterVis::levelplot(tempRast[[grep("Bq95", names(tempRast))]],
                     layout = c(5, 5), zscaleLog = FALSE,
                     at = ats, col.regions = cols,
                     names.att =
                       paste0(rep(paste0("BB:", 0:4), each = 5), ", Yr:", 1:5)
)
```

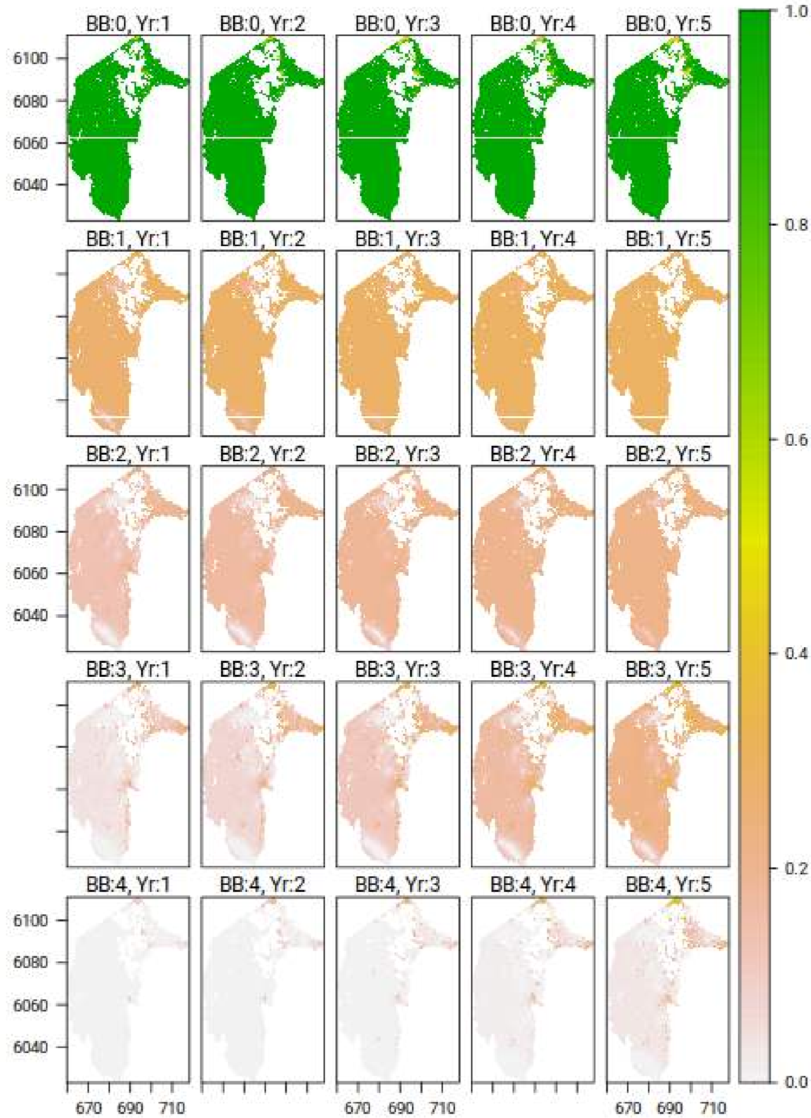


Figure 21: Map of the posterior 0.95 quantile of the aggregated modified Braun-Blanquet scores for serrated tussock. Based on data provided by the ACT Government (see license).

References

Berridge, D. M. and J. Whitehead (1991). Analysis of failure time data with ordinal categories of response. *Statistics in Medicine 10*, 1703–1710.

Fuglstad, G., D. Simpson, F. Lindgren, and H. Rue (2019). Constructing priors that penalize the complexity of Gaussian random fields. *Journal of the American Statistical Association 114*, 445–452.

Gómez-Rubio, V. (2020). *Bayesian inference with INLA*. Boca Raton, FL: Chapman & Hall/CRC Press.

Harrell Jr, F. E. (2023). *rms: Regression Modeling Strategies*. R package version 6.7-1.

Krainski, E. T., V. Gómez-Rubio, H. Bakka, A. Lenzi, D. Castro-Camilo, D. Simpson, F. Lindgren, and H. Rue (2019). *Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA*. Boca Raton, FL: Chapman & Hall/CRC Press.

Lindgren, F. and H. Rue (2015). Bayesian spatial modelling with R-INLA. *Journal of Statistical Software 63*(19), 1–25.

Palmi-Perales, F., V. Gómez-Rubio, R. S. Bivand, M. Cameletti, and H. Rue (2022). Bayesian inference for multivariate spatial models with R-INLA. *arXiv*. arXiv:2212.10976v1.

Tutz, G. (2012). *Regression for Categorical Data*. New York: Cambridge University Press.

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association 99*, 250–261.