An Introduction to RISDM

Scott D. Foster
Data61, CSIRO, Hobart, Tasmania, Australia

Abstract

Species distribution models (SDMs) describe the distribution of a species through space, environment and possibly time. They are based on two types of data input: those that inform the biology of where the species is found (and possibly isn't), and those that describe the environment throughout the study region. An SDM is formed by describing the variation in the species data in terms of the variation in the environmental data (a correlative model). SDMs are a useful and proven tool for science and management.

Standard implementations of SDMs requires a possibly over-restrictive homogeneity in sources of biological data. Typically, the biological data is one of: presence-only (PO), presence-absence (PA), or abundance (AA) data. Results from SDMs using single sources of data will suffer from inherent biases and limitations to that data type. For example, PO data contains observer-bias and presence-absence (PA) and abundance (AA) data is often less geographically expansive and of limited quantity (and hence information content).

Intuitively, it is appealing to incorporate as much biological information into the SDM as possible. This means that multiple data sources and multiple data types should be integrated in a single model. This is the central idea that *integrated* SDMs, or ISDMs, bring (Fletcher Jr. *et al.* 2019; Miller *et al.* 2019; Isaac *et al.* 2020). However, the structure of ISDMs is a necessary step-change in complexity to single data source SDMs – the vagueries of the each contirbuting data sets need to be modelled as well as the differences between data sets. In addition, there are statistical specifications that need to be handled as well, such as handling multiple likelihoods in a single model.

The R-package `RISDM` provides a task-specific tool for fitting ISDMs. RISDM utilises the computational engine of `INLA` Rue *et al.* (2009) as the back end, but hides the `INLA`-specific details from the user (such as the creation of stacks, the specification of spatial models and the specification of multiple likelihoods). The `RISDM` interface is simplified as much as possible, but still remains flexible enough for a wide class of models to be fitted. In particular, the workflow is purposefully designed to follow that of a `glm` or `gam`: model fit, model diagnostics, model summary/interpretation, and prediction.

In this document, we illustrate the process of fitting an ISDM to an invasive weed data set from northern Australia. We illustrate the steps of analysis, and show the effect of altering some of the main parameters/options/control.

Keywords: Integrated Species Distribution Models, ISDM, INLA, SPDE, R.

First Things First (setting up R for using `RISDM`)

Before starting with this introduction to `RISDM`, we need to make sure that everything is set up properly. Much of this will vary from computer to computer, but you must have a working

This is an introduction to the RISDM R-pacakge (available from https://cran.r-project.org/package=RISDM).

version of R installed (preferably the recent release) and a working version of INLA (Rue *et al.* 2009). This vignette was created using R-version R version 4.2.1 (2022-06-23), `INLA` release 23.04.24, and `RISDM` release 1.2.7. It does not matter whether you prefer to use R through a development environment (such as RStudio) or through the command line – the results will be the same. You will need to have installed a recent version of `INLA` and `RISDM`. These can be installed by starting R and then submitting (at command line):

```
install.packages("INLA",repos=c(getOption("repos"),
                 INLA="https://inla.r-inla-download.org/R/stable"), dep=TRUE)
#vignette not built to save computation
devtools::install_github( repo="Scott-Foster/RISDM", build_vignettes=FALSE)
```

The `RISDM` package will then need to be loaded. The raster package is also loaded to handle spatial data.

```
library( RISDM)
library( raster)
```

Introduction to RISDM, via Example

The structure and function calls within `RISDM` are designed to follow that of the `glm()` and `gam()` that are part of the well-used `stats` (R Core Team 2022) and `mgcv` (Wood 2017) packages respectively. The `RISDM` approach is designed to broaden the user-base of ISDM models by leveraging off familiar analytical and computational concepts. While very advanced users may want to fit models that are outside of the scope of `RISDM`, the `RISDM` framework could still be used to provide many of the constructs needed for lower-level modelling with `INLA` (Rue *et al.* 2009).

This document is structured, more-or-less, to match the steps that an analyst follows in an analysis workflow. At least for an archetypical analysis. That is:

1. Preparation (data and analysis mesh)
2. Specifying and Evaluating ('fitting') the model
3. Checking the model for any obvious departures, and understanding/summarising the model.
4. Prediction

The `RISDM` package is introduced here using data on gamba grass in the Northern Territory, Australia. These data are a small subset of all those available, and are intended to be an illustrative 'toy' – containing enough complexity to be real, but not so complex that they provide too much distraction from the method and computation. The data are presented in Foster *et al.* (in review) and the reader should refer there for some more detail and for references.

Preparation

Covariate Data – setting the environmental scene

The first piece of the data puzzle is the environmental data. These data are assumed to be defined on a raster (from the raster package) and must form a raster brick. Forming them into a raster stack ensures that the individual raster layers all have the same extent and the same resolution etc. It is not necessary, but it is highly recommended, that the analyst considers the pattern of missing values within the raster stack – where some covariates are available but others are not. Missing values in covariates in any correlative model can substantially reduce the information content.

The covariate data for the gamba grass example is included as part of the `RISDM` package. There are three raster layers:

1. Accessibility, using the method by (Weiss *et al.* 2018) was used after small modification for the northern Australian context. Here, it measures the travel time to any settlement with a population of more than 200 (Geosciences Australia 2006). The observed distribution of this covariate was right-skewed, so a square root transformation was taken. This was to try and reduce the leverage of unusually large observations. One method of spreading gamba grass is thought to be through human movement. So accessibility may correlate with gamba distribution.

2. Elevation, measured through a digital elevation model (DEM; from Gallant and Austin 2015), may delineate gamba grass distribution (Adams *et al.* 2015). Like accessibility, a square root transformation was used to endeavour to reduce leverage of unusual locations within the analysis.

3. Soil moisture at the root zone (SMRZ Frost *et al.* 2018) may limit gamba grass distribution as it could require moist but not saturated soils (Flores *et al.* 2005; Petty *et al.* 2012).

The gamba grass covariates can be read directly into R and compiled into a raster stack. The distribution of the covariate data has been inspected (elsewhere) and corrective steps have been employed. For the gamba data, the accessibility and the DEM were square-root transformed to reduce skewness of their distributions, which will also lead to reducing the influence in the analysis of very high values. This is to reduce the leverage of these observations and to try and obtain the most amount of reliable information from these data.

```
filenames <- paste0( system.file("extdata", package="RISDM"),
                    c("/GambaExample_sqrtACC_23Mar28.tif",
                      "/GambaExample_sqrtDEM_23Mar28.tif",
                      "/GambaExample_SMRZ_23Mar28.tif"))
covars <- stack( filenames)
names( covars) <- c("ACC","DEM","SMRZ")
print( colnames( coordinates( covars)))  #to see how things are labelled.

[1] "x" "y"

plot( covars)
```

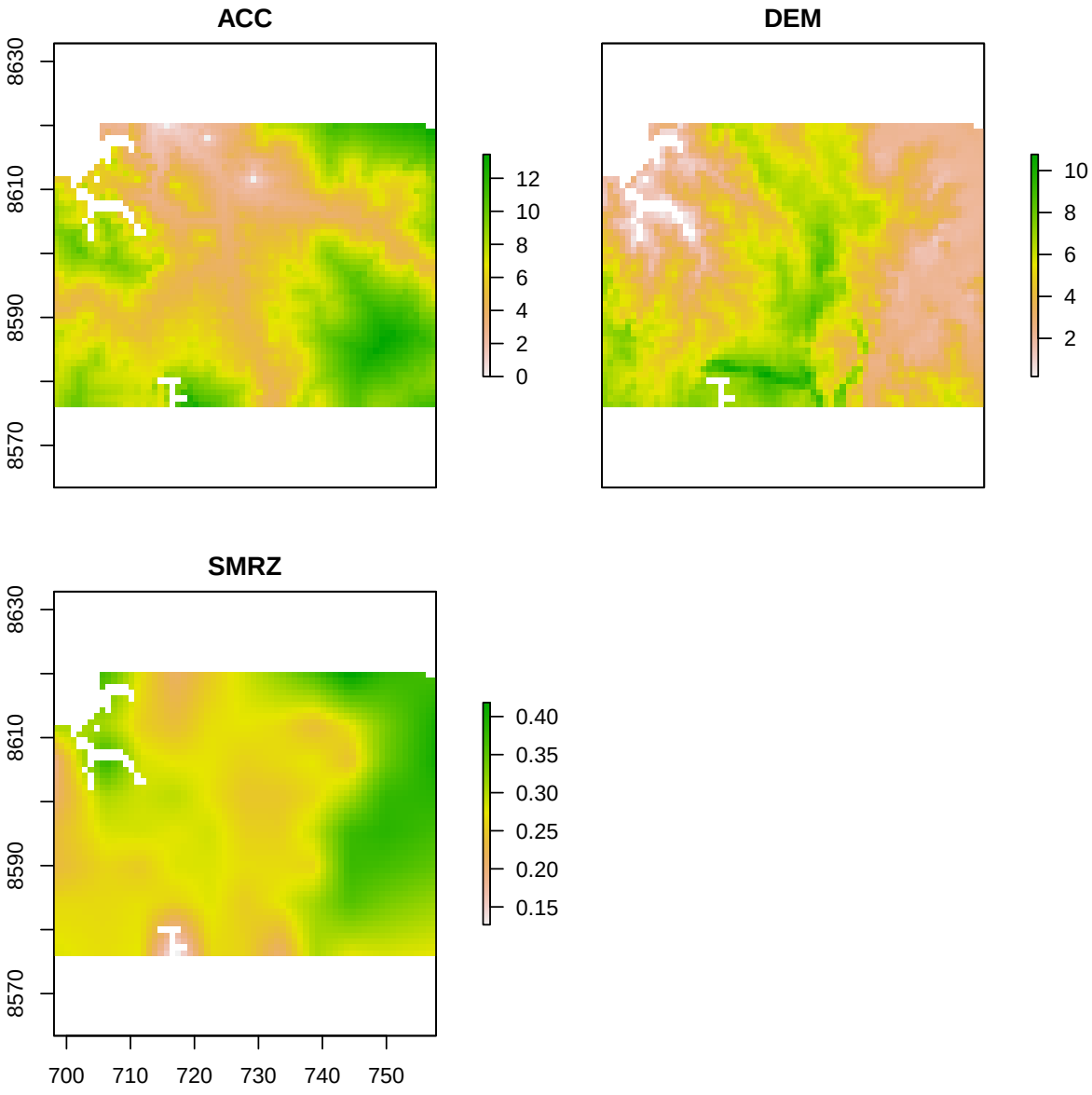


Figure 1: Environmental covariate data for the gamba grass example. Accessibility (ACC) measuring effective distance for humans to travel. Digital elevation model (DEM) giving the altitude. Soil moisture at the root zone (SMRZ) measuring the wetness where it matters for plants. See text for details and references.

It may be that the raster data that an analyst wants to use comes in different resolutions, different projections, different extents and so on. If the analyst is not familiar with conversion of these, then it is suggested that they should look into the `projectRaster`, `aggregate`, `disaggregate`, `mask`, `crop` and `stack` functions from the `raster` package (Hijmans 2022).

For the environmental data in this example (see Figure 1), we have used an equal area project (UTM zone 52 south) with kilometre map units. We recommend the use of projected rasters, as this eases specification and interpretation of spatial processes. However, `RISDM` will still work with whatever coordinate reference system is chosen.

The resolution of the raster is used to define the resolution of the underlying point process in the ISDM. It is reasonably argued that performing point-process computation on grids smaller than this is wasteful as environmental conditions would not then change between many neighbouring cells. As the raster defines point process computational scale, it can sometimes be useful – at the start of a modelling exercise – to coarsen the resolution just so models run faster.

Bias Data

In addition to the environmental covariates, the raster stack can (and should if available) covariates that could delineate the sampling bias of the PO data (e.g. Warton *et al.* 2013). That is, covariates that are suspected of describing the pattern of search effort that could then result in a set of presences. For the gamba grass example, the accessibility (ACC) variable is likely to play this delineating role. When there are more than one data type, there is no issue that a single covariate describes both the distribution of the species *and* the search bias.

Observation Data

The observation data are those that inform the where the species is, is not, and in what quantity/abundance. Each type of data (PO, PA or AA) is required to have its own data object. They are all required to have the same projection as the raster stack for the environment. This assumes that the coordinate names are the same too. The observation data sets are:

Presence-Only (PO) data A `data.frame` containing at least the locations (x and y coordinates) of the presences. The data.frame may also contain extra contextual information, like the survey/database the presence record was recorded in. This contextual information is currently not used in the analysis.

Presence-Absense (PA) and Abundance (AA) data A `data.frame` for PA and a `data.frame` for AA (separately, but neither *has to be* specified). The data.frames contain at least:

1. The locations (x and y coordinates) where the observation occurred.
2. The value of the observation at these locations. For PA data, this is a '0' for an absence and a '1' for a presence. For AA data, this is a positive integer (or zero).
3. The geographical area (sample area) that the sampling method searched to record the observation. Intuitively, a search area of 1m^2 is more likely to produce a '0' than a search area of 1000m^2. If sample area is unknown then it must be either guessed (!?) or have some constant inputed. In either case, the results are unlikely to then be reflective of overall levels of species abundance, but the distribution *may very well be* relative to the actual distribution.

4. (optional) Extra information that could explain variation in the PA or AA data. Such information could include data that might induce heterogeneous detectability (e.g. different sampling tools, different operators, and so on), but it may also include continuous variables. These variables could be included as sampling artefacts, which are only used to model the observations *within its own data type*.

Double-Count (DC) data This is experimental. Double count data arising from two independent observers measuring the same spatial location. These data enable an estimate of detectability for each observer. DC data is not discussed further in this vignette.

The gamba grass observational data is included in the `RISDM` package. To load them into R and to get an idea about their content, use:

```
gamba_PO <- readRDS( system.file("extdata", "Gamba_PO_23Mar28.RDS",
                                 package="RISDM"))
gamba_PA <- readRDS( system.file("extdata", "Gamba_PA_23Mar28.RDS",
                                package="RISDM"))

str( gamba_PO)

'data.frame': 4219 obs. of  6 variables:
 $ genus  : chr  "andropogon" "andropogon" "andropogon" "andropogon" ...
 $ species: chr  "gayanus" "gayanus" "gayanus" "gayanus" ...
 $ date   : Date, format: "2000-06-29" "2001-04-18" ...
 $ year   : num  2000 2001 2001 2001 2001 ...
 $ x      : num  747 737 738 735 728 ...
 $ y      : num  8605 8607 8611 8613 8612 ...

str( gamba_PA)

'data.frame': 500 obs. of  9 variables:
 $ genus   : chr  "andropogon" "andropogon" "andropogon" "andropogon" ...
 $ species : chr  "gayanus" "gayanus" "gayanus" "gayanus" ...
 $ date    : Date, format: "2016-07-01" "2016-07-01" ...
 $ observer: Factor w/ 4 levels "B","J","P","Missing": 4 2 1 1 4 3 4 3 4 4 ...
 $ PA      : logi  FALSE TRUE TRUE TRUE FALSE TRUE ...
 $ Area    : num  40000 40000 40000 40000 40000 40000 40000 40000 40000 40000 ...
 $ year    : num  2016 2016 2016 2016 2016 ...
 $ x       : num  732 732 720 710 740 ...
 $ y       : num  8619 8615 8585 8585 8589 ...
```

For the gamba grass PO data, there are `4219` observations, and their locations are stored in the columns 'x' and 'y'. These labels match the covariate raster stack from before. There is possible confusion as there is also a longitude and a latitude variable, but these are a legacy of a previous reference system. As stated before however, these observation data will be used in a projected form. The remainder of the data provides contextual information about date

of sample, survey program, and so on. These data, for PO observations, are not currently used in `RISDM`.

For the PA data, there are `500` observations, that are randomly chosen from a much larger database. Of these observations (outcome in the 'PA' column), a proportion of `0.45` were presences and `0.55` absences. The locations of the observations are in the 'x' and 'y' columns, and the sampling area is stored in the `Area` column. The sampling area is the same for all observations in these data (they arise from a formal survey methodology). Most of the contextual variables in gamba_PA are not all that useful. However, there might be some detectability differences between observers, and this variable could be added as a sampling artefact.

Analysis Mesh

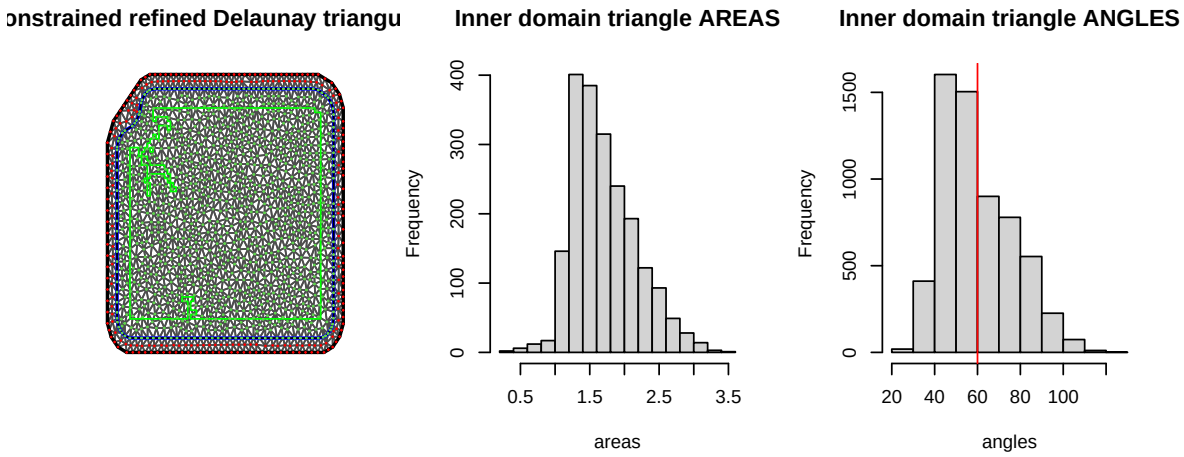
The geostatistical method utilised in `RISDM`, that of Lindgren *et al.* (2011), relies on creating a spatial mesh over which the spatial model is calculated. Some care is needed when specifying the mesh as choices in mesh can lead to changes in inference (Righetto *et al.* 2020; Verdoy 2021). We provide a function in `RISDM`, `makeMesh` to produce a mesh using arguments that we think are more intuitive than those provided in INLA. We note that `makeMesh` 'simply' repacakges the information supplied and submits it to the `INLA` functions. As such, `makeMesh` provides no new methodology.

In `makeMesh`, the mesh creation task is primarily parameterised by the analyst's guess as to the likely effective range for the spatial random effect (distance to small ~ 0.1 expected correlation) and also the number of nodes in both the analysis area and an extension area. The analysis area is taken to be a nonconvex hull surrounding the non-NA values in one of the covariate raster layers, and the extension area is simply a buffered version of the hull. The extension is useful to allow for spatial dependence to be properly described at locations near the edges. For finer control, other arguments can also be given. If not supplied, then defaults are set using rules-of-thumbs obtained largely from Bakka *et al.* (2018) and Krainski *et al.* (2019). An elementary mesh for gamba grass is now generated (note that doPlot=FALSE is chosen as `checkMesh` reproduces these shortly). In this example, the expected dependence is 3km – so the mesh should be OK to estimate dependencies of that order (order larger). There are up to 1000 nodes in the analysis area and up to 350 in the extension area.

```
my.mesh <- makeMesh( covars$DEM, max.n=c(1000, 350), dep.range=3, doPlot=FALSE)
```

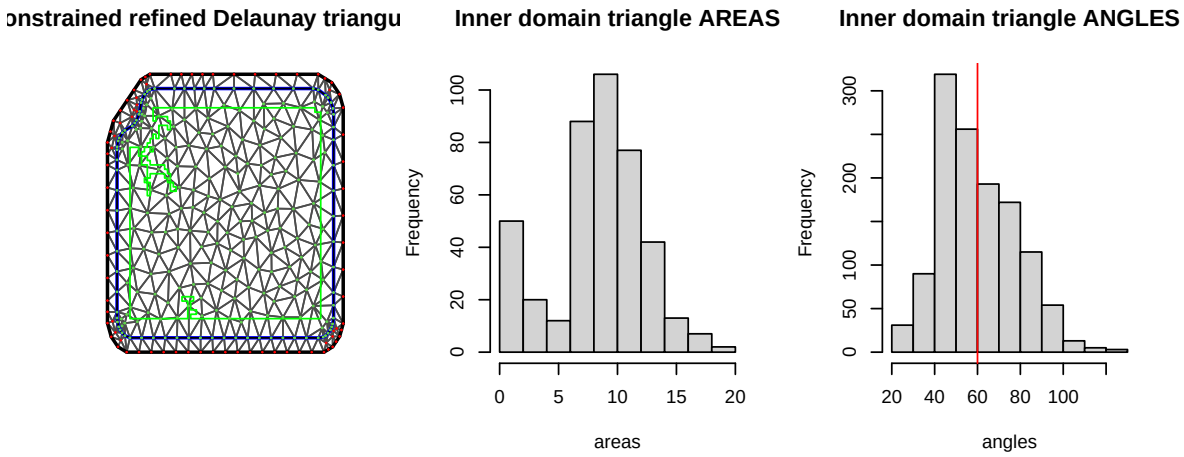
A 'good' mesh is one that covers the regions, particularly the inner region, with triangles that are approximately equally sized and are approximately equilateral (Krainski *et al.* 2019). Further, the edge lengths should be relatively small compared to the (posterior) range of spatial dependence – otherwise the model might erroneously report that there is no dependence in the data when there is. If the edge lengths are too small however, then excessive computation may be needed. The first two conditions can be checked visually using a plot of the mesh (shown in Krainski *et al.* 2019) but also looking at the distribution of triangle areas and angles. The `RISDM` function `checkMesh` produces these plots.

```
checkMesh( my.mesh, my.mesh$hull, ras=covars$DEM)
```



This mesh appears to be adequate – there are not too many large or small triangles and not too many triangles with overly small or large angles. However, in Foster *et al.* (in review), we chose (perhaps unnecessarily) to extend the outer boundary slightly to allow for a better distribution of nodes. This was achieved by adding the argument `offset=7.5`, which is measured in map units (kilometres in this case). As an example of a 'bad' mesh, consider the same spatial region but with fewer nodes.

```
my.mesh.bad <- makeMesh( covars$DEM, max.n=c(250, 30), dep.range=3, doPlot=FALSE)
checkMesh( my.mesh.bad, my.mesh$hull, ras=covars$DEM)
```



The resulting 'bad' mesh has mostly good shaped trianlges, but those around the corners of the inner and outer areas are much smaller and more 'bunched' than elsewhere. This is picked up in the distribution of the triangle areas, which is quite bimodal. However, the triangle shapes appear to be adequate. In general, the more regular the survey area (non-NA values in the raster layer), the easier it will be to produce an adequate mesh. The survey area for gamba grass is quite regular and hence even this 'bad' mesh is not spectacularly bad.

Specifying and Evaluating the Model

With the data objects and the mesh all curated and created, it is time to (finally) start the formal modelling process. There are, unfortunately, a number of moving parts that need to be specified. Let's go through them in turn.

Species Distribution

How the species responds to the environment, as quantified by the layers in the covariate stack (see Section), is defined through a formula. Terms in this formula apply to *all the different data types*. Specification of this formula parallels almost precisely as one would do for any glm-like modelling process – see the details section from `?glm` for more information about forms accepted. This specification allows a wide variety of models to be fitted: linear, curvi-linear (e.g. quadratics), interactions, regression splines, and so on. The formula will take the environmental variables from the covariate stack, and so the terms in the formula must be present there. Currently, the formula will not work with (or at least have unknown, untested behaviour) with factor raster layers – please be wary when using them and consider expanding them into a set of 'dummy' raster layers that indicate the various levels of the factor.

Like the `glm` function, the distributionFormula argument can be specified using a saved formula object or a newly specified formula object. It is important to note that, *un*like the `glm` function, there is no left-hand side to the formula – there are multiple sources of outcomes and they are hence specified in another argument. Another important aspect of specifying the distribution formula is that *any intercept is stripped out* and to help remind users they are recommended to explicitly add a '0' or a '-1' into the formula. In the ISDM, there are mutliple intercepts and are specified for each data type (at least).

To give some concrete examples about what formulas can be considered in the distribution model, consider these. The last one is the one that is used in Foster *et al.* (in review) and is the one that will be continued with here.

```
#linear in SMRZ only
my.form <- ~0+SMRZ
#interacting between SMRZ and DEM
my.form <- ~0+SMRZ*DEM
#a B-spline regression basis with low degrees of freedom
my.form <- ~0+splines::bs(SMRZ, df=3)
#a(n orthogonal) polynomial in SMRZ and DEM
my.form <- ~0+poly(SMRZ,2)+poly(DEM,2)
#adding accessibility too (as per paper).
my.form <- ~0+poly(SMRZ,2)+poly(DEM,2)+acc
```

The default action for all covariates in the distribution formula is that they are standardised. This means that they are transformed (linearly – subtracating mean and deviding by standard deviation), so that the covariate will have zero mean and standard deviation one. This also applies for basis-exansion covariates (like those from `poly` or bs) – we have found that the scaling for `poly` is too small. This does not effect the fit of the model but it does improve the numerical stability and has the side effect of altering the scale of the associated model parameter. In turn, this side-effect can make it easier to specify priors for the covariate

parameters – see for details about how this is done. This behaviour can be 'turned off' by using the `standardiseCovariates` element of the control argument to `isdm`.

One thing that may, at first, seem like an odd omission from the distribution formula is the spatial random effect. This is *never included*. Rather it is included as part of the machinations of `isdm` (default behaviour) or is not included by explicit request. This is controlled by the `addRandom` element of the control argument to `isdm`.

It is difficult to indicate what a 'good' and a 'bad' distribution formula will look like. It depends solely on the particular species and the covariates that are available. However, analysts should consider elementary pathological cases usually exemplified with linear models. These include avoiding over-fitting. For more basic advice on specification, and some rememdies to common problems please see Neter *et al.* (1996) or any introductory textbook focussing on linear models.

Observation Bias (for PO data)

The biasFormula argument of `isdm` specifies how the point pattern that generates the PO data *differs* from the species distribution point pattern. The biasFormula affects *only the PO data* and specifies the model component for the sampling bias for the PO data (see Warton *et al.* 2013, for example). The sampling bias arises from spatially non-uniform sampling effort over the study region. An example of why this might happen is, when PO data is citizen science based, presences are likely to be denser where there are more people available to look for them (ie near population centres). This sampling bias pattern is obscuring the true species distribution and hence should be removed from predictions, if possible. The ISDM allows for this by inspecting the (unbiased) patterns in the PA and AA data, and comparing to the pattern in the PO data.

Specifying the biasFormula argument mirrors that for specifying the distributionFormula argument. It is any formula object which takes data from the raster stack. It is limited by the same limitations and transformations as the distributionFormula (see Section). However, note that an intercept is needed – it is the intercept for the PO point process. If removed, in the biasFormula argument, then `isdm` will assume that it is intended and not fit an intercept. An intercept is included by default, just like `glm`.

The biasFormula argument should contain terms that are likely to affect the anathropegenic search pattern. Often these will include variables like 'travel time to location' and 'search intensity of other species'. However, specifics will depend on the data available and the particular system being modelled.

In the gamba grass example, Foster *et al.* (in review) used a simple structure based on accessibility. There is no estimability issue with including the term in both the biasFormula and in the distributionFormula – the latter is based on both PO and PA/AA data, while the former is based only on the PO data. However, in the particular case when there is only PO data, then the same covariate should not be included in both formulas.

Here are some examples of bias formulas for the gamba grass example. We take the last, simple formula to progress with.

```
#intercept only == no heterogeneity in search effort
my.biasForm <- 1
#regression spline in acc
```

```
my.biasForm <- ~1+splines::bs( acc, df=3)
#linear in acc
my.biasForm <- ~1+acc
```

Sampling Artefacts (for AA and PA data)

The PA and AA data may contain sources of variation that are not a result of species distribution. This is in spite of the assumption that the PA and AA data are the result of a well-planned survey effort. Such variation may arise from (but not limited to): 1) different measurement tools/procedures, 2) blocking or stratification, or 3) seasons or weather. In fact, any source of variation that may cause extraneous variation in the data can be modelled here. We call these sources of variation *sampling artefacts* as they are due to the sampling process and are generally not of primary interest.

There should be a sampling artefact formula for each of the data types included in the model. However, the PO sampling artefact is nonsensical and is ignored. Sampling artefacts *only affect those data types that they are defined for* and are in addition to the distribution formula for these observations. The formulae follow the same general rules as all formulas except that we currently *do not* recommend the use of basis expansion within the formula itself. If the user wants to use basis expansion, then they should create a temporary design matrix object, that includes the basis expanded variables, prior to calling `isdm` and use those names instead. Of course, there are exceptions to this rule, but they have not been tested and hence it is safer just to follow the pre-expansion principle.

Once again, there is no restriction about what varaibles are included in the artefactFormulas. Variables can appear in here that also appear in the distributionFormula (for example). However, the user must seriously ask themselves 'why'? Why would the variation within the PA/AA data be dependent on that variable *after the distribution has been conditioned out*?

For the gamba grass example, there is only one source of PA data (and no AA data). The PA data all come from a single data source, which were all measured using the same sampling tool (an aerial survey). For this reason, it is likely to be reasonable to assume that there are no sampling artefacts within these PA data. This is the assumption made in Foster *et al.* (in review) and is the assumption that we move forward with. However, the data also contain the observer that took the measurement. Different observers may see/perceive/report differently and so we present their inclusion an (untested) alternative for instructional purposes.

```
#observer differences as a sampling artefact
list( PA=~1+observer)

$PA
~1 + observer

#intercept only == no sampling artefact
#               == no heterogeneity within each data type
list( PA=~1)

$PA
~1
```

Bringing it All Together: an initial model estimation

We now have enough pieces to estimate the model. This will rely on default parameters, which could be dangerous in some situations. Nevertheless, this is done to demonstrate the mechanics of the estimation. It is stressed that this is a preliminary model estimate – wihtout due consideration to priors (especially those of the spatial random effect) the results may be 'odd'. Even though it is not explicitly stated, this model contains a spatial random effect in the distribution formula. This term is considered fundamental enough, that it has to be explicitly removed (instructions later).

The components that make up this model estimate have nearly all been introduced before. The exceptions are those that specify which variable should be used for what in which data set. That was a confusing sentence, so we'll break it down.

The responseNames argument lists what variables should be used for outcomes for each of the data types. It is a text label. Here, it specifies that the variable "PA" should be used from the data.frame `gamba_PA`, which is specified in observationList$PAdat.

The sampleAreaNames argument informs which variable within each of the PA and AA data.frames give the sample area. As stated before the sample area is the area that the sampling covered. For the gamba PA data, the sampling area was constant, and held in the un-creatively named variable "Area". There is no requirement for the PO data to have an area associated with it (these are assumed to be a point).

Finally, `isdm` must be told what the coordinates are called. These must be the same for all data structures. It is specified through the control$coord.names argument.

```
fm <- isdm( observationList=list( POdat=gamba_PO,
                                  PAdat=gamba_PA),
            covars=covars,
            mesh=my.mesh,
            responseNames=c( PO=NULL, PA="PA"),
            sampleAreaNames=c( PO=NULL, PA="Area"),
            distributionFormula=~0+poly( DEM, 2) + poly( SMRZ,2) + ACC,
            biasFormula=~1+ACC,
            artefactFormulas=list( PA=~1),
            control=list( coord.names=c("x","y")))
```

The return from `isdm` is silent, so there is nothing special to see upon its successful completion. Given this initial fit, further refinements of the model can be demonstrated.

Priors for Model Effects

All Bayesian analyses require the user to specify a prior distribution, and this ISDM is no different. In `RISDM`, we have focussed on ease of specification rather than full flexibility. We focus on specifying vague priors and separate only priors for intercepts and priors for covariate effects. We envisage that most users will specify models with vague priors, reflecting little or now prior knowledge, which fits easily into this format.

The priors are specified via the control list argument and describe Gaussian distributions. In particular, the priors are specified through the prior.mean element and the prior.sd (or

prior.prec) arguments. The prior.mean is a single scalar giving the prior expectation of *all* the model's effects. This includes all intercepts and all covariate effects for all the data types. Almost always, prior.mean will be zero, indicating that the effect has a priori equal chance of being positive or negative.

The prior variation for the covariate effects are split into two types: those for intercepts and those for covariate effects. We prefer to specify prior variation through standard deviation, but also allow for precision to be specified (inversely related). These variances are specified through the `int.sd` and `int.prec` elements of the control list. By default the sd for the intercepts is very large (1000) and that for the covariate effects is large (10). We note that if a standard deviation is specified then the precision is ignored.

Remember, from Section that all covariates are scaled prior to fitting the model. This means that the effects are *in some sense* on the same numerical scale. In turn this implies that it may be justifiable, or at least practical, to specify this common prior for effects. If a particular application requires a different approach then the priors will have to be specified directly.

If the user wants to specify priors directly, then they can by specifying prior.list element of the control list directly. This argument is *passed directly* to the INLA call, and so the specification is solely the responsibility of the user. INLA requires that these are passed as a list of two named elements: mean and prec. Each element contains the named vector of values for the prior. Note that only those effects that are desired to be *different* from the default are required to be named and specified (I think).

Here are some control list examples that specifies various prior formulations. Note that these specify only those control elements for the priors; other terms may be added to this list to control other aspects of the model or the computational aspects.

```
#the very vague everything
my.control <- list( prior.mean=0, int.sd=1000, other.sd=1000)
#(much) tighter prior for effects
my.control <- list( prior.mean=0, int.sd=1000, other.sd=0.1)
#the default
my.control <- list( prior.mean=0, int.sd=1000, other.sd=10)
```

Spatial Effects via SPDE

The `RISDM` model, by default, includes a spatial random effect. This spatial effect is defined according to the SPDE approach defined in Lindgren *et al.* (2011) and practical aspects discussed in Lindgren and Rue (2015) and is included for 2 purposes: 1) to induce a spatial autocorrelation into the observations (the random effect is spatially correlated); and 2) to explain spatially patterned random variation (noise) that has not been explained by the environmental covariates.

The specification of the distribution of the random effects depends on two parameters: the standard deviation of the effects, and the spatial range of their dependence. The spatial range is approximately the distance that samples must be separated by before data from those locations are expected to be almost independent (correlation of approximately 0.1). Both these parameters require a prior distribution to be specified, and `RISDM`, like `INLA`, follows Simpson *et al.* (2017) by using complexity priors. These priors are defined by defining

the prior chance that the parameter falls above (for prior standard deviation) or below (for spatial range) specified values. For the gamba grass example, it may be reasonable that the standard deviation has probability of 0.1 (10% chance) of being above 5. We feel that this is a vague prior for random effects on a log-link scale. Likewise, a vague prior for spatial dependence could be that there is a probability of 0.1 that the range is less than 1km. The previous initial model can be updated using these priors, giving the model in Foster *et al.* (in review).

```
fm <- isdm( observationList=list( POdat=gamba_PO,
                                  PAdat=gamba_PA),
            covars=covars,
            mesh=my.mesh,
            responseNames=c( PO=NULL, PA="PA"),
            sampleAreaNames=c( PO=NULL, PA="Area"),
            distributionFormula=~0+poly( DEM, 2) + poly( SMRZ,2) + ACC,
            biasFormula=~1+ACC,
            artefactFormulas=list( PA=~1),
            control=list( coord.names=c("x","y"),
                          int.sd=1000, other.sd=10, prior.mean=0,
                          prior.range=c(1,0.1), prior.space.sigma=c( 5,0.1)))
```

The addition of the spatial random effect can be suppressed by the control list element. That is by including addRandom=FALSE into the control=list(...) argument. Code below. The model will be much (!) faster to estimate. We'll return to this model later in the vignette.

```
fm.noRand <- isdm( observationList=list( POdat=gamba_PO,
                                         PAdat=gamba_PA),
                   covars=covars,
                   mesh=my.mesh,
                   responseNames=c( PO=NULL, PA="PA"),
                   sampleAreaNames=c( PO=NULL, PA="Area"),
                   distributionFormula=~0+poly( DEM, 2) + poly( SMRZ,2) + ACC,
                   biasFormula=~1+ACC,
                   artefactFormulas=list( PA=~1),
                   control=list( coord.names=c("x","y"),
                                 int.sd=1000, other.sd=10, prior.mean=0,
                                 addRandom=FALSE))
```

Diagnostics and Summary

To understand how the model has performed, and to check that things have gone as could reasonably be expected, it is important to both check the model for obvious (large) departures from assumptions and to check if the estimated model passes a 'laugh test'. Both these checks have been performed in statstical sciences for a long time, but perhaps only the diagnostic formed by checking for departures is formally defined anywhere. Tools for both approaches will be given now.

Summary Tables

To obtain an elementary understanding of the estimated model, a `summary` method is provided. This, like the method for `glm`, gives information about the estimated parameters and a statistic about the model itself. Unlike a `glm` summary method, the `isdm` summary method will not print the model call itself as it is generally quite long, cryptic and boring. If users want to see the `INLA` call then they can, as part of the return object (fmmodcall in this case).

The printed summary object shows univariate summaries of the modelled parameters' posterior distributions. These are arranged into the type of effect (distribution, bias, artefact) for ease of interpretation. In this case, we see that both quadratic terms in the distribution formula poly.DEM.2.2 and poly.SMRZ.2.2 are negative and with credible intervals that do not cover zero. This implies that both relationships are concave in shape, over the observed covariate ranges. This is reassuring as it aggrees somewhat with niche theory, but it is not the final word (linear effect may still outweigh the quadratic). The accessibility covariate is strongly negative, implying that those locations that are remote (high accessibility values) are less likely to have gamba grass. The other model components can be likewise interpreted.

```
summary( fm)

$DISTRIBUTION
                      mean         sd    0.025quant    0.5quant 0.975quant
poly.DEM.2.1   0.23898284 0.12155316 -0.0008500752  0.23948562  0.4759722
poly.DEM.2.2  -0.39567311 0.08560622 -0.5635819921 -0.39567213 -0.2277700
poly.SMRZ.2.1  0.03762888 0.24846406 -0.4561822628  0.03957806  0.5204454
poly.SMRZ.2.2 -0.56086572 0.20891204 -0.9757261022 -0.55928315 -0.1549694
ACC           -1.93507649 0.20521488 -2.3398858606 -1.93425913 -1.5348900

$PO_BIAS
                  mean        sd 0.025quant  0.5quant 0.975quant
PO_Intercept -3.378854 0.2483792  -3.894270 -3.369568 -2.9173449
PO_ACC       -1.141882 0.1378065  -1.412156 -1.141877 -0.8716358

$PA_ARTEFACT
                  mean        sd 0.025quant 0.5quant 0.975quant
PA_Intercept -14.09951 0.2767023   -14.6647 -14.0921  -13.57715

$SPATIAL
                            mean         sd 0.025quant 0.5quant 0.975quant
Range for isdm.spat.XXX 3.050353 0.37127935   2.417180 3.015977   3.876085
Stdev for isdm.spat.XXX 2.269714 0.09543609   2.086649 2.267873   2.463519

$marg.lik
[1] -3868.7

attr(,"class")
[1] "summary.isdm"
```

Model Comparison

The `summary.isdm` method returns, as a member of the list, an INLA estimate of the marginal likelihood. This is sometimes referred to as 'model evidence'. It can be used to compare models, in particular using Bayes Factors. For further information, the reader is referred to Kass and Raftery (1995) but also see Gelman *et al.* (2013).

Residual Plots

Residuals give an analyst a view of the data through the lens of the model. As such, they can help identify inadequacies from tenuous model assumptions. Residuals *can* even help identify possible remedies to inadequacies (e.g. Neter *et al.* 1996).

In linear models (e.g. multiple linear regressions) residuals have a natural definition – the observation minus the model prediction. This works in that case because those deviations are assumed by the model to be normally distributed (and homoscedastic) and there are no random effects, apart from residuals, in the model. In general use, residuals do not incorporate the variability in the model prediction. In a full Bayesian treatment, this extra variability would be included.

In `RISDM`, randomised quantile residuals (RQR; Dunn and Smyth 1996) are used. These residuals have a number of desireable features. Most notably, their distribution (assuming model assumptions are correct) is known, and is common for all different data types. This is in contrast to, say, Pearson or deviance residuals whose distribution is unknown for a finite sample (Dunn and Smyth 1996). The RQR residuals incorporate only the variation from the final fitted value (including the spatial random effect) to the observation. This mirrors the residual treatment in the popular `mgcv` package (Wood 2017), even if `mgcv` uses different types of residuals. If the model is adequately specified, then the RQR should be normally distributed and homoscedastic – like residuals from a multiple linear regression. Departures from normality, or patterns along environmental gradients or through space may indicate problems with the model.

In `RISDM` there are multiple sets of residuals – one for each data type. All these residuals can be calculated using the `residuals.isdm` method. Most often, the user will want to plot these residuals and this can be done in an automated way using the `plot.isdm` method, in ways that may be useful for diagnosing inadequacies. These mirror the treatment in introductory textbooks on linear models (e.g. Neter *et al.* 1996). Resdiuals are plotted per data type using a residuals versus fitted and a quantile-quantile (Q-Q) plot. There should be no patterning in the residuals versus fitted plot, in either mean pattern or in residual variation. The following plot for PA residuals shows no cause for concern here, and the PO residuals are harder to interpret. The Q-Q plot should follow a straight line. The following plot seems fine for PA data, but questionable for PO data. `RISDM` will also plot the PO RQR residuals as a spatial layer. Like any covariate, there should be no patterning and its presence could indicate deficiencies.

```
plot( fm, covars=covars, nFigRow=2, ask=FALSE)
```

One open research questions around residuals is trying to determine how large the departures need to be before the model needs altering. This will depend on many different aspects, including the type of inference sought. Hence, it is not clear if the residuals in Figure 2

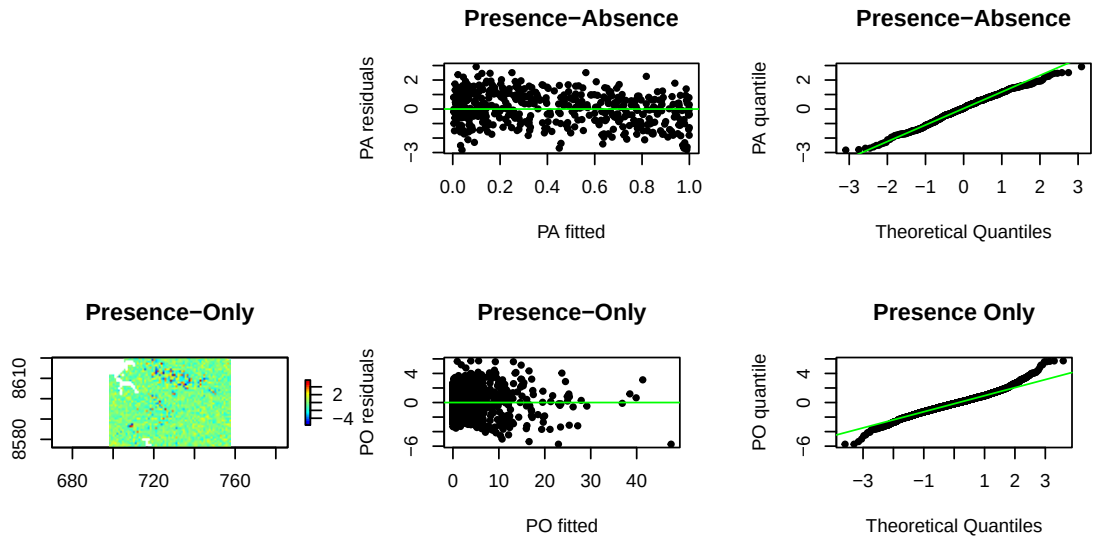


Figure 2: Residual plots for the gamba grass data, and the model containing quadratic effects and a spatial term.

indicate an adequate model or not. However, it is possible (and easy) to demonstrate a model with *a worse fit*. To do this, we revisit the model but exclude the spatial random effects. The resulting residual plots are in Figure 3. The PA data still seem adequately represented but the patterns in the PO data are accentuated and are now obviously worrisome. There are strong spatial patterns and the residuals are obviously non-normal.

```
plot( fm.noRand, covars=covars, nFigRow=2, ask=FALSE)
```

One of the potential short-comings of RQR residuals is that they contain a random component. This component is largest for data types that have few categories dichotomous data, like PA data, are subject to the largest amount possible. This fact may contribute to the apparent adequate residual plots for the model without spatial random effects.

Prediction

Predictions from the fitted model are performed using the `predict.isdm` method. This method proceeds by taking posterior draws of the parameters, using `INLA`'s `inla.posterior.sample` function, and then predicting using the sampled parameters into a user-supplied raster stack. In this routine, the spatial random effects are treated the same as the parameters. This is demonstrated using both the model with (and without) spatial random effects.

One choice that the user should make is which intercept should be included in the model. This choice has a real consequence in the predictions as some datasets (or even data types) will naturally have higher/lower intercepts due to tool efficacy and possibly just by measuring the geographical area that it did.

By default the `predict` method will predict on the intensity scale. This gives the expected number of individuals within a raster cell at each of the raster's locations. However, this

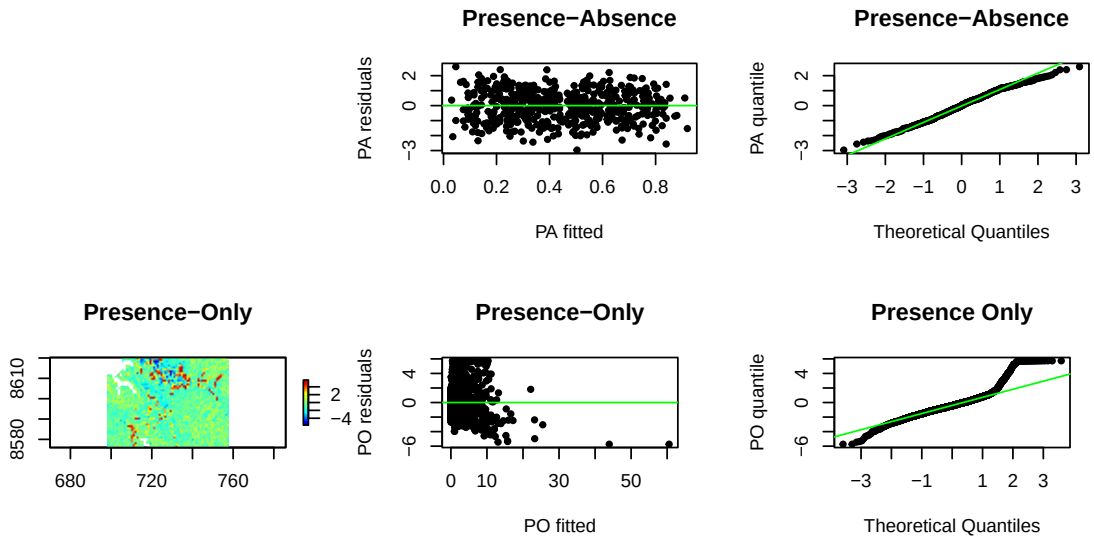


Figure 3: Residual plots for the gamba grass data, and the model containing quadratic effects extitbut no spatial term.

is not the only choice and the user can choose to predict the probability of presence or the linear predictor. Probability is a non-linear transformation of the intensity and, as such, is dependent upon the intercept value chosen. An intercept that is abnormally high will produce values that are too close to one. Likewise, an intercept that is low will have deflated probability. Care should be taken and the choice should be reasoned and well-considered.

The intensity and probability predictions in Figure 4 appear to be very similar. They are. This is because the intensity and probability are very small, and when they are very small the non-linear transformation between them turns out to be very well approximated by the identity function.

The `predict` method outputs a list with a number of elements. Arguably, the most useful of these is the 'field' element, which contains a raster stack with a number of summaries of the prediction posterior. Most notably, the prediction stack contains the median prediction and the 2.5% percentile and the 97.5% percentile. These values can be taken to be the point and the interval estimate for the predicted property. Also included in the prediction stack is the posterior mean and the posterior standard deviation. It has been the authors experience that the median is a more robust summary of central location in this situation, and it is recommended.

```
#You should use a much(!) larger value of S.
#You may want to choose a larger value of n.threads too.
fm$preds <- predict( fm, covars=covars, S=50,
                        intercept.terms="PA_Intercept")
plot( fm$preds$field)

#Predicting probability too
fm$preds.probs <- predict( fm, covars=covars,
```

```
                            S=50, intercept.terms="PA_Intercept",
                            type="probability")
plot( fm$preds.probs$field)
```

By default, the `predict.isdm` method includes the terms from the distribution function (including spatial random effect) and whatever intercept is chosen. However, variations on this can be achieved:

- The inclusion of the random effect can be switched on and off by using the 'includeRandom' argument.
- The inclusion of the fixed effect component of the linear predictor can be switched on/off by using the 'includeFixed=FALSE' argument.
- The terms in the sampling bias model component (not included by default) can be included or not using the 'includeBias' argument.

Combinations of these three arguments will give the user any combination of effect types that they want. This option is included to inspect the relative contribution from the terms, for understanding, education and possibly diagnosis. As yet, it is unclear if there is any inferential use.

Interpretation

Interpretation of any non-experimental (e.g. survey and unplanned) data requires care. This is because the inferences may change depending on what assumptions the analyst makes, and what data (covariates) are available. Just one example: if a key covariate is not measured, or is measured with high variance, then the effect of that covariate may be erroneously partitioned to other sources. Nevertheless, in the hope (and assumption) that these types of scenarios are the minority, it can be useful to try and understand how the model is working. Our opinion is that these procedures should be considered as 'hypothesis generating' rather than hard evidence for any particular theory. The reason is that there could be multiple pathways to the result. Just one of the probably less-interesting pathways is that the result depends just on random chance in the data.

Interpreting the model involves inspecting the posterior distribution of the model's parameters. When the terms are linear, this is pretty simple to do. However, when there is some sort of basis expansion in the formula (e.g. quadratic or B-spline) then it becomes more difficult. One solution (not used here) is to pre-compute all basis expansions before fitting the model. The responses can then be constructed using the posterior distributions. Another option is to let `predict.isdm` do the work – we will use this now. Note that the data, for the covariate of interest, must be exactly the same as the data used in the model estimation. If this is not the case, then there will be a different basis expansion envoked which will create a different relationship (not the one modelled).

The covariate effects for gamba grass are presented in Figure 5. Note that there is a bit of coding involved; there is no function within `RISDM` to do this directly and the predict function must be 'appropriately tricked'. Besides plotting nearly always takes some extra lines of code. Whilst details are important, the process is actually fairly straight-forward: 1) create a raster

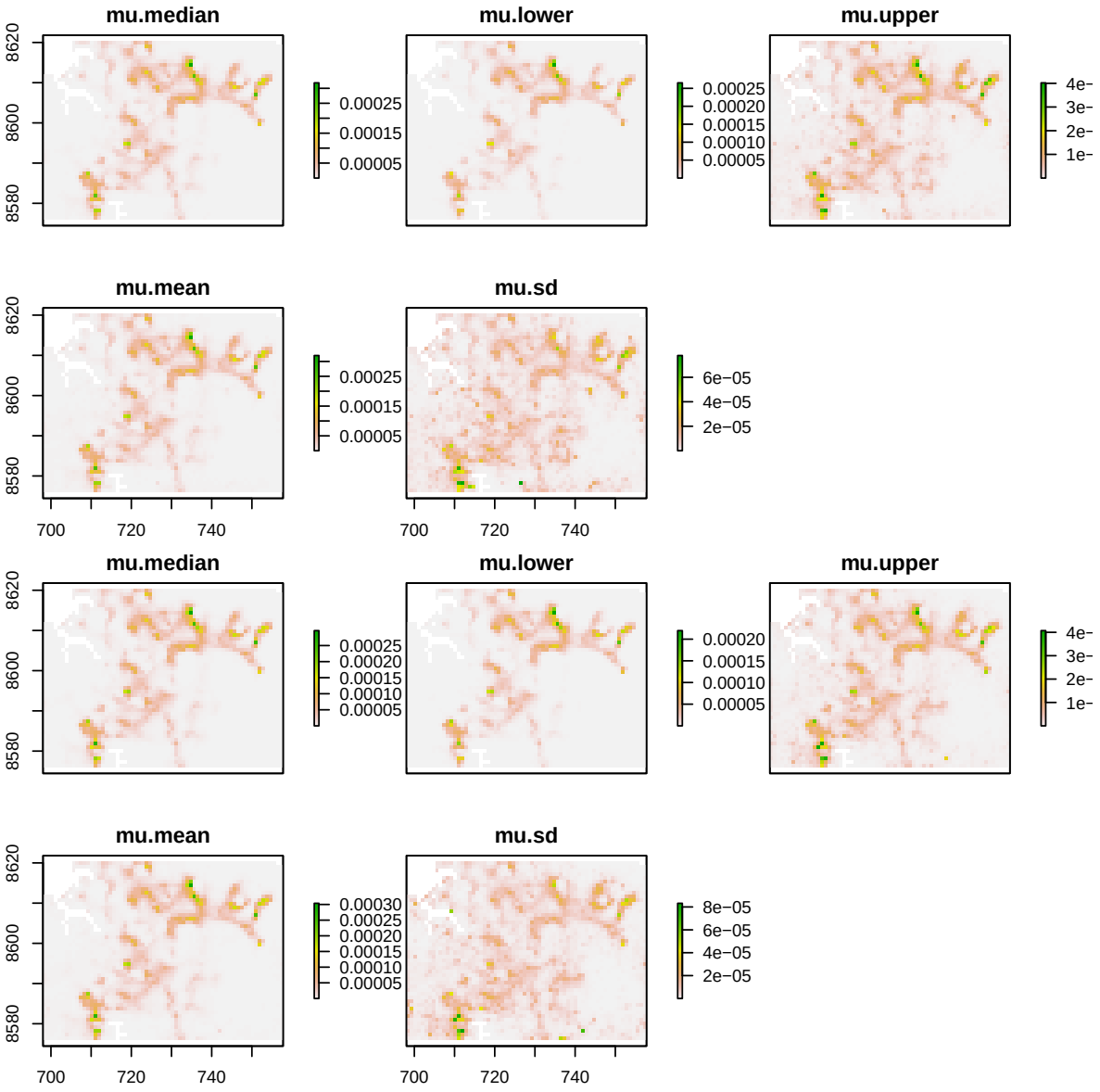


Figure 4: Predictions from the quadratic model including random effects. Upper two rows are for the intensity, and bottom two rows are predictions of the probability of presence (within a raster cell)

layer with a constant value for the habitatArea offset, 2) predict specifying which terms should be included, and 3) plot)

```
#the data for interpretation
#adding a temporary cell area layer
covarsForInter <- addLayer( covars, covars[[1]])
names( covarsForInter) <- c( names( covars), "tmp.habiArea")
#area is now constant with log(1)=0 contribution
values( covarsForInter$tmp.habiArea) <- 1

#You could use a much(!) larger value of S.
interpPreds <- predict( fm, covars=covarsForInter,
                        habitatArea="tmp.habiArea", S=50,
                        includeFixed="SMRZ", includeRandom=FALSE, type="link")

#compile covariate and prediction
pred.df <- as.data.frame( cbind( SMRZ=values( covars$SMRZ),
     values( interpPreds$field[[c("mu.median","mu.lower","mu.upper")]])))
#plot
pred.df <- pred.df[!is.na( pred.df$SMRZ),]
pred.df <- pred.df[order( pred.df$SMRZ),]
matplot( pred.df[,1], pred.df[,2:4], pch="", xlab="SMRZ", ylab="Effect",
                                                  main="Effect plot for SMRZ")
polygon( x=c( pred.df$SMRZ, rev( pred.df$SMRZ)),
            c(pred.df$mu.upper, rev(pred.df$mu.lower)),
                                                    col=grey(0.95), bor=NA)
lines( pred.df[,c("SMRZ","mu.median")], type='l', lwd=2)
```

DIY Extensions

Whilst flexible, the `RISDM` package, and in particular the `isdm` function, do not cover the full dazzling variety of models that can be fitted using `INLA`. This is intentional as it keeps the syntax cleaner and learning curve less step. However, many extensions of the ISDM approach will require solving the same set of problems that `RISDM` handles. For this reason, advanced users may wish to use `RISDM` as a component for their workflow. `RISDM` will handle building the `INLA` stacks for multiple data sources and building the multiple likelihood structures.

To enable this, the `isdm` function returns some useful data objects. Top of this list is the `INLA` data stack, held in the 'stack' element of the object list. Also useful is the return of the `INLA` object itself, held in the 'mod' object of the object list. These are accessed from, for example, `fm$stack` and `fm$mod`.

Single Data Types

Whilst created with multiple data types in mind, `RISDM` will quite happily estimate models based on single data types too. We'll do this now for a bit of comparative 'fun'. The fits produce numerical issues (see output), which should be investigated and corrected for in

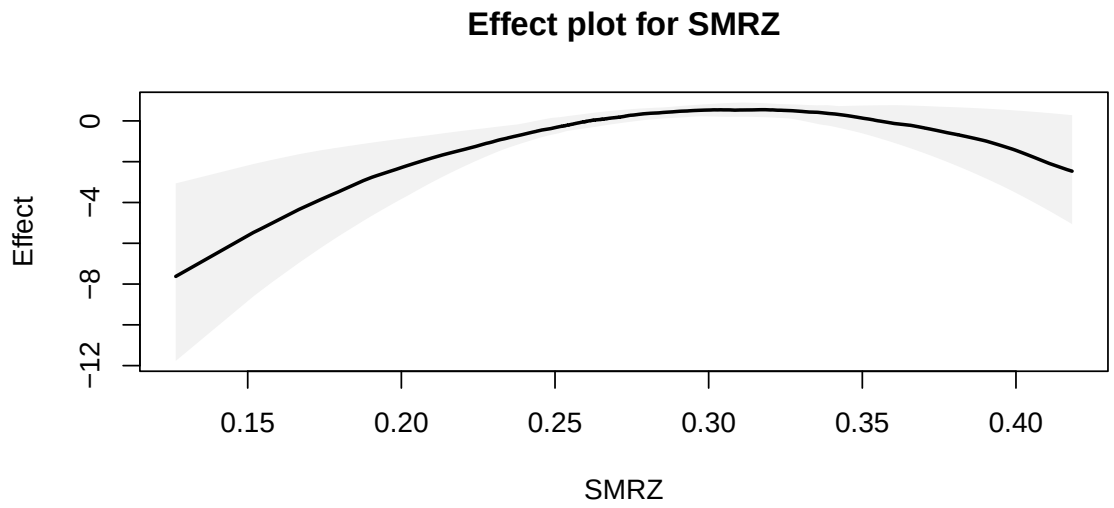


Figure 5: Relationship with Soil Moisture (SMRZ). Black solid line is the median relationship and grey shaded area is the 95 percent CI.

a real analysis setting. However, to demonstrate the mechanics of fitting single data type models they are *almost* ignorable. In part, their existance supports the thesis that combining data can produce more information and better models than any single species model. If the warnings weren't present, then these models and predictions could be compared against the integrated model (e.g. Figure 4).

```
#PO data only
fm.PO <- isdm( observationList=list( POdat=gamba_PO),
            covars=covars,
            mesh=my.mesh,
            responseNames=NULL,
            sampleAreaNames=NULL,
            distributionFormula=~0+poly( DEM, 2) + poly( SMRZ,2) + ACC,
            biasFormula=~1+ACC,
            artefactFormulas=NULL,
            control=list( coord.names=c("x","y"),
                          int.sd=1000, other.sd=10, prior.mean=0,
                          prior.range=c(1,0.1), prior.space.sigma=c( 5,0.1)))
fm.PO$preds <- predict( fm.PO, covars=covars, S=50,
                        intercept.terms="PO_Intercept")
plot( fm.PO$preds$field)
```

```
#PA data only
fm.PA <- isdm( observationList=list( PAdat=gamba_PA),
            covars=covars,
```

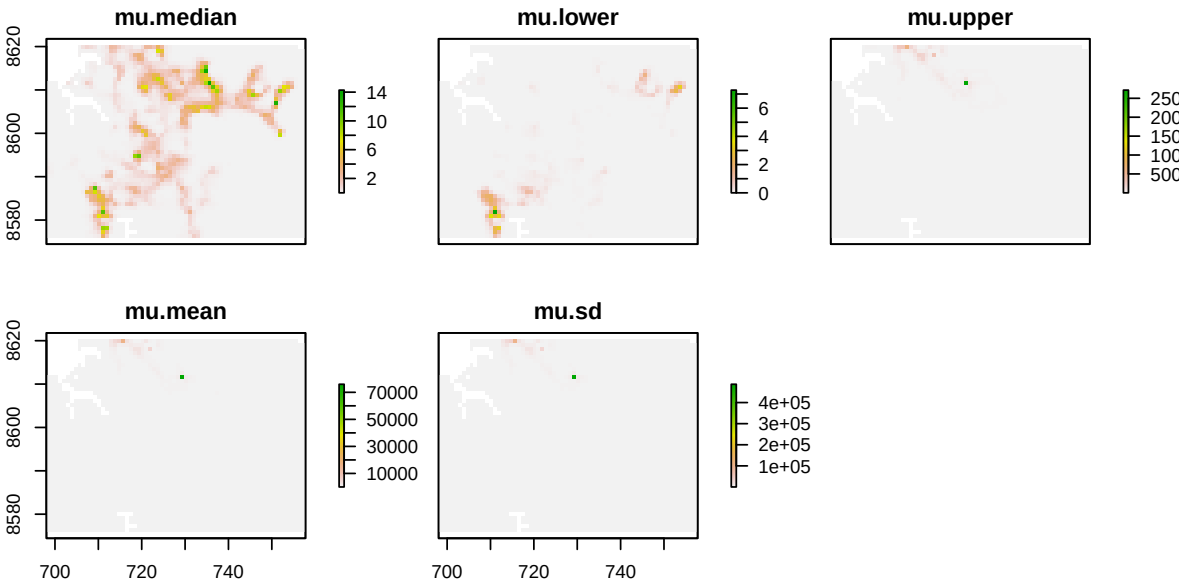


Figure 6: Intensity model and predictions from estimation using only PA data.

```
            mesh=my.mesh,
            responseNames=c( PA="PA"),
            sampleAreaNames=c( PA="Area"),
            distributionFormula=~0+poly( DEM, 2) + poly( SMRZ,2) + ACC,
            artefactFormulas=list( PA=~1),
            control=list( coord.names=c("x","y"),
                          int.sd=1000, other.sd=10, prior.mean=0,
                          prior.range=c(1,0.1), prior.space.sigma=c( 5,0.1)))

Error in inla.inlaprogram.has.crashed() :
  The inla-program exited with an error. Unless you interupted it yourself, please rerun w
  If this does not help, please contact the developers at <help@r-inla.org>.

 *** inla.core.safe:  inla.program has crashed: rerun to get better initial values. try=1/

 *** inla.core.safe:  rerun with improved initial values

fm.PA$preds <- predict( fm.PA, covars=covars, S=50,
                        intercept.terms="PA_Intercept")
plot( fm.PA$preds$field)
```

Last Things Last

The only remaining thing to do is to tidy up our workspace. This is just removing all objects for this analysis from your workspace. I like to do this, in tutorial situations, but you may

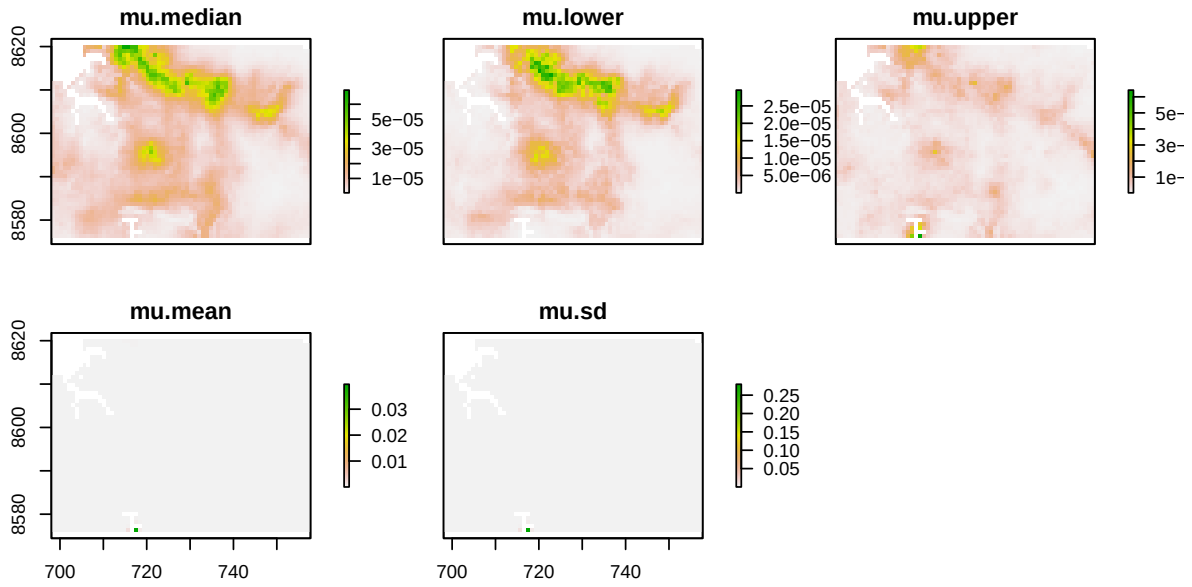


Figure 7: Intensity model and predictions from estimation using only PA data.

not. It is entirely up to you whether you clean or not.

```
#You may wish to tidy your workspace.
rm( covars, fm, fm.noRand, fm.PA, fm.PO, gamba_PA, gamba_PO, filenames,
              my.biasForm, my.form, my.control, my.mesh, my.mesh.bad)
```

Acknowledgements

This work was partially funded by the Australian Government Department of Agriculture, Fisheries and Forestry's Established Pest Animals and Weeds Management Pipeline Program and Supporting Communities Manage Pests and Weeds Program.

References

Adams VM, Petty AM, Douglas MM, Buckley YM, Ferdinands KB, Okazaki T, Ko DW, Setterfield SA (2015). "Distribution, demography and dispersal model of spatial spread of invasive plant populations with limited data." *Methods in Ecology and Evolution*, **6**(7), 782–794. doi:https://doi.org/10.1111/2041-210X.12392.

Bakka H, Rue H, Fuglstad GA, Riebler A, Bolin D, Illian J, Krainski E, Simpson D, Lindgren F (2018). "Spatial modeling with R-INLA: A review." *WIREs Computational Statistics*, **10**(6), e1443. doi:https://doi.org/10.1002/wics.1443.

Dunn PK, Smyth GK (1996). "Randomized Quantile Residuals." *Journal of Computational and Graphical Statistics*, **5**(3), 236–244. doi:10.1080/10618600.1996.10474708.

Fletcher Jr RJ, Hefley TJ, Robertson EP, Zuckerberg B, McCleery RA, Dorazio RM (2019). "A practical guide for combining data to model species distributions." *Ecology*, **100**(6), e02710. doi:10.1002/ecy.2710.

Flores TA, Setterfield SA, Douglas MM (2005). "Seedling recruitment of the exotic grass Andropogon gayanus (Poaceae) in northern Australia." *Australian Journal of Botany*, **53**, 243–249.

Foster S, Peel D, Hosack G, Hoskins A, Mitchell D, Proft K, Froese J (in review). "RISDM: Species Distribution Modelling from Multiple Data Sources in R." *TBA*, **NA**, NA–NA.

Frost A, Ramchurn A, Smith A (2018). "The Australian landscape water balance model (AWRA-L v6). Technical description of the Australian water resources assessment landscape model version 6."

Gallant JC, Austin JM (2015). "Derivation of terrain covariates for digital soil mapping in Australia." *Soil Research*, (53), 895–906.

Gelman A, Carlin J, Stern H, Dunson D, Vehtari A, Rubin D (2013). *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. ISBN 9781439840955.

Geosciences Australia (2006). "GEODATA TOPO 250K Series 3 (Shape file format)."

Hijmans RJ (2022). *raster: Geographic Data Analysis and Modeling*. R package version 3.6-3, URL https://CRAN.R-project.org/package=raster.

Isaac NJ, Jarzyna MA, Keil P, Dambly LI, Boersch-Supan PH, Browning E, Freeman SN, Golding N, Guillera-Arroita G, Henrys PA, Jarvis S, Lahoz-Monfort J, Pagel J, Pescott OL, Schmucki R, Simmonds EG, O'Hara RB (2020). "Data Integration for Large-Scale Models of Species Distributions." *Trends in Ecology & Evolution*, **35**(1), 56 – 67. doi:https://doi.org/10.1016/j.tree.2019.08.006.

Kass RE, Raftery AE (1995). "Bayes Factors." *Journal of the American Statistical Association*, **90**(430), 773–795.

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

Lindgren F, Rue H (2015). "Bayesian Spatial Modelling with R-INLA." *Journal of Statistical Software, Articles*, **63**(19), 1–25. ISSN 1548-7660. doi:10.18637/jss.v063.i19.

Lindgren F, Rue H, Lindström J (2011). "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **73**(4), 423–498. doi:10.1111/j.1467-9868.2011.00777.x.

Miller DAW, Pacifici K, Sanderlin JS, Reich BJ (2019). "The recent past and promising future for data integration methods to estimate species' distributions." *Methods in Ecology and Evolution*, **10**(1), 22–37. doi:10.1111/2041-210X.13110.

Neter J, Kutner MH, Nachtsheim CJ, Wasserman W (1996). *Applied Linear Statistical Models*. Irwin, Chicago.

Petty AM, Setterfield SA, Ferdinands KB, Barrow P (2012). "Inferring habitat suitability and spread patterns from large-scale distributions of an exotic invasive pasture grass in north Australia." *Journal of Applied Ecology*, **49**(3), 742–752. doi:https://doi.org/10.1111/j.1365-2664.2012.02128.x.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Righetto A, Faes C, Vandendijck Y, Ribeiro Jr P (2020). "On the choice of the mesh for the analysis of geostatistical data using R-INLA." *Communications in Statistics - Theory and Methods*, **49**(1), 203–220. doi:10.1080/03610926.2018.1536209.

Rue H, Martino S, Chopin N (2009). "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**(2), 319–392. doi:10.1111/j.1467-9868.2008.00700.x.

Simpson DP, Rue H, Riebler A, Martins TG, Sørbye SH (2017). "Penalising model component complexity: A principled, practical approach to constructing priors." *Statistical Science*, **32**(1), 1–28.

Verdoy PJ (2021). "Enhancing the SPDE modeling of spatial point processes with INLA, applied to wildfires. Choosing the best mesh for each database." *Communications in Statistics - Simulation and Computation*, **50**(10), 2990–3030. doi:10.1080/03610918.2019.1618473.

Warton D, Renner I, Ramp D (2013). "Model-Based Control of Observer Bias for the Analysis of Presence-Only Data in Ecology." *PLoS ONE*, **8**(11), 1–9. doi:10.1371/journal.pone.0079168.

Weiss DJ, Nelson A, Gibson HS, Temperley W, Peedell S, Lieber A, Hancher M, Poyart E, Belchior S, Fullman N, Mappin B, Dalrymple U, Rozier J, Lucas TCD, Howes RE, Tusting LS, Kang SY, Cameron E, Bisanzio D, Battle KE, Bhatt S, Gething PW (2018). "A global map of travel time to cities to assess inequalities in accessibility in 2015." *Nature*, **553**, 333–336. doi:10.1038/nature25181.

Wood S (2017). *Generalized Additive Models: An Introduction with R*. 2 edition. Chapman and Hall/CRC.

Appendix

Computational details

This vignette was created using the following R and add-on package versions

- R version 4.2.1 (2022-06-23), `x86_64-pc-linux-gnu`

- Locale: `LC_CTYPE=en_AU.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_AU.UTF-8`, `LC_COLLATE=en_AU.UTF-8`, `LC_MONETARY=en_AU.UTF-8`, `LC_MESSAGES=en_AU.UTF-8`, `LC_PAPER=en_AU.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_AU.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 20.04.6 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0`
- LAPACK: `/usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: knitr 1.39, raster 3.6-3, RISDM 1.2.9, sp 1.6-0
- Loaded via a namespace (and not attached): brio 1.1.1, cachem 1.0.4, callr 3.6.0, cli 3.6.0, codetools 0.2-18, colorRamps 2.3, compiler 4.2.1, crayon 1.4.2, deldir 1.0-9, desc 1.4.1, devtools 2.4.3, digest 0.6.28, ellipsis 0.3.2, evaluate 0.15, fansi 1.0.4, fastmap 1.1.0, foreign 0.8-82, fs 1.5.0, glue 1.6.2, grid 4.2.1, highr 0.8, htmltools 0.5.2, INLA 23.04.24, lattice 0.20-45, lifecycle 1.0.3, magrittr 2.0.3, maptools 1.1-5, Matrix 1.5-3, MatrixModels 0.5-1, memoise 2.0.0, mnormt 2.0.2, numDeriv 2016.8-1.1, parallel 4.2.1, pillar 1.8.1, pkgbuild 1.2.0, pkgconfig 2.0.3, pkgload 1.2.4, prettyunits 1.1.1, processx 3.5.0, ps 1.6.0, purrr 1.0.1, R6 2.5.1, Rcpp 1.0.10, remotes 2.4.2, rgdal 1.5-32, rgeos 0.5-9, rlang 1.0.6, rmarkdown 2.14, rprojroot 2.0.2, rstudioapi 0.13, sessioninfo 1.1.1, sn 2.1.1, splancs 2.01-40, splines 4.2.1, stats4 4.2.1, stringi 1.5.3, stringr 1.4.0, terra 1.7-29, testthat 3.1.4, tibble 3.1.8, tmvnsim 1.0-2, tools 4.2.1, usethis 2.1.6, utf8 1.2.2, vctrs 0.5.2, withr 2.5.0, xfun 0.31

Affiliation:

Scott D. Foster
Data61, CSIRO
Marine Laboratories
GPObox 1538
Hobart 7001
Australia E-mail: scott.foster@data61.csiro.au