Flexible modelling with `inlabru`: A distance sampling case study

Andy Seaton

University of St Andrews

Acknowledgements

Prof Janine Illian, University of Glasgow

Prof David Borchers, University of St Andrews

Dr Richard Glennie, University of St Andrews

Prof Finn Lindgren, University of Edinburgh

Dr Fabian Bachl, University of Edinburgh

Dr Rick Camp, US Geological Survey

Dr David Miller, University of St Andrews

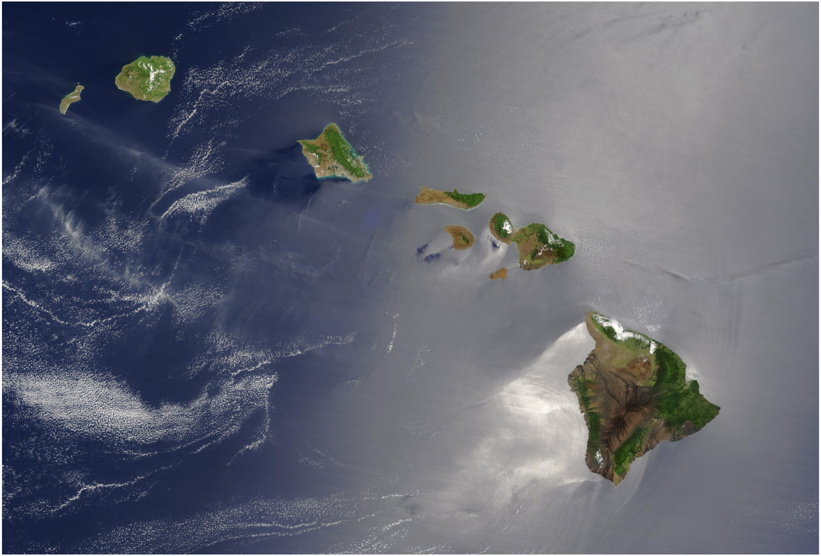
Aims of the talk

1. Demonstrate how to fit distance sampling models in `R-INLA`/`inlabru`.

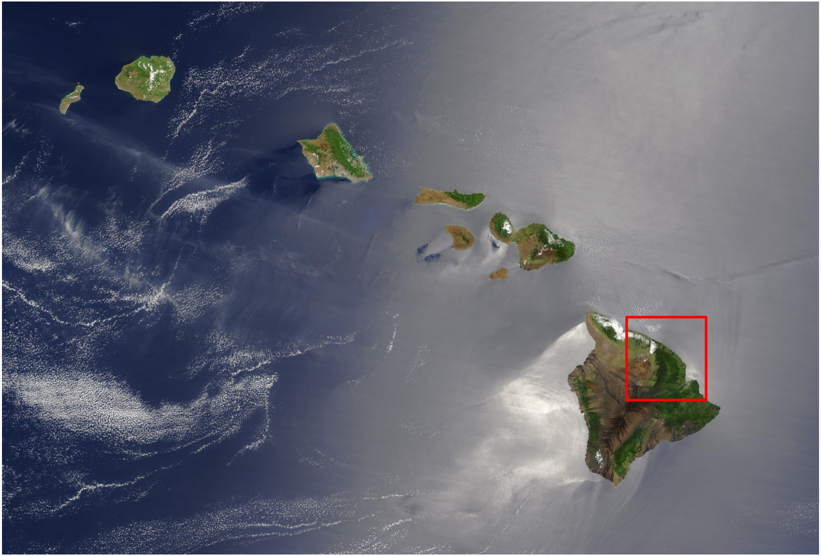
Aims of the talk

1. Demonstrate how to fit distance sampling models in `R-INLA`/`inlabru`.
2. Highlight *general* features of `inlabru` that are applicable in a wide range of contexts (not just distance sampling).

Akepa Case Study

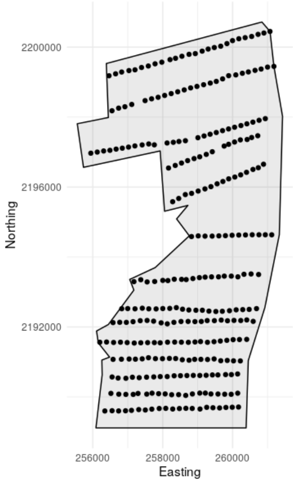


Akepa Case Study



Akepa Case Study

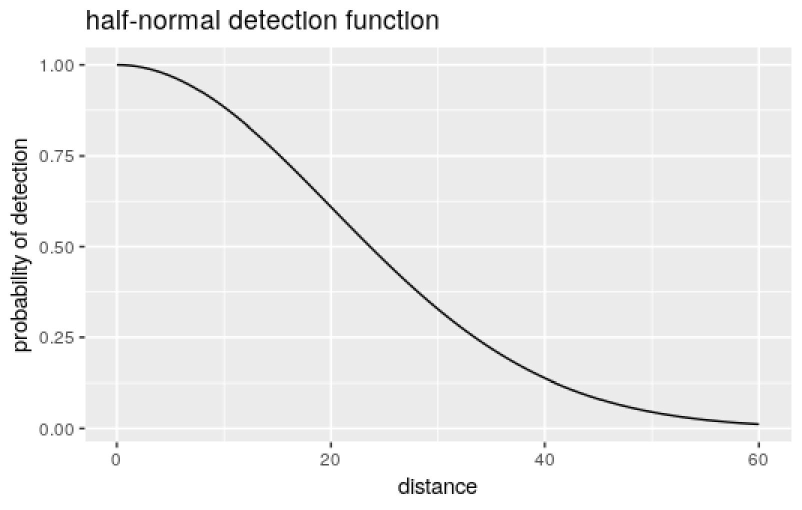
Hawaii Forest Bird Survey transect locations



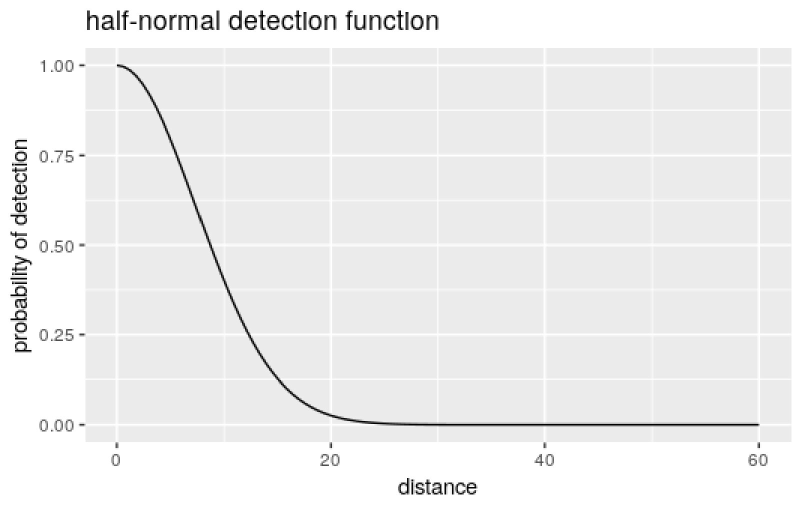
Akepa Case Study

source: Jack Jeffrey, US Fish and Wildlife Service

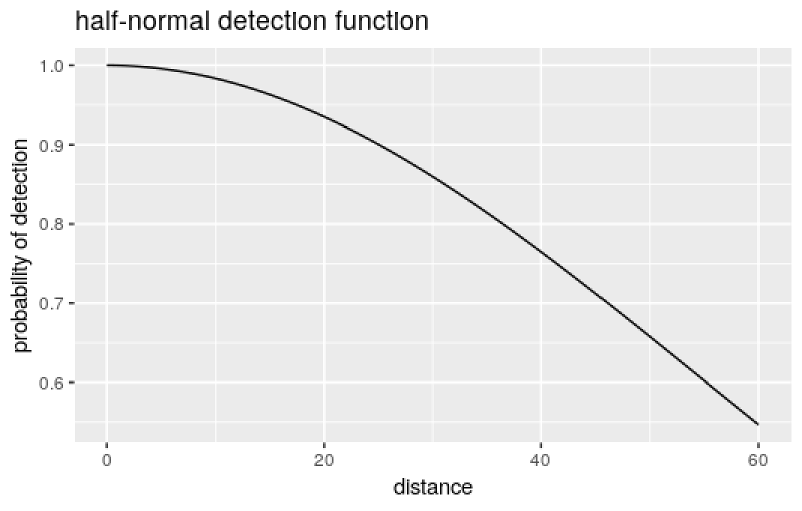
What is distance sampling?



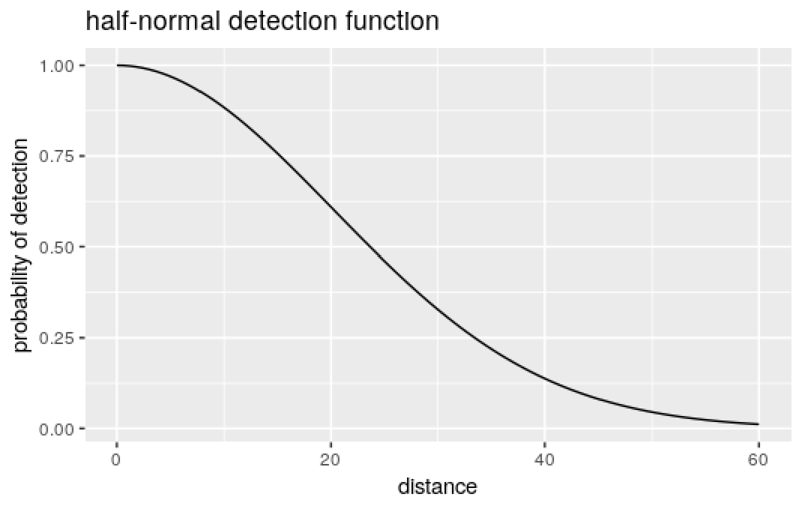
What is distance sampling?



What is distance sampling?



What is distance sampling?



Note the intercept assumed equal to 1

What is distance sampling?

Spatial modelling with distance sampling data

The traditional two-stage approach:

Spatial modelling with distance sampling data

The traditional two-stage approach:

1. Estimate the detection function

Spatial modelling with distance sampling data

The traditional two-stage approach:

1. Estimate the detection function
2. Bin data into counts (discretise space)

Spatial modelling with distance sampling data

The traditional two-stage approach:

1. Estimate the detection function
2. Bin data into counts (discretise space)
3. For each bin, calculate the ‘effective sampled area’

Spatial modelling with distance sampling data

The traditional two-stage approach:

1. Estimate the detection function
2. Bin data into counts (discretise space)
3. For each bin, calculate the ‘effective sampled area’
4. Fit a count GLM/GAM with an ‘effective sampled area’ offset term

Spatial modelling with distance sampling data

The traditional two-stage approach:

1. Estimate the detection function
2. Bin data into counts (discretise space)
3. For each bin, calculate the 'effective sampled area'
4. Fit a count GLM/GAM with an 'effective sampled area' offset term

Pick your favourite GLM/GAM software (`mgcv` is a popular choice)

Spatial modelling with distance sampling data

The traditional two-stage approach:

1. Estimate the detection function
2. Bin data into counts (discretise space)
3. For each bin, calculate the 'effective sampled area'
4. Fit a count GLM/GAM with an 'effective sampled area' offset term

Pick your favourite GLM/GAM software (`mgcv` is a popular choice)

Notable alternative: `unmarked` package (Fiske et al., 2011) does one stage maximum likelihood with a multinomial formulation (Royle et al., 2004) (distances + space must be discrete classes).

R-INLA allows us to fit GLMs/GAMs

R-INLA

R-INLA allows us to fit GLMs/GAMs

Why bother?

`R-INLA` allows us to fit GLMs/GAMs

Why bother?

1. `R-INLA` does fast approximate Bayesian computation using Laplace approximations (avoids MCMC)

`R-INLA`

`R-INLA` allows us to fit GLMs/GAMs

Why bother?

1. `R-INLA` does fast approximate Bayesian computation using Laplace approximations (avoids MCMC)
2. Incorporate prior knowledge of the species (in practice, this is rare to see)

R-INLA

R-INLA allows us to fit GLMs/GAMs

Why bother?

1. R-INLA does fast approximate Bayesian computation using Laplace approximations (avoids MCMC)
2. Incorporate prior knowledge of the species (in practice, this is rare to see)
3. **One-stage model fit** - simultaneously estimate detection and spatial models

R-INLA

R-INLA allows us to fit GLMs/GAMs

Why bother?

1. R-INLA does fast approximate Bayesian computation using Laplace approximations (avoids MCMC)
2. Incorporate prior knowledge of the species (in practice, this is rare to see)
3. **One-stage model fit** - simultaneously estimate detection and spatial models
4. **Sample from joint and marginal posterior distributions** - uncertainty in detectability is accounted for naturally

R-INLA

R-INLA allows us to fit GLMs/GAMs

Why bother?

1. R-INLA does fast approximate Bayesian computation using Laplace approximations (avoids MCMC)
2. Incorporate prior knowledge of the species (in practice, this is rare to see)
3. **One-stage model fit** - simultaneously estimate detection and spatial models
4. **Sample from joint and marginal posterior distributions** - uncertainty in detectability is accounted for naturally
5. Use popular R-INLA features (**point process likelihoods**, **scalable sparse random effects**, spatio-temporal models, barrier model, extreme value distributions, penalised complexity priors)

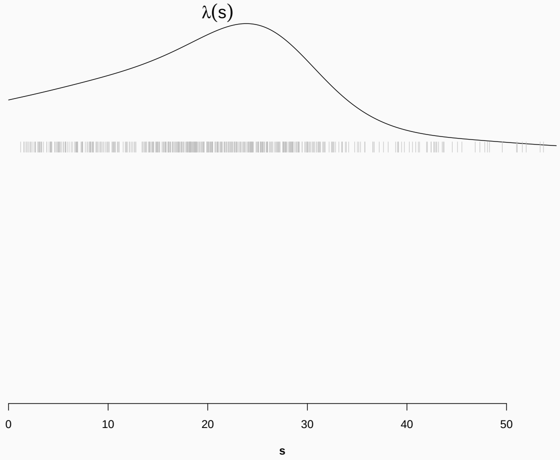
R-INLA

R-INLA allows us to fit GLMs/GAMs

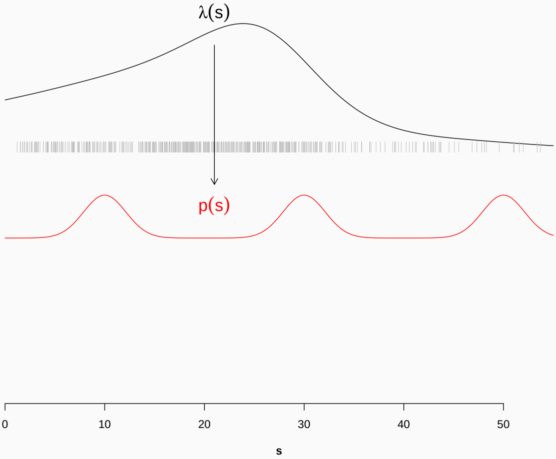
Why bother?

1. R-INLA does fast approximate Bayesian computation using Laplace approximations (avoids MCMC)
2. Incorporate prior knowledge of the species (in practice, this is rare to see)
3. **One-stage model fit** - simultaneously estimate detection and spatial models
4. **Sample from joint and marginal posterior distributions** - uncertainty in detectability is accounted for naturally
5. Use popular R-INLA features (**point process likelihoods**, **scalable sparse random effects**, spatio-temporal models, barrier model, extreme value distributions, penalised complexity priors)
6. Support for multiple likelihoods (e.g. multiple data sources)

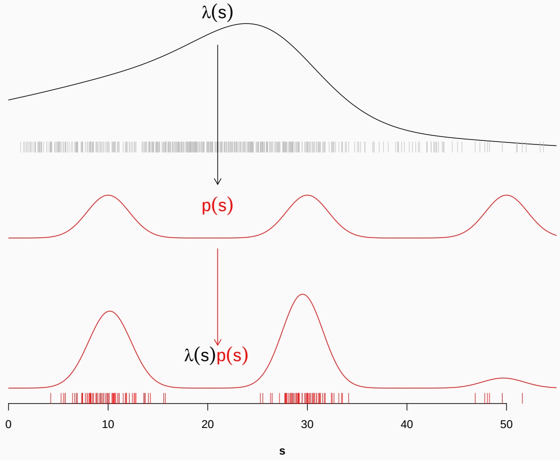
A Point Process Perspective



A Point Process Perspective

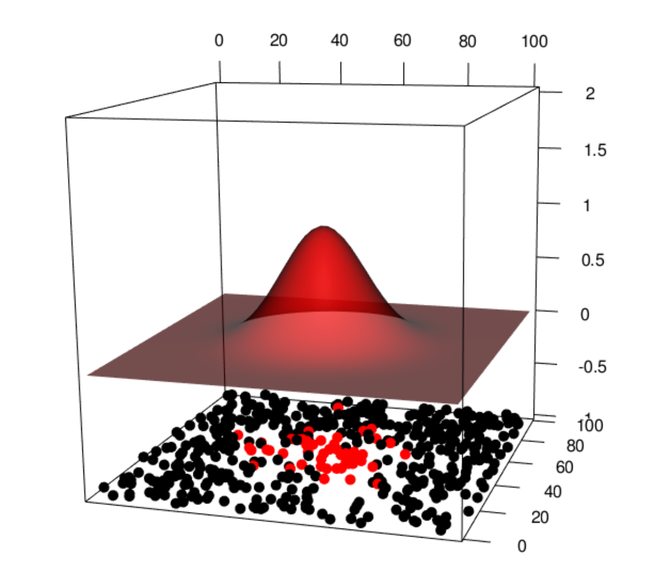


A Point Process Perspective



A Point Process Perspective

A Point Process Perspective



A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

$$N(A) \sim \text{Poisson}(\tilde{\Lambda})$$

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

$$N(A) \sim \text{Poisson}(\tilde{\Lambda})$$
$$\tilde{\Lambda} = \int_A \tilde{\lambda}(s)\mathrm{d}s$$

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

$$N(A) \sim \text{Poisson}(\tilde{\Lambda})$$
$$\tilde{\Lambda} = \int_A \tilde{\lambda}(s)\mathrm{d}s$$
$$\log \tilde{\lambda}(s) = \log \lambda(s) + \log p(s)$$

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

$$N(A) \sim \text{Poisson}(\tilde{\Lambda})$$

$$\tilde{\Lambda} = \int_A \tilde{\lambda}(s)\mathrm{d}s$$

$$\log \tilde{\lambda}(s) = \log \lambda(s) + \log p(s)$$

$\log \tilde{\lambda}(s)$ is the **predictor**

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

$$N(A) \sim \text{Poisson}(\tilde{\Lambda})$$
$$\tilde{\Lambda} = \int_A \tilde{\lambda}(s)\mathrm{d}s$$
$$\log \tilde{\lambda}(s) = \log \lambda(s) + \log p(s)$$

$\log \tilde{\lambda}(s)$ is the **predictor**

$\log \lambda(s)$ is the **spatial process model**

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

$$N(A) \sim \text{Poisson}(\tilde{\Lambda})$$
$$\tilde{\Lambda} = \int_A \tilde{\lambda}(s)\mathrm{d}s$$
$$\log \tilde{\lambda}(s) = \log \lambda(s) + \log p(s)$$

$\log \tilde{\lambda}(s)$ is the **predictor**

$\log \lambda(s)$ is the **spatial process model**

$\log p(s)$ is the **observation process model**

A Point Process Perspective

The intensity for *detected* points is $\tilde{\lambda}(s) = \lambda(s)p(s)$

Let $\tilde{N}(A)$ denote the number of *detected* points within a region A

$$N(A) \sim \text{Poisson}(\tilde{\Lambda})$$
$$\tilde{\Lambda} = \int_A \tilde{\lambda}(s)\mathrm{d}s$$
$$\log \tilde{\lambda}(s) = \log \lambda(s) + \log p(s)$$

$\log \tilde{\lambda}(s)$ is the **predictor**

$\log \lambda(s)$ is the **spatial process model**

$\log p(s)$ is the **observation process model**

This model is implemented in `inlabru` as a `"cp"` (Cox process) likelihood

A Point Process Perspective

We don't know exactly where the points are

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

For a single transect with centre $\boldsymbol{s}_0$

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

For a single transect with centre $\boldsymbol{s}_0$

$$\boldsymbol{s}(r, \theta) = \boldsymbol{s}_0 + r[\cos\theta, \sin\theta]^T$$

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

For a single transect with centre $\boldsymbol{s}_0$

$$\boldsymbol{s}(r, \theta) = \boldsymbol{s}_0 + r[\cos\theta, \sin\theta]^T$$

We only observe the distance r and don't know the angle θ

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

For a single transect with centre $\boldsymbol{s}_0$

$$\boldsymbol{s}(r, \theta) = \boldsymbol{s}_0 + r[\cos\theta, \sin\theta]^T$$

We only observe the distance r and don't know the angle θ

$$\tilde{\lambda}(r) = \int_{c(r)} p(r)\lambda(\boldsymbol{s})\mathrm{d}\boldsymbol{s}$$

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

For a single transect with centre $\boldsymbol{s}_0$

$$\boldsymbol{s}(r, \theta) = \boldsymbol{s}_0 + r[\cos\theta, \sin\theta]^T$$

We only observe the distance r and don't know the angle θ

$$\begin{aligned}\tilde{\lambda}(r) &= \int_{c(r)} p(r)\lambda(\boldsymbol{s})\mathrm{d}\boldsymbol{s} \\ &= \int_0^{2\pi} r p(r)\lambda(\boldsymbol{s}(r, \theta))\mathrm{d}\theta\end{aligned}$$

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

For a single transect with centre $\boldsymbol{s}_0$

$$\boldsymbol{s}(r,\theta) = \boldsymbol{s}_0 + r[\cos\theta, \sin\theta]^T$$

We only observe the distance r and don't know the angle θ

$$\begin{aligned}
\tilde{\lambda}(r) &= \int_{c(r)} p(r)\lambda(\boldsymbol{s})\mathrm{d}\boldsymbol{s} \\
&= \int_0^{2\pi} rp(r)\lambda(\boldsymbol{s}(r,\theta))\mathrm{d}\theta \\
&= 2\pi rp(r)\lambda(\boldsymbol{s}_0)
\end{aligned}$$

A Point Process Perspective

We don't know exactly where the points are

We can derive the appropriate intensity (point transects presented here)

For a single transect with centre $\boldsymbol{s}_0$

$$\boldsymbol{s}(r, \theta) = \boldsymbol{s}_0 + r[\cos\theta, \sin\theta]^T$$

We only observe the distance r and don't know the angle θ

$$\begin{aligned}\tilde{\lambda}(r) &= \int_{c(r)} p(r)\lambda(\boldsymbol{s})\mathrm{d}\boldsymbol{s} \\ &= \int_0^{2\pi} r p(r)\lambda(\boldsymbol{s}(r,\theta))\mathrm{d}\theta \\ &= 2\pi r p(r)\lambda(\boldsymbol{s}_0)\end{aligned}$$

$2\pi r$ accounts for the increasing circumference as distance increases

A Point Process Perspective

$$\log \tilde{\lambda}(s, r) = \log 2\pi r + \log p(r) + \log \lambda(s)$$

A Point Process Perspective

$$\log \tilde{\lambda}(s, r) = \log 2\pi r + \log p(r) + \log \lambda(s)$$

$$p(r) = \exp\left(-r^2/2\sigma^2\right) \quad \text{where } \sigma^2 > 0$$

A Point Process Perspective

$$\log \tilde{\lambda}(s, r) = \log 2\pi r + \log p(r) + \log \lambda(s)$$

$$p(r) = \exp\left(-r^2/2\sigma^2\right) \text{ where } \sigma^2 > 0$$

This is a non-linear thing in our predictor!

A Point Process Perspective

$$\log \tilde{\lambda}(s, r) = \log 2\pi r + \log p(r) + \log \lambda(s)$$

$$p(r) = \exp\left(-r^2/2\sigma^2\right) \text{ where } \sigma^2 > 0$$

This is a non-linear thing in our predictor!

If we knew $p(r)$ exactly then this would act like an offset term

A Point Process Perspective

$$\log \tilde{\lambda}(s, r) = \log 2\pi r + \log p(r) + \log \lambda(s)$$

$$p(r) = \exp\left(-r^2/2\sigma^2\right) \text{ where } \sigma^2 > 0$$

This is a non-linear thing in our predictor!

If we knew $p(r)$ exactly then this would act like an offset term

Guess $p(r)$, fit a model, search for a better guess, iterate until convergence

A Point Process Perspective

$$\log \tilde{\lambda}(s, r) = \log 2\pi r + \log p(r) + \log \lambda(s)$$

$$p(r) = \exp\left(-r^2/2\sigma^2\right) \text{ where } \sigma^2 > 0$$

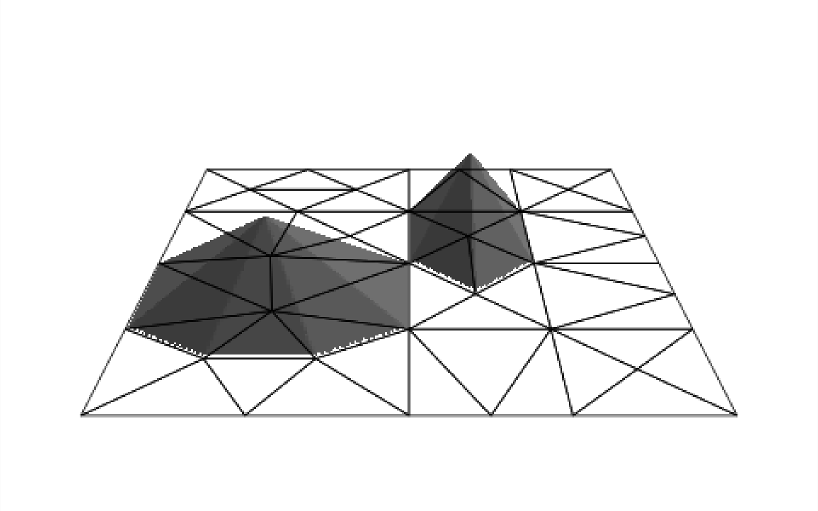
This is a non-linear thing in our predictor!

If we knew $p(r)$ exactly then this would act like an offset term

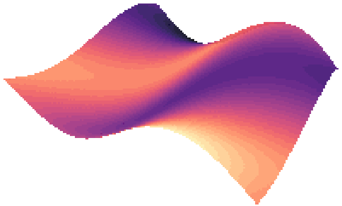
Guess $p(r)$, fit a model, search for a better guess, iterate until convergence

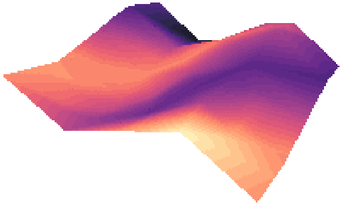
we call this approach *iterated* INLA and it is implemented in `inlabru` (details in forthcoming paper...)

Spatial Model



Spatial Model





Summary

To summarise:

Summary

To summarise:

- Take a point process perspective

Summary

To summarise:

- Take a point process perspective
- Observations depend on a *thinned* intensity

Summary

To summarise:

- Take a point process perspective
- Observations depend on a *thinned* intensity
- Thinning by the detection function

Summary

To summarise:

- Take a point process perspective
- Observations depend on a *thinned* intensity
- Thinning by the detection function
- Spatial random effect to estimate the intensity

Summary

To summarise:

- Take a point process perspective
- Observations depend on a *thinned* intensity
- Thinning by the detection function
- Spatial random effect to estimate the intensity
- log-Gaussian Cox point process likelihood (implemented in `inlabru`)

Summary

To summarise:

- Take a point process perspective
- Observations depend on a *thinned* intensity
- Thinning by the detection function
- Spatial random effect to estimate the intensity
- log-Gaussian Cox point process likelihood (implemented in `inlabru`)
- Iterated INLA for inference (implemented in `inlabru`)

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

What is inlabru?

inlabru is, in part, a **wrapper** for R-INLA

Aims to make fiddly R-INLA things easier to do:

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)

What is inlabru?

inlabru is, in part, a **wrapper** for R-INLA

Aims to make fiddly R-INLA things easier to do:

- Support for spatial modelling (handles sp objects)
- ‘Automatic’ stack-building

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- ‘Automatic’ stack-building
- User-named model components (no more remembering your indexing)

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- 'Automatic' stack-building
- User-named model components (no more remembering your indexing)
- Support for joint likelihood models

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- 'Automatic' stack-building
- User-named model components (no more remembering your indexing)
- Support for joint likelihood models
- `predict(...)` methods and `plot(...)` functions

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- 'Automatic' stack-building
- User-named model components (no more remembering your indexing)
- Support for joint likelihood models
- `predict(...)` methods and `plot(...)` functions

`inlabru` is also an **extension** of `R-INLA`

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- 'Automatic' stack-building
- User-named model components (no more remembering your indexing)
- Support for joint likelihood models
- `predict(...)` methods and `plot(...)` functions

`inlabru` is also an **extension** of `R-INLA`

- define non-linear model components in the predictor

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- 'Automatic' stack-building
- User-named model components (no more remembering your indexing)
- Support for joint likelihood models
- `predict(...)` methods and `plot(...)` functions

`inlabru` is also an **extension** of `R-INLA`

- define non-linear model components in the predictor
- Iterated INLA

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- 'Automatic' stack-building
- User-named model components (no more remembering your indexing)
- Support for joint likelihood models
- `predict(...)` methods and `plot(...)` functions

`inlabru` is also an **extension** of `R-INLA`

- define non-linear model components in the predictor
- Iterated INLA
- Cox process likelihood

What is `inlabru`?

`inlabru` is, in part, a **wrapper** for `R-INLA`

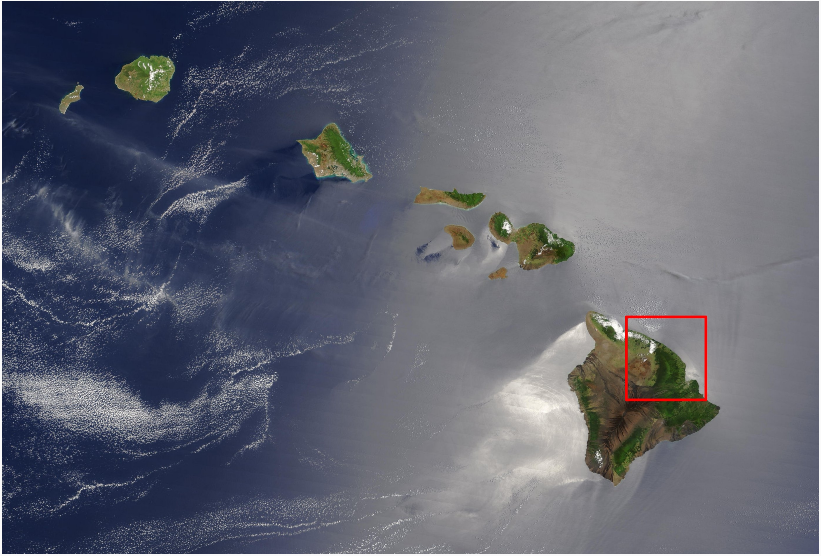
Aims to make fiddly `R-INLA` things easier to do:

- Support for spatial modelling (handles `sp` objects)
- 'Automatic' stack-building
- User-named model components (no more remembering your indexing)
- Support for joint likelihood models
- `predict(...)` methods and `plot(...)` functions

`inlabru` is also an **extension** of `R-INLA`

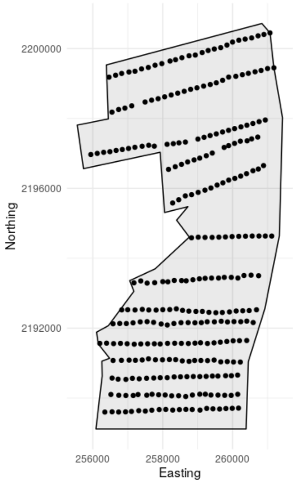
- define non-linear model components in the predictor
- Iterated INLA
- Cox process likelihood
- Support for building numerical integration schemes

Akepa Case Study

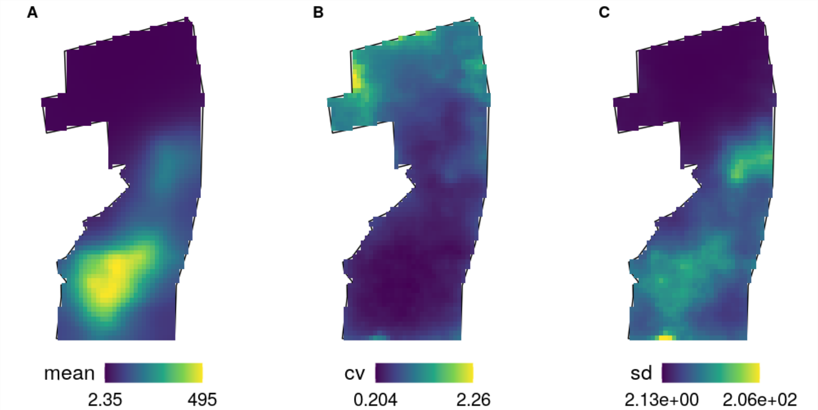


Akepa Case Study

Hawaii Forest Bird Survey transect locations

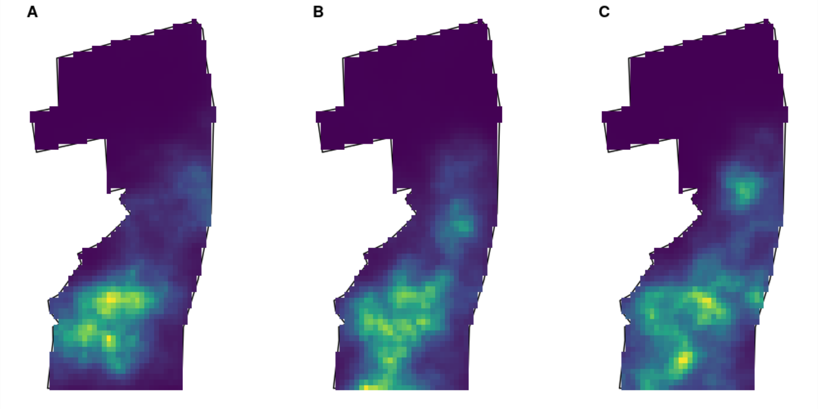


Akepa results



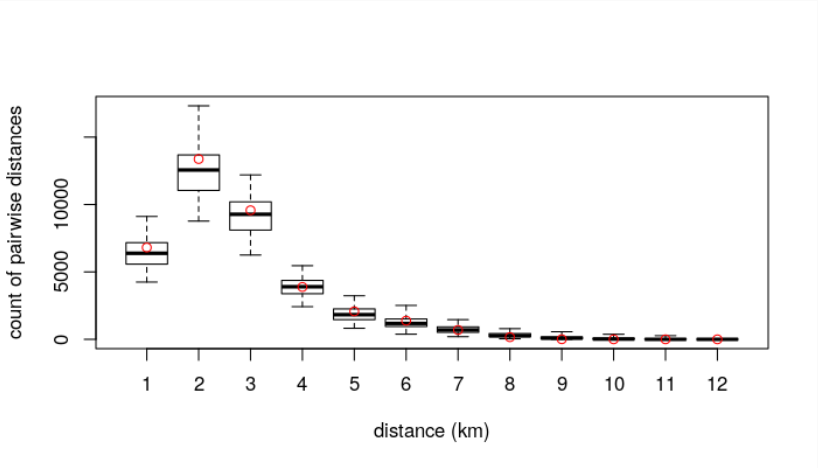
Summary statistics of the posterior intensity field

Akepa results



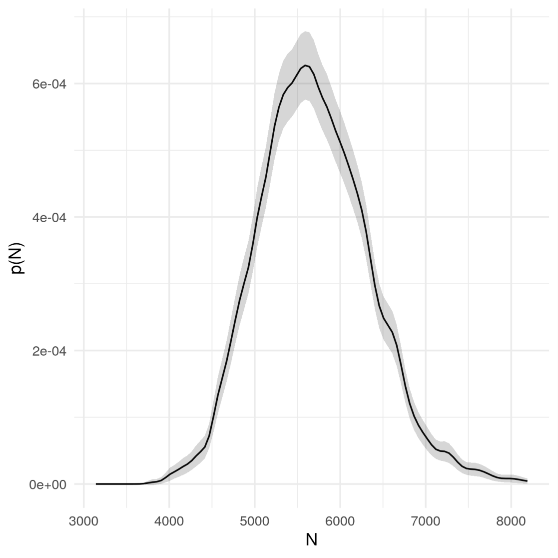
Three realisations of the posterior intensity field

Akepa results



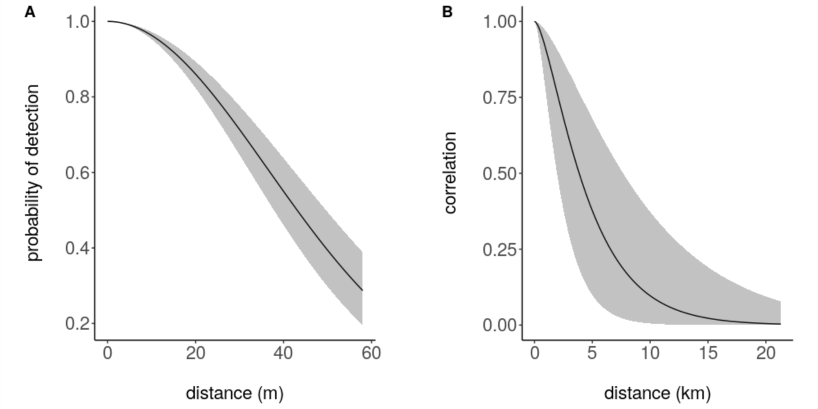
A "distance sampling adjusted" Ripley's K-function

Akepa results



Posterior of realised abundance $\pm$ 2 Monte Carlo standard errors

Akepa results



Posterior detection function and Matérn covariance function

Aims of the talk

1. Demonstrate how to fit distance sampling models in `R-INLA`/`inlabru`.
2. Highlight *general* features of `inlabru` that are applicable in a wide range of contexts (not just distance sampling).

Non-linear model components in spatial ecology

Directly model detectability

Non-linear model components in spatial ecology

Directly model detectability

Many papers in species distribution modelling use a log-linear effect of a covariate “to account for detectability”.

Non-linear model components in spatial ecology

Directly model detectability

Many papers in species distribution modelling use a log-linear effect of a covariate “to account for detectability”.

This term in the model not constrained to be between 0 and 1

Non-linear model components in spatial ecology

Directly model detectability

Many papers in species distribution modelling use a log-linear effect of a covariate "to account for detectability".

This term in the model not constrained to be between 0 and 1

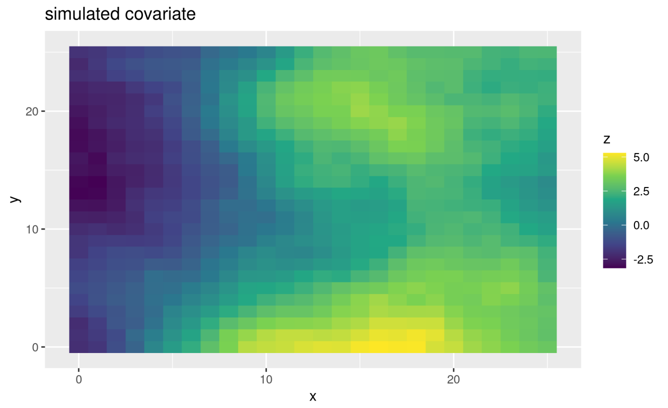
`inlabru` allows us to use `pnorm(...)`, for example.

Non-linear model components in spatial ecology

Functional responses

e.g. a "saturating" response:

$$f(z|\alpha,\gamma) = \alpha(1 - \exp(-\gamma z))$$

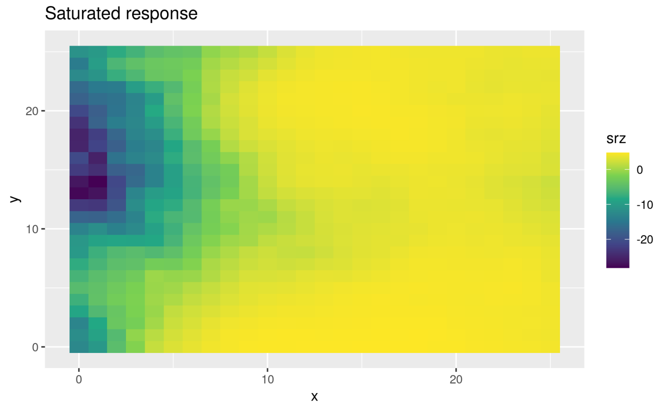


Non-linear model components in spatial ecology

Functional responses

e.g. a "saturating" response:

$$f(z|\alpha,\gamma) = \alpha(1 - \exp(-\gamma z))$$

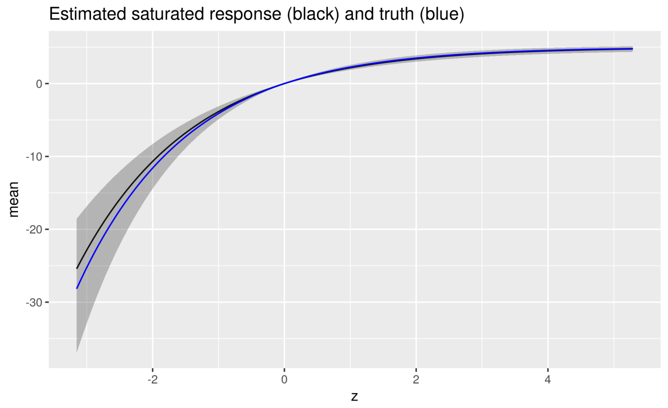


Non-linear model components in spatial ecology

Functional responses

e.g. a "saturating" response:

$$f(z|\alpha,\gamma) = \alpha(1 - \exp(-\gamma z))$$



Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships
- I'm sure many more! (please tell me)

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships
- I'm sure many more! (please tell me)

Replace random effects with parametric dependence structures:

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships
- I'm sure many more! (please tell me)

Replace random effects with parametric dependence structures:

- Instead of, for example, AR1 through time, replace AR1 with some ecological theory?

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships
- I'm sure many more! (please tell me)

Replace random effects with parametric dependence structures:

- Instead of, for example, AR1 through time, replace AR1 with some ecological theory?
- e.g. Ricker growth model

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships
- I'm sure many more! (please tell me)

Replace random effects with parametric dependence structures:

- Instead of, for example, AR1 through time, replace AR1 with some ecological theory?
- e.g. Ricker growth model
- The population ecology literature is vast

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships
- I'm sure many more! (please tell me)

Replace random effects with parametric dependence structures:

- Instead of, for example, AR1 through time, replace AR1 with some ecological theory?
- e.g. Ricker growth model
- The population ecology literature is vast

One more idea in the works:

Non-linear model components in spatial ecology

Build an *interpretable* predictor expression that incorporates our understanding of the ecology.

Other functional responses:

- Trade-offs between resources (prey-choice literature)
- Selectivity curves for fisheries data
- Force positive / negative / quadratic relationships
- I'm sure many more! (please tell me)

Replace random effects with parametric dependence structures:

- Instead of, for example, AR1 through time, replace AR1 with some ecological theory?
- e.g. Ricker growth model
- The population ecology literature is vast

One more idea in the works:

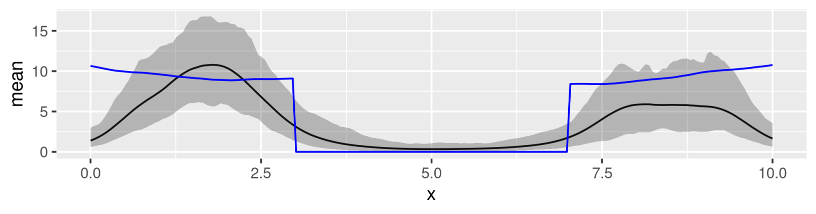
- Level set Cox process - allow a mixture of random fields

A word of warning

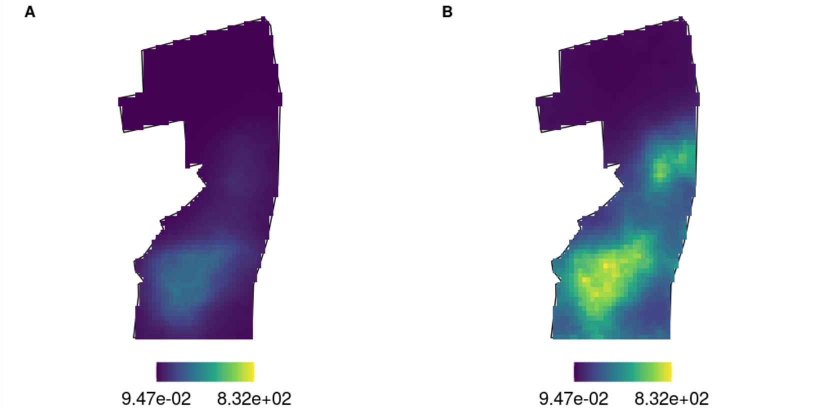
References

- F. E. Bachl, F. Lindgren, D. L. Borchers, and J. B. Illian.
  Inlabru: An R package for Bayesian spatial modelling from ecological survey data.
  Methods in Ecology and Evolution, 2019.
- M. V. Bravington, D. L. Miller, and S. L. Hedley.
  Reliable variance propagation for spatial density surface models.
  arXiv:1807.07996 [stat], July 2018.
- R. J. Camp, D. L. Miller, L. Thomas, S. T. Buckland, and S. J. Kendall.
  Using density surface models to estimate spatio-temporal changes in population densities and trend.
  Ecography, 2020.
- I. Fiske and R. Chandler.
  Unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance.
  Journal of Statistical Software, 43:1–23, 2011.
- D. L. Miller, R. Glennie, and A. E. Seaton.
  Understanding the Stochastic Partial Differential Equation Approach to Smoothing.
  Journal of Agricultural, Biological and Environmental Statistics, Sept. 2019.
- J. A. Royle, D. K. Dawson, and S. Bates.
  Modeling abundance effects in distance sampling.
  Ecology, 85(6):1591–1597, 2004.
- Y. Yuan, F. E. Bachl, F. Lindgren, D. L. Borchers, J. B. Illian, S. T. Buckland, H. Rue, and T. Gerrodette.
  Point process models for spatio-temporal distance sampling data from a large-scale survey of blue whales.
  The Annals of Applied Statistics, 2017.

Extras!

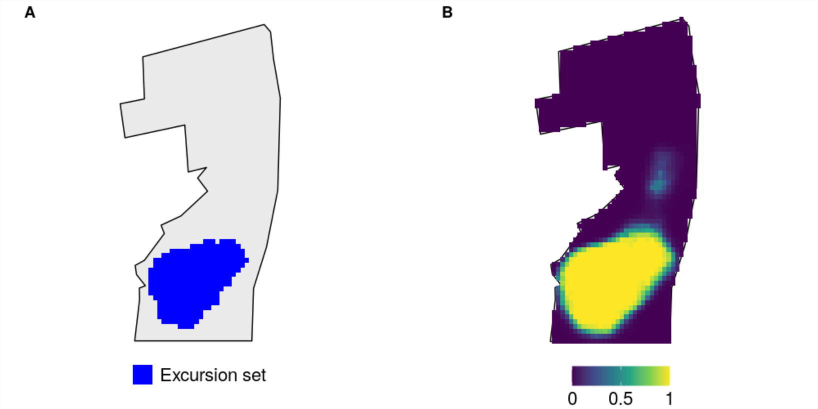


Extras!



0.025 and 0.975 pointwise prediction quantiles for the posterior intensity field

Extras!



Excursion set for 1 bird per hectare and 95% probability level and corresponding excursion function