`GeoModels` Tutorial: simulation, estimation and prediction of spatial count data using Poisson random fields

Diego Morales-Navarrete
Moreno Bevilacqua

August 5, 2023

Introduction

In this tutorial we show how to analyze spatial count data using Poisson random fields as proposed in Morales-Navarrete et al. (2021) i.e. random fields with Poisson marginal distribution using the *R* package `GeoModels` (Bevilacqua et al. (2018)).

We first load the *R* libraries needed in this tutorial and set the name of the model in the `GeoModels` package.

```
rm(list=ls());
require(GeoModels);
require(fields);
model="Poisson";  # model name in the  GeoModels package
```

Poisson random fields

The definition of a Poisson random field starts by considering a 'parent' Gaussian random field $G = \{G(\boldsymbol{s}), \boldsymbol{s} \in A\}$, where $\boldsymbol{s}$ represents a location in the domain A. In this tutorial we consider the spatial case *i.e.* $A \subseteq \mathbb{R}^2$. However, the package `GeoModels` allows to work also with spatio-temporal data or data defined on a sphere of arbitrary radius. The Gaussian random field G is assumed weakly stationary with zero mean, unit variance and correlation function $\rho(\boldsymbol{h}) = \mathrm{cor}(G(\boldsymbol{s}+\boldsymbol{h}), G(\boldsymbol{s}))$.

Let G_1, G_2 be two independent copies of G and let us define the random field $W = \{W(\mathbf{s}), \mathbf{s} \in A\}$ as:

$$W(\mathbf{s}) := \frac{1}{2\lambda(\mathbf{s})} \sum_{k=1}^{2} G_k^2(\mathbf{s}). \tag{1}$$

where $\lambda(\mathbf{s}) > 0$ is a non-random function. It turns out that W is a stationary random field with marginal exponential distribution with parameter $\lambda(\mathbf{s})$ that is $W(\mathbf{s}) \sim Exp(\lambda(\mathbf{s}))$ with $\mathbb{E}(W(\mathbf{s})) = 1/\lambda(\mathbf{s})$.

By considering an infinite sequence of independent copies $W_1, W_2 \ldots$, of W a Poisson random field, $N_t := \{N_t(\mathbf{s}), \mathbf{s} \in A\}$ for $t > 0$ can be defined as:

$$N_t(\mathbf{s}) := \begin{cases} 0 & if \quad 0 \leq t < S_1(\mathbf{s}) \\ \max\limits_{n \geq 1}\{S_n(\mathbf{s}) \leq t\} & if \quad S_1(\mathbf{s}) \leq t \end{cases}, \tag{2}$$

where $S_n(\mathbf{s}) = \sum_{i=1}^{n} W_i(\mathbf{s})$ is the n-fold convolution of W. $N_t(\mathbf{s})$ represents the random total number of events that have occurred up to time t at location site $\mathbf{s}$ that is for each $\mathbf{s}$ it is a Poisson counting process. In addition, given two spatial locations $\mathbf{s}_1$, $\mathbf{s}_2$, the associated Poisson counting processes $N_t(\mathbf{s}_1)$ and $N_t(\mathbf{s}_2)$ are correlated.

The previous model can be viewed as a spatial generalization of the Poisson counting process (Cox, 1970; Mainardi et al., 2007), where we consider as ‘inter-arrival times” independent copies of positive random fields instead of an $i.i.d.$ sequence of positive random variables. By construction for each $\mathbf{s}$, $N_t(\boldsymbol{s}) \sim Pois(t\lambda(\mathbf{s}))$ is the marginal distribution. For the purpose of this tutorial and without loss of generality we assume $t = 1$ and we set $N := N_t$. Then $\mathbb{E}(N(\mathbf{s})) = Var(N(\mathbf{s})) = \lambda(\mathbf{s})$ and the correlation function of the non-stationary Poisson random field is given by (Morales-Navarrete et al., 2021):

$$\rho_N(\mathbf{s}_i, \mathbf{s}_j) = \frac{\rho^2(\mathbf{h})(1-\rho^2(\mathbf{h}))}{\sqrt{\lambda(\mathbf{s}_i)\lambda(\mathbf{s}_j)}} \sum_{r=0}^{\infty} \gamma^* \left(r+1, \frac{\lambda(\mathbf{s}_i)}{1-\rho^2(\mathbf{h})}\right) \gamma^* \left(r+1, \frac{\lambda(\mathbf{s}_j)}{1-\rho^2(\mathbf{h})}\right), \tag{3}$$

with $\mathbf{h} = \mathbf{s}_i - \mathbf{s}_j$ and $\gamma^*(\cdot, \cdot)$ the regularized lower incomplete gamma function.

Simulation of Poisson random fields

We first set the spatial coordinates (Figure 1):

```
set.seed(1989);
N=500;
coords=cbind(runif(N),runif(N));
plot(coords ,pch=20,xlab="",ylab="");
```

Stationary Poisson random fields

Let us assume that $\lambda(\boldsymbol{s}) = \lambda$, i.e., we are assuming a constant mean for the Poisson random field. In this case the correlation (3) simplifies to

$$\rho_N(\mathbf{h}, \lambda) = \rho^2(\mathbf{h}) \left[1 - \exp\left(-z(\mathbf{h}, \lambda)\right) \left(I_0\left(z(\mathbf{h})\right) + I_1\left(z(\mathbf{h}, \lambda)\right)\right)\right], \tag{4}$$

where $z(\mathbf{h}, \lambda) = 2\lambda(1-\rho^2(\mathbf{h}))^{-1}$ and $I_a(\cdot)$ is the Bessel function of the first kind of order a.

To obtain a simulation from a stationary Poisson random field we need to specify the mean and a parametric correlation model $\rho(\boldsymbol{h})$ for the underlying Gaussian random field.

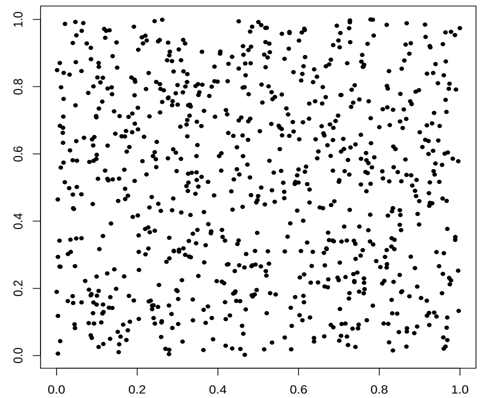

Figure 1: Spatial location sites used in the tutorial.

For the correlation function $\rho(\boldsymbol{h})$ of the "parent" Gaussian random field G we assume an isotropic Matérn model (Matérn, 1986):

$$\rho_{\alpha,\gamma}(\boldsymbol{h}) = \frac{2^{1-\gamma}}{\Gamma(\gamma)} \left(\frac{\|\boldsymbol{h}\|}{\alpha}\right)^{\gamma} \mathcal{K}_{\gamma}\left(\frac{\|\boldsymbol{h}\|}{\alpha}\right), \qquad \|\boldsymbol{h}\| \geq 0. \tag{5}$$

where $\mathcal{K}_\gamma$ is a modified Bessel function of the second kind of order γ, $\gamma > 0$ is the smoothness parameter and $\alpha > 0$ the spatial scale parameter. Then, we set the parameter associated to this correlation model:

```
corrmodel = "Matern";         ## correlation model
scale = 0.25/3;                ## scale parameter
smooth=0.5;                   ## smooth parameter
nugget=0;                     ## nugget parameter
```

Finally we set the mean parameter, i.e., the β parameter in $\lambda = e^{\beta}$:

```
mean = 1.5;# mean parameter
```

Simulation is performed using Cholesky decomposition for the two Gaussian random fields involved. We are now ready to simulate a realization of the Poisson random field N using the function *GeoSim*:

```
param=list(nugget=nugget,mean=mean, scale=scale,
                smooth=smooth);
data_s = GeoSim(coordx=coords,corrmodel=corrmodel,
                 param=param,model=model)$data
```

Note that empirical mean and variance and theoretical mean are very close as expected:

```
> mean(data_s);var(data_s)
[1] 4.516
[1] 4.402549
> exp(mean)
[1] 4.481689
```

The following figure shows the distribution of the data

```
plot(table(data_s),ylab = "Frequency")
```

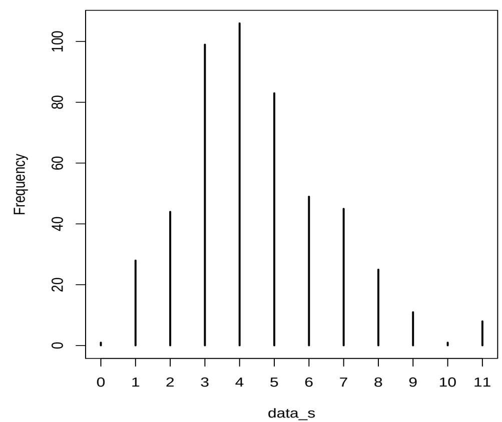

Figure 2: Distribution of the simulated Poisson random field.

Non Stationary Poisson random fields

A non stationary Poisson random field can be specified by assuming that $\lambda(\boldsymbol{s}) = \exp\{X(\boldsymbol{s})^T\boldsymbol{\beta}\}$ where $X(\boldsymbol{s})$ is a k-dimensional vector of covariates and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^T$ is a k-dimensional vector of (unknown) parameters. In this tutorial we assume $k = 2$.

Thus, in order to obtain a realization from a non stationary Poisson random field we need to specify the regression parameters and a parametric correlation model $\rho(\boldsymbol{h})$ for the underlying Gaussian random field G.

For the correlation function $\rho(\boldsymbol{h})$ we assume a special case of the isotropic Generalized Wendland class (Bevilacqua et al., 2019):

$$\rho_{\alpha,\delta}(\boldsymbol{h}) := \begin{cases} (1 - ||\boldsymbol{h}||/\alpha)^{\delta} & ||\boldsymbol{h}|| < \alpha \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

Then, we set the parameter associated to this correlation model:

```
corrmodel = "Wend0";          ## correlation model
scale = 0.2;                 ## scale parameter
power2=4 ;                    ## power parameter
nugget=0;                      ## nugget parameter
```

Finally we set the regression parameters and the regression matrix:

```
mean = 1.5 # regression parameter beta_0
mean1=-0.25 # regression parameter beta_1
a0=rep(1,N);a1=runif(N)
X=cbind(a0,a1); ## regression matrix
```

To simulate a realization of the Poisson random field N we use the function *GeoSim*:

```
param=list(nugget=nugget,mean=mean,mean1=mean1, scale=scale,
                power2=power2);
data_ns <- GeoSim(coordx=coords ,corrmodel=corrmodel , param=param ,
                        X=X,model=model)$data;
```

Estimation of Poisson random fields

Estimation of regression and correlation parameters of the Poisson random field N can be performed using pairwise likelihood estimation. Let $\Pr(N(\mathbf{s}_i) = n_i, N(\mathbf{s}_j) = n_j)$ the density of the bivariate random vector $(N(\boldsymbol{s}_i), N(\boldsymbol{s}_j))^T$ given in Morales-Navarrete et al. (2021).

Given a partial realization $(n(\boldsymbol{s}_1), \dots, n(\boldsymbol{s}_l))^T$ of the Poisson random process N defined in equation (2). Then, the pairwise likelihood function is defined as:

$$pl(\boldsymbol{\theta}) = \sum_{i=1}^{l} \sum_{j \neq i}^{l} log(\Pr(N(\mathbf{s}_i) = n_i, N(\mathbf{s}_j) = n_j))c_{ij}, \tag{7}$$

In this case $\boldsymbol{\theta} = (\boldsymbol{\beta}, \alpha)^T$ and an efficient way to specify the (non symmetric) weights from computational and efficient viewpoint is based on neighborhoods:

$$c_{ij}(k) = \begin{cases} 1 & \mathbf{s}_i \in N_k(\mathbf{s}_j) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $N_k(\mathbf{s}_l)$ is the set of the neighbors of order $k = 1, 2, \ldots$ of the point $\mathbf{s}_l$.

The pairwise likelihood estimator $\hat{\boldsymbol{\theta}}_{pl}$ is obtained maximizing (7) with respect to $\boldsymbol{\theta}$. In the `GeoModels` package, we can choose the fixed parameters and the parameters that can be estimated. Pairwise likelihood estimation can be performed using the function *GeoFit*. In this example, we perform optimization of (7) using the function `nlminb` that allows box-constrained optimization using PORT routines. However other type of optimization algorithms available in R can be used (`BFGS` or `Nelder-Mead` for instance).

Stationary Case

```
optimizer="nlminb";

fixed1<-list(nugget=0,smooth=0.5);
start1<-list(mean=1.5,scale=0.25/3);
lower<-list(mean=-5,scale=0);
upper<-list(mean=5,scale=2);

neighb=3;
corrmodel = "Matern";
fit1 <- GeoFit(data=data_s,coordx=coords,corrmodel=corrmodel,
optimizer=optimizer,lower=lower,upper=upper,
neighb=neighb,start=start1,fixed=fixed1, model = model);
```

Note that the option *neighb=3* set the neighbors of the weight function (8) i.e. $k = 3$. The object `fit1` include informations about the pairwise likelihood estimation:

```
##################################################################
Maximum  Composite-Likelihood Fitting of Poisson Random Fields
Setting: Marginal Composite-Likelihood
Model: Poisson
Type of the likelihood objects: Pairwise
Covariance model: Matern
Optimizer: nlminb
Number of spatial coordinates: 500
Number of dependent temporal realisations: 1
Type of the random field: univariate
Number of estimated parameters: 2
```

```
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -6189.99
Estimated parameters:
   mean    scale
1.51398  0.07921
####################################################################
```

An alternative, less efficient and computationally easier estimator can be obtained by assuming a misspecified Gaussian model in the pairwise likelihood estimation method. Specifically, if in the estimation step we assume a Gaussian random field with the same mean, variance and correlation function of the Poisson random field (see Morales-Navarrete et al. (2021)), then a weighted misspecified Gaussian pairwise likelihood estimation can be performed changing the name of the model in the function `GeoFit`:

```
fit2 <- GeoFit(data=data_s,coordx=coords,corrmodel=corrmodel,
 optimizer=optimizer, lower=lower,upper=upper,neighb=neighb,
start=start1,fixed=fixed1, model = "Gaussian_misp_Poisson")
```

The two estimates are quite similar in this case but in general the misspecified Gaussian assumption leads to a loss of efficiency that increase when degreasing the expectation of the Poisson random field (see Morales-Navarrete et al. (2021) for a comparison between the two estimators).

```
> unlist(fit1$param)
      mean       scale
1.51398109 0.07921053
> unlist(fit2$param)
      mean       scale
1.51140569 0.06933797
```

Finally, empirical and estimated semi-variogramas can be graphically compared, using the function `GeoCovariogram`, as follow:

```
# Empirical estimation of the variogram:
vario <- GeoVariogram(data=data,coordx=coords,maxdist=0.4)
# comparing empirical and estimated semi-variograms
GeoCovariogram(fit1,show.vario=TRUE, vario=vario,pch=20)
```

Non Stationary Case

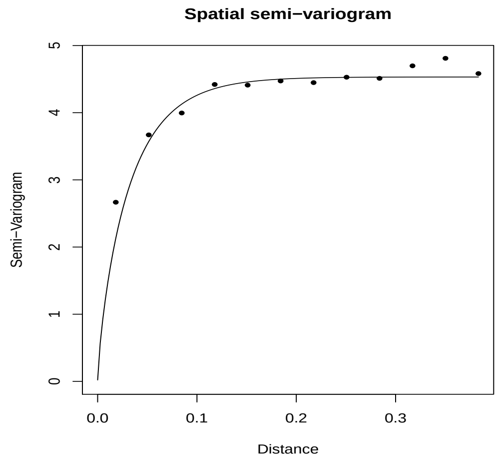

Figure 3: Empirical and estimated semi-variogramas for Poisson data.

```
optimizer="nlminb";
corrmodel = "Wend0";
fixed2<-list(nugget=0,power2=4);
start2<-list(mean=1.5,mean1=-0.25,scale=0.2);
lower<-list(mean=-5,mean1=-5,scale=0);
upper<-list(mean=5,mean1=5,scale=2);
fit1_ns <- GeoFit(data=data_ns,coordx=coords,corrmodel=corrmodel,
optimizer=optimizer,
lower=lower,upper=upper,X=X,
neighb=neighb,start=start2,fixed=fixed2, model = model);
```

The object `fit1_ns` include informations about the pairwise likelihood estimation:

```
fit1_ns
##################################################################
Maximum  Composite-Likelihood Fitting of Poisson Random Fields
Setting: Marginal Composite-Likelihood
Model: Poisson
Type of the likelihood objects: Pairwise
Covariance model: Wend0
Optimizer: nlminb
Number of spatial coordinates: 500
Number of dependent temporal realisations: 1
Type of the random field: univariate
```

```
Number of estimated parameters: 3
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -6433.92
Estimated parameters:
   mean     mean1     scale
 1.5531   -0.2445    0.1476
##################################################################
```

The weighted misspecified Gaussian pairwise likelihood estimation can be performed changing the name of the model in the function `GeoFit`:

```
fit2_ns <- GeoFit(data=data_ns,coordx=coords,corrmodel=corrmodel,
optimizer=optimizer,
lower=lower,upper=upper,X=X,
neighb=neighb,start=start2,fixed=fixed2, model = "Gaussian_misp_Poisson")
```

The two estimates are quite similar in this case but in general the misspecified Gaussian assumption leads to a loss of efficiency.

```
unlist(fit1_ns$param)
 mean         mean1        scale
 1.5530973  -0.2444645   0.1475903
unlist(fit2_ns$param)
      mean         mean1         scale
 1.5603494  -0.2369659   0.1422740
```

Prediction of Poisson random fields

For a given spatial location $\boldsymbol{s}_0$ with associated covariates $X(\boldsymbol{s}_0)$, the optimal linear prediction of a Poisson random field is given by:

$$\widehat{N(\mathbf{s}_0)} = \lambda(\mathbf{s}_0) + \boldsymbol{c}^T\Sigma^{-1}(\boldsymbol{N} - \boldsymbol{\lambda}) \tag{9}$$

where $\boldsymbol{\lambda} = (\lambda(\mathbf{s}_1), \dots, \lambda(\mathbf{s}_l))^T$, $\boldsymbol{c} = [\sqrt{\lambda(\mathbf{s}_0)\lambda(\mathbf{s}_i)}\rho_N(\mathbf{s}_0, \mathbf{s}_i)]_{i=1}^l$ and $\Sigma = \sqrt{\boldsymbol{\lambda}\boldsymbol{\lambda}^T} \odot [\rho_N(\mathbf{s}_i, \mathbf{s}_j)]_{i,j=1}^l$ where $\odot$ is the matrix Schur product. The associated mean squared error is given by:

$$MSE(\widehat{N(\mathbf{s}_0)}) = \lambda(\mathbf{s}_0) - \boldsymbol{c}^T\Sigma^{-1}\boldsymbol{c}. \tag{10}$$

The predictor can be viewed as an optimal Gaussian predictor assuming (3) as correlation function. If the parameters are unknown, both (9) and (10) can be computed replacing the parameters with the pairwise likelihood estimates. Kriging and associated MSE can be obtained using the GeoKrig function.

Stationary Case

We first need to specify the spatial locations to predict and, in this example, we consider a spatial regular grid:

```
xx=seq(0,1,0.015)
loc_to_pred=as.matrix(expand.grid(xx,xx))
```

Then the optimal linear prediction (9), using the estimated parameters, can be performed using the GeoKrig function (computation can be time consuming):

```
param_est=as.list(c(fit1$param,fixed1))
corrmodel = "Matern";
pr=GeoKrig(data=data_s, coordx=coords,loc=loc_to_pred,
     corrmodel=corrmodel,model=model,mse=TRUE,param= param_est)
```

Finally, a kriging map with associate mean square error (Figure 4) can be obtained with the following code:

```
par(mfrow=c(1,3));
colour = rainbow(100)
#### map of  data
quilt.plot(coords[,1], coords[,2], data_s,col=colour,main="Data")
#### map prediction
map=matrix(pr$pred,ncol=length(xx))
image.plot(xx,xx,map,col=colour,xlab="",ylab="",main="Kriging")
#map mean squared error
map_mse=matrix(pr$mse,ncol=length(xx))
image.plot(xx,xx,map_mse,col=colour,xlab="",ylab="",main="MSE")
```

Non Stationary Case

We need to specify the spatial locations to predict and the covariates in those locations, in this example, we consider a spatial regular grid:

```
set.seed(609)
NN=nrow(loc_to_pred)
```

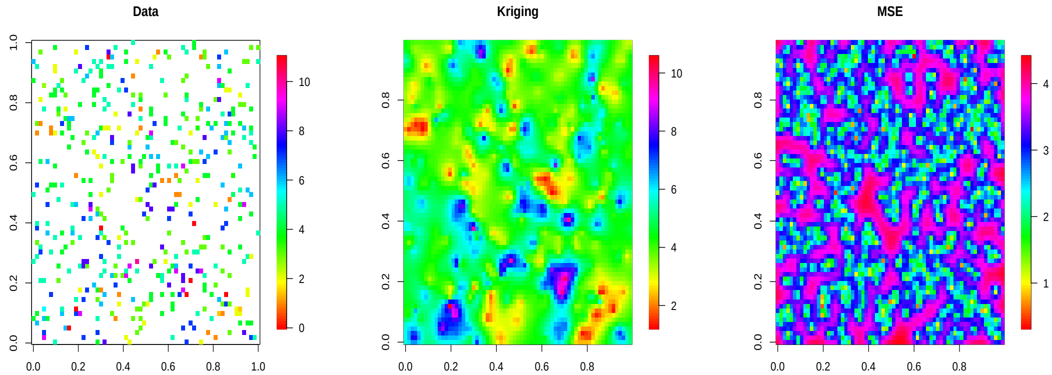

Figure 4: From left to right: observed spatial data, associated kriging map and mean square error map.

```
a0=rep(1,NN);a1=runif(NN)
Xloc=cbind(a0,a1); ## r
corrmodel = "Wend0";
param_est=as.list(c(fit1_ns$param,fixed2))
pr=GeoKrig(data=data_ns, coordx=coords,loc=loc_to_pred,X=X,Xloc=Xloc,
     corrmodel=corrmodel,model=model,mse=TRUE,param= param_est)
```

Then the optimal linear prediction (9), using the estimated parameters, can be performed using the `GeoKrig` function (computation can be time consuming):

```
param_est=as.list(c(fit1_ns$param,fixed2))
pr=GeoKrig(data=data_ns, coordx=coords,loc=loc_to_pred,X=X,Xloc=Xloc,
     corrmodel=corrmodel,model=model,mse=TRUE,param= param_est)
```

Finally, a kriging map with associate mean square error (Figure 5) can be obtained with the following code:

```
par(mfrow=c(1,3));
colour = rainbow(100);
#### map of  data
quilt.plot(coords[,1], coords[,2], data_ns,col=colour,main="Data");
# linear kriging
map=matrix(pr$pred,ncol=length(xx));
image.plot(xx,xx,map,col=colour,xlab="",ylab="",main="Kriging");
#associated mean squared error
```

```
map_mse=matrix(pr$mse,ncol=length(xx));
image.plot(xx,xx,map_mse,col=colour,xlab="",ylab="",main="MSE")
```

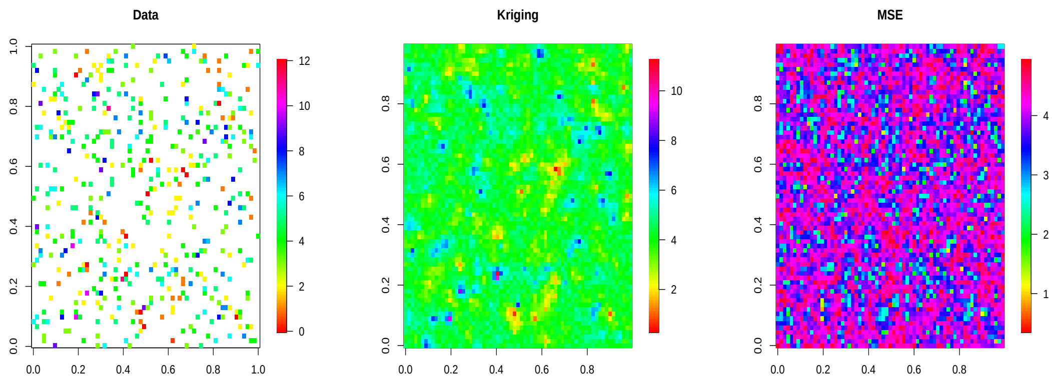

Figure 5: From left to right: observed spatial data, associated kriging map and mean square error map.

References

Bevilacqua, M., T. Faouzi, R. Furrer, and E. Porcu (2019). Estimation and prediction using generalized wendland functions under fixed domain asymptotics. *The Annals of Statistics 47*, 828–856.

Bevilacqua, M., V. Morales-Oñate, and C. Caamaño-Carillo (2018). *GeoModels: A Package for Geostatistical Gaussian and non Gaussian Data Analysis*. R package version 1.0.3-4.

Cox, D. (1970). *Renewal Theory*. London: Methuen & Co.

Mainardi, F., R. Gorenflo, and A. Vivoli (2007). Beyond the poisson renewal process: A tutorial survey. *Journal of Computational and Applied Mathematics 205*(2), 725 – 735. Special issue on evolutionary problems.

Matérn, B. (1986). *Spatial Variation: Stochastic Models and their Applications to Some Problems in Forest Surveys and Other Sampling Investigations* (2nd ed.). Heidelberg: Springer.

Morales-Navarrete, D., M. Bevilacqua, C. Caamaño-Carillo, and L. Castro (2021). Modelling point referenced spatial count data: A poisson process approach. *ArXiv e-prints*.