`GeoModels` Tutorial: simulation, estimation and prediction of global spatial data on the planet Earth using Gaussian random fields.

Moreno Bevilacqua

Introduction

This tutorial illustrates how to analyze spherical data (*i.e.* spatial data on the surface of a sphere) with the `R` package `GeoModels` (Bevilacqua and Morales-Oñate (2018)). Albeit the main focus of the tutorial is on Gaussian random fields (GRFs) defined over a sphere of arbitrary radius, any of the non GRFs implemented in the package `GeoModels` can be coupled with spherical data.

We denote with $\mathbb{S}^2$ the sphere of $\mathbb{R}^3$ of arbitrary radius $R > 0$, defined as

$$\mathbb{S}^2 = \{\boldsymbol{x} \in \mathbb{R}^3 : ||\boldsymbol{x}|| = R\}.$$

This tutorial shows how to simulate, estimate and predict GRFs on a sphere of arbitrary radius using a specific covariance model proposed in Alegría et al. (2020). Other kinds of covariance models can be used for spherical data in the `GeoModels` package but we think that the class proposed in Alegría et al. (2020) is quite flexible for the reasons explained below. We start by loading the libraries needed for the analysis and set the radius of the sphere:

```
rm(list=ls())
require(devtools)
install_github("vmoprojs/GeoModels")
library(GeoModels)
library(mapproj)
library(globe)
library(fields)
library(sphereplot)
set.seed(1891)
```

Gaussian random fields defined on the sphere

We consider a zero mean and unit variance GRF, $Y = \{Y(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{S}^2\}$, defined on a sphere of radius $R > 0$. A point on $\mathbb{S}^2$ can be parametrized trough spherical coordinates using radians as (R, lon^*, lat^*), where $lon^* \in [-\pi, \pi]$ is the longitude and $lat^* \in [-\pi/2, \pi/2]$ is the latitude. Alternatively, one can use decimal degrees (R, lon, lat) where $lon \in [-180, 180]$ and $lat \in [-90, 90]$ (the `GeoModels` package uses the second parameterization).

The great circle (GC) distance corresponds to the shortest path joining two points on the spherical surface and it is the natural distance to be used when analyzing data on a sphere. Alternatively, one can compute the chordal (CH) distance, that is the segment below the arc joining any pair of points located over the spherical shell. Let us give some details on the computation of GC and CH distances. Given two location sites in longitude and latitude (expressed in decimal degrees), that is, $\boldsymbol{x}_i = (\text{lon}_i, \text{lat}_i)$ and $\boldsymbol{x}_j = (\text{lon}_j, \text{lat}_j)$ with $-180 \leq \text{lon}_x \leq 180$, $-90 \leq \text{lat}_x \leq 90$, $x = i, j$ and given the radius R of the Earth, the GC distance, d_{GC}, is given by:

$$d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j) = R\theta_{ij},$$

where

$$\theta_{ij} = [\arccos\{\sin a_i \sin a_j + \cos a_i \cos a_j \cos(b_i - b_j)\}]$$

is the GC distance on the unit sphere. Here, $a_i = (\text{lat}_i)\pi/180$, $a_j = (\text{lat}_j)\pi/180$, $b_i = (\text{lon}_i)\pi/180$, $b_j = (\text{lon}_j)\pi/180$.

The CH distance is instead uniquely defined through the relation

$$d_{CH}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 2Rsin(\theta_{ij}/2). \tag{1}$$

CH obviously underestimates the GC distance, and the approximation error increases with the size of the considered portion of the planet.

We assume $\text{cor}(Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j))) = \rho(d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j))$ where $\rho : [0, R\pi] \to \mathbb{R}$ is continuous with $\rho(0) = 1$. Under this setting, the correlation function is called *geodesically isotropic* (Gneiting, 2013; Porcu et al., 2016). We then consider the location and scale transformation

$$Z(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sigma Y(\boldsymbol{x})$$

where $\mu(\boldsymbol{x}) \in \mathbb{R}$ is a spherical varying mean and $\sigma > 0$. The package `GeoModels` allows a mean regression specification, that is $\mu(\boldsymbol{x}) = M(\boldsymbol{x})^T\boldsymbol{\beta}$ where $M(\boldsymbol{x})$ is a vector of spherical covariates. Nevertheless, for the sake of simplicity this tutorial assumes a constant mean: $\mu(\boldsymbol{x}) = \mu$.

Then $\mathbb{E}(Z(\boldsymbol{x})) = \mu$ and $\text{cov}(Z(\boldsymbol{x}_i), Z(\boldsymbol{x}_j)) = \sigma^2\rho(d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j))$ where σ^2 is the variance parameter and in order to obtain a realization from a spherical RF we need to specify a mean, a variance parameter and a valid (*i.e.*positive definite) geodesically isotropic parametric correlation function $\rho_\gamma(\cdot)$.

We first set the spherical coordinates in decimal degrees on the unit sphere using the function `pointsphere` of the package `sphereplot` (Robotham, 2013):

```
NN=700 ## number of location sites on the sphere
coords=pointsphere(NN,c(-180,180),c(-90,90),c(1,1))[,1:2]
```

Since a sphere of radius $R = 1$ is a (rescaled) satisfactory approximation of planet Earth we can visualized (Figure 1) the locations on the planet Earth using some functions of the package `globe` (Baddeley et al., 2017) with the following code:

```
globeearth(eye=place("newyorkcity"))
globepoints(loc=coords,pch=20,cex=0.4)
globeearth(eye=place("everest"))
globepoints(loc=coords,pch=20,cex=0.4)
```

Figure 1: Location sites on the planet Earth used in this tutorial

An issue when considering correlation models defined on the sphere is that classical models defined on the euclidean spaces are not necessarily suitable on the sphere (Gneiting, 2013). For instance, the popular Matérn model equipped with the great circle distance defined as:

$$\mathcal{M}_{\nu,\alpha}(d_{GC}) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{d_{GC}}{\alpha}\right)^{\nu} \mathcal{K}_{\nu}\left(\frac{d_{GC}}{\alpha}\right),$$

is not valid on the sphere when $\nu > 0.5$. Here, $\mathcal{K}_{\nu}$ is a modified Bessel function of the second kind of order ν.

A interesting covariance model defined on the unit sphere having the same features of the Matérn model is the the so called $\mathcal{F}$ family of correlation functions proposed in Alegría et al. (2020), defined as:

$$\mathcal{F}_{\tau,\alpha,\nu}(d_{GC}) = \frac{B(\alpha, \nu+\tau)}{B(\alpha,\nu)} {}_2F_1(\tau, \alpha, \alpha+\nu+\tau; \cos(d_{GC})), \tag{2}$$

where τ, α and ν are strictly positive parameters and where ${}_2F_1$ is the Gauss Hypergeometric function

$${}_2F_1(a,b,c;x) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{x^k}{k!}, \quad |x| < 1,$$

with $(\cdot)_k$ being the Pochhammer symbol. Finally, $B(\cdot,\cdot)$ is the Beta function. Alegría et al. (2020) show that (2) is k times differentiable at the origin if and only if $\lfloor \nu/2 \rfloor > k$, for any τ, α, and where $\lfloor x \rfloor$ denotes the largest integer being smaller than $x \in \mathbb{R}$. From this point of view the $\mathcal{F}$ class of correlation can be viewed as the analogous of the Matérn class on the sphere. In the `GeoModels` package the $\mathcal{F}$ family is implemented as

$$\mathcal{F}_{1/\alpha, 1/\alpha+0.5, \nu}(d_{GC}) = \frac{B(1/\alpha+0.5, \nu+1/\alpha)}{B(1/\alpha+0.5,\nu)} {}_2F_1(1/\alpha, 1/\alpha+0.5, 2/\alpha+0.5+\nu; \cos(d_{GC})), \tag{3}$$

This kind of parametrization allows to solve problem of identifiability associated to (2) and the α and ν parameters are correlation range and smoothness parameters respectively as in the Matérn model.

We set the $\mathcal{F}$ correlation model with associated parameters and the other parameters of the GRF. In the `GeoModels` package the $\mathcal{F}$ model is implemented with the name of `Smoke` model.

```
corrmodel = "Smoke"    ## correlation model and parameters
scale=0.3
smooth=0.5
sill=1
nugget=0
mean=0
```

Here the `scale` parameter corresponds to α, `sill` is the variance parameter σ^2 and `smooth` is the smoothness parameter ν. The choice of $\nu = 0.5$ leads to a non mean square differentiable Gaussian random field. We are now ready to simulate a GRF on the unit sphere using the function `GeoSim`:

```
param=list(mean=mean, sill=sill, nugget=nugget, smooth=smooth,
           scale=scale)
data = GeoSim(coordx=coords, corrmodel=corrmodel, param=param,
            distance="Geod",radius=1)$data
```

Note that the option `distance="Geod"` coupled with the location sites given in lon/lat format in degimal degrees (`coordx=coords`) and a valid correlation model on the sphere (`corrmodel`) with a given `radius`, allows to perform a simulation based on Cholesky decomposition of a GRF on $\mathbb{S}^2$.

Estimation of Gaussian random fields on the sphere

Let $\rho_{\alpha,\nu}(d_{GC}) = \mathcal{F}_{1/\alpha,1/\alpha+0.5,\nu}(d_{GC})$. Given a realization $\boldsymbol{Z} = (Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2), \ldots, Z(\boldsymbol{x}_N))^T$ from a GRF defined on $\mathbb{S}^2$ with correlation (3) we can perform maximum likelihood estimation maximizing the log-likelihood function

$$l(\boldsymbol{\beta}) = -0.5log(|\sigma^2 C_{\alpha,\nu}|) - 0.5\frac{(\boldsymbol{Z}-\mu\boldsymbol{1})^T C_{\alpha,\nu}^{-1}(\boldsymbol{Z}-\mu\boldsymbol{1})}{\sigma^2}$$

with respect to $\boldsymbol{\beta} = (\mu, \sigma^2, \alpha, \nu)^T$ where $C_{\alpha,\nu} = [\rho_{\alpha,\nu}(d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j))]_{i,j=1}^N$ is the correlation matrix.

We can use the `GeoFit` function to perform maximum likelihood estimation. We first choose the fixed parameters and the parameters that must be estimated.

```
fixed<-list(nugget=nugget)
start<-list(mean=mean,scale=scale,sill=sill,smooth=smooth)
```

We consider a quasi-Newton optimization as implemented in the `nlminb` function that allows box-constrained optimization using PORT routines. We first fix the lower and upper bound of the constrained optimization:

```
optimizer="nlminb"
I=Inf
upper<-list(mean=I,scale=I,sill=I,smooth=I)
lower<-list(mean=-I,scale=0,sill=0,smooth=0)
```

We are now ready to perform maximum likelihood estimation using the `GeoFit` function. This option can be time consuming depending on the number of point on the sphere.

```
fixed<-list(nugget=nugget)
```

```
start<-list(mean=mean,scale=scale,sill=sill,smooth=smooth)
fit_geo_ml <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
                     likelihood="Full",type="Standard",
                     optimizer=optimizer,lower=lower,upper=upper,
                     start=start,fixed=fixed,distance="Geod",radius=1)
```

Default maximization algorithm used in the `GeoFit` function is the Nelder-mead method as implemented in the `R` function `optim`.

Computation of the GC distances involved in the covariance matrix can be obtained using the function `rdist.earth` of the package `fields` (Douglas Nychka et al., 2015):

```
geod_ds=rdist.earth(coords,miles=F,R=1)
max_geod=max(geod_ds)
```

Since maximum likelihood can be computationally expensive, in particular when the computation of the correlation function involves some special functions as in the Matérn of $\mathcal{F}$ cases, we can use pairwise likelihood estimation obtained by maximizing the function

$$pl(\boldsymbol{\beta}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} log(f_{Z(\boldsymbol{x}_i),Z(\boldsymbol{x}_j)}(z_i, z_j; \boldsymbol{\beta}))w_{ij} \tag{4}$$

with respect to $\boldsymbol{\beta}$. It can be performed with the following code

```
fit_geo_pl <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
          likelihood="Marginal",type="Pairwise",maxdist=max_geod/40,
          optimizer=optimizer,lower=lower,upper=upper,
          start=start,fixed=fixed,distance="Geod",radius=1)
```

In (4) $f_{Z(\boldsymbol{x}_i),Z(\boldsymbol{x}_j)}(z_i, z_j; \boldsymbol{\beta})$ is the Gaussian bivariate density and w_{ij} are non-negative weights, not depending on $\boldsymbol{\beta}$, specified as:

$$w_{ij} := \begin{cases} 1 & d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j) < d^* \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

Note that the option `maxdist=max_geod/40` sets the compact support d^* of the weight function (5) as a fraction of the maximum GC distance. A suitable choice of the weights allows to improve both the statistical and computational efficiency oh the pairwise likelihood method (Bevilacqua and Gaetan (2015)).

The two estimates can be obtained from the objects `fit_geo_ml` and `fit_geo_pl` as:

```
fit_geo_ml$param
      mean       scale        sill      smooth
-0.1614562   0.2307397   0.8679414   0.5479971
fit_geo_pl$param
      mean       scale        sill      smooth
-0.1976939   0.2081987   0.9125241   0.5743856
```

Recall that the Matérn model is not a suitable covariance model on the sphere but it is still a valid model when coupled with the chordal distance. In the `GeoModels` package, estimation can be performed using also the CH distance, or the Euclidean distance after a suitable projection of the spherical coordinates. Computation of CH distances involved in the covariance matrix can be obtained using (1), that is:

```
chor_ds=2*sin(geod_ds/2)
max_chor=max(chor_ds)
```

Then we can use pairwise likelihood estimation using chordal distances (setting `distance="Chor"`) in order to estimate a Matérn covariance model:

```
fit_chor_pl <- GeoFit(data=data,coordx=coords,corrmodel="Matern",
           likelihood="Marginal",type="Pairwise",maxdist=max_chor/40,
           optimizer=optimizer,lower=lower,upper=upper,
           start=start,fixed=fixed,distance="Chor",radius=1)
```

We can also consider pairwise likelihood estimation of the Matérn model using a suitable projection of the spherical points coupled with the Euclidean distance. We use the function `mapprojec` of the package `mapproj` (McIlroy et al., 2017) in order to obtain projected coordinates using sinusoidal projection

```
prj=mapproject(coords[,1], coords[,2], projection="sinusoidal")
coords_prj=cbind(prj$x,prj$y)
```

and we can visualize the projected points (Figure 2) with the following code:

```
sinusoidal.proj = map(database= "world", ylim=c(-90,90),
         xlim=c(-180,180), col="grey80", fill=TRUE,
         plot=FALSE, projection="sinusoidal")
map(sinusoidal.proj)
points(coords_prj,pch=20,cex=0.4)
```

Other types of projections can be chosen in the package `mapproj` such us, for instance, the mercator or cylindrical projections.

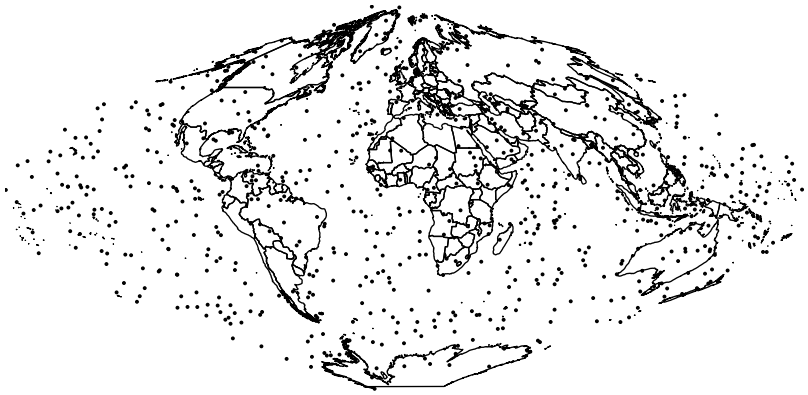

Figure 2: Sinusoidal projection

Then, we can estimate with pairwise likelihood using Euclidean distances:

```
eucl_ds=dist(coords_prj)
max_eucl=max(eucl_ds)
fit_eucl_pl <- GeoFit(data=data,coordx=coords_prj,corrmodel="Matern",
            maxdist=max_eucl/40,likelihood="Marginal",type="Pairwise",
            optimizer=optimizer,lower=lower,upper=upper,
            start=start,fixed=fixed,distance="Eucl",radius=1)
```

The two estimates of the Matérn model using chordal and euclidean distances can be obtained from the objects `fit_chor_ml` and `fit_uecl_pl` as:

```
fit_chor_pl$param
```

```
      mean       scale        sill      smooth
-0.2866617   0.1993616   0.8901290   0.4896472
 fit_eucl_pl$param
      mean       scale        sill      smooth
-0.2031913   0.2096906   0.9227965   0.4293301
```

Boxplots of the distances (Figure 4) can be useful to depict the different behaviour of the three type of distances in this specific example:

```
boxplot(c(geod_ds),c(chor_ds),c(eucl_ds),
           names=c("Greatcircle","Chordal","Euclidean"))
```

Given the estimation of the mean and variance parameters, the estimated residuals

$$\widehat{Y}(\boldsymbol{x}_i) = \frac{Z(\boldsymbol{x}_i) - \hat{\mu}}{(\widehat{\sigma}^2)^{\frac{1}{2}}} \quad i = 1, \ldots, N$$

can be viewed as a realization of a zero mean unit variance spherical GRF with correlation model (3). Using pairwise likelihood estimates, the residuals can be computed using the `GeoResiduals` function:

```
res=GeoResiduals(fit_geo_pl)
```

Then the marginal distribution assumption on the residuals can be graphically checked with a qq-plot (Figure 3, left part) using the function `GeoQQ`.

```
### checking model assumptions: marginal distribution
GeoQQ(res)
```

The correlation model assumption can be graphically checked comparing the empirical and the estimated semivariogram functions using the `GeoVariogram` and `GeoCovariogram` functions (Figure 3, right part):

```
### checking model assumptions: variogram model
vario = GeoVariogram(data=res$data,coordx=coords,
           maxdist=max_geod/2,distance="Geod",radius=1)
GeoCovariogram(res,vario=vario,show.vario=TRUE,pch=20)
```

Prediction of Gaussian random fields on the sphere

For a given location $\boldsymbol{x}_0 \in \mathbb{S}^2$, the optimal prediction in this example is computed by:

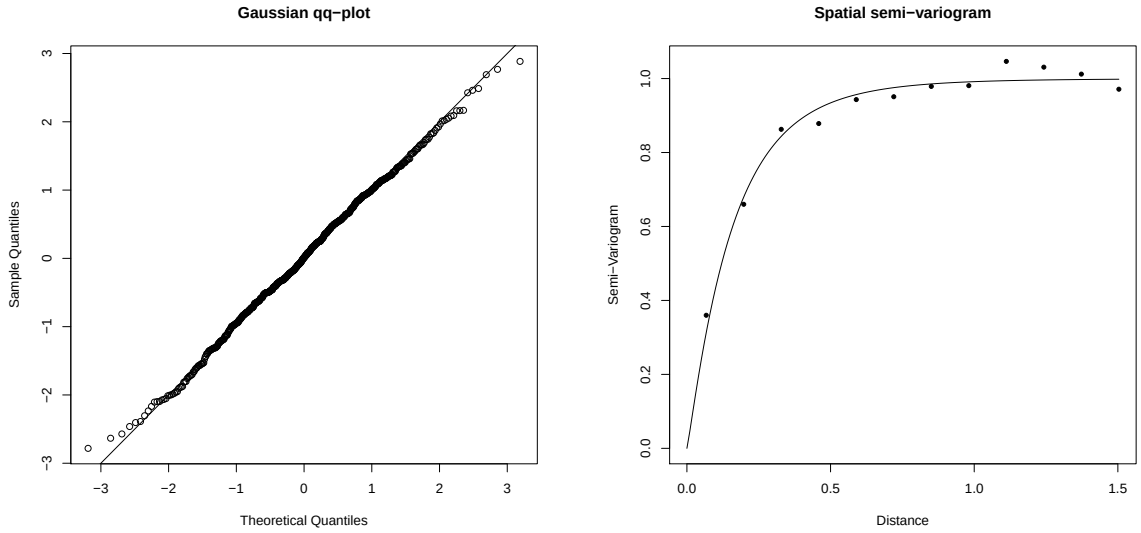

Figure 3: Left: QQ-plot for the model residuals. Right: empirical vs estimated semi-variogram function for the residuals

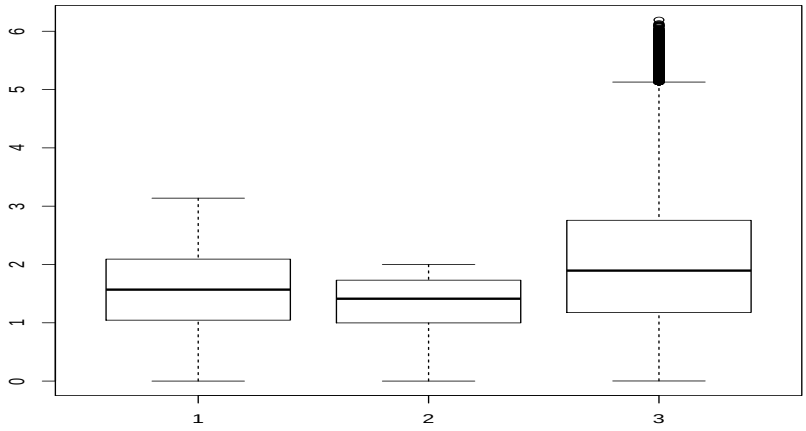

Figure 4: Boxplot of great circle, chordal and euclidean distances (from left to right)

$$\widehat{Z}(\boldsymbol{x}_0) = \hat{\mu} + \sum_{i=1}^{N} \hat{\lambda}_i [Z(\boldsymbol{x}_i) - \hat{\mu}] \tag{6}$$

where $\hat{\mu}$ is a mean estimation and the vector of weights $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_1, \ldots, \hat{\lambda}_N)'$ is given by $\hat{\boldsymbol{\lambda}} = \widehat{C}^{-1}_{\hat{\alpha},\hat{\nu}} \hat{\boldsymbol{c}}_{\hat{\alpha},\hat{\nu}}$ where $\widehat{C}_{\hat{\alpha},\hat{\nu}}$ is the estimated correlation matrix and $\hat{\boldsymbol{c}}_{\hat{\alpha},\hat{\nu}}$ is the estimated correlation between the locations and the point to predict. Moreover, the associated mean

squared error is computed through

$$MSE(\widehat{Z}(\boldsymbol{x}_0)) = \hat{\sigma}^2(1 - \hat{\boldsymbol{c}}_{\hat{\alpha},\hat{\nu}}^T \widehat{C}_{\hat{\alpha},\hat{\nu}}^{-1} \hat{\boldsymbol{c}}_{\hat{\alpha},\hat{\nu}}). \tag{7}$$

Optimal prediction and associated MSE can be computed using the `GeoKrig` function. Suppose we want to predict the GRF around the north pole area. First we need to specify the spherical location to predict (in lon/lat format):

```
## location to predict given in lon/lat format
loc_to_pred=as.matrix(expand.grid(seq(-180,180,2),seq(60,90,1)))
```

Then, the optimal predictor (6) and associated MSE (7) (using the maximum pairwise likelihood estimated parameters) can be performed with the following code:

```
param_est<-as.list(c(fixed,fit_geo_pl$param))
## computing optimal prediction. and associated MSE
pr_geo=GeoKrig(data=data,loc=loc_to_pred,coordx=coords,
    corrmodel=corrmodel,radius=1,distance="Geod",
    param=param_est,mse=TRUE))
```

The matrix of locations to predict, predictions and associated MSE can be obtained as:

```
predictions=cbind(loc_to_pred,pr_geo$pred,pr_geo$mse)
head(predictions)
    Var1 Var2
[1,] -180   60 -0.01367512 0.4959492
[2,] -178   60 -0.02045909 0.4952045
[3,] -176   60 -0.02782939 0.4851513
[4,] -174   60 -0.02756870 0.4615931
[5,] -172   60 -0.01173698 0.4202640
[6,] -170   60  0.02640055 0.3573138
```

and a graphical visualization (Figure 5) of the prediction and associated MSE can be obtained with the following code:

```
globeearth(eye=place("newyorkcity"))
globepoints(loc=loc_to_pred,pch=20,
col =  heat.colors(length(pr_geo$pred),alpha=0.1)[rank(pr_geo$pred)])
##
globeearth(eye=place("newyorkcity"))
globepoints(loc=loc_to_pred,pch=20,
col =  cm.colors(length(pr_geo$mse),alpha=0.1)[rank(pr_geo$mse)])
```

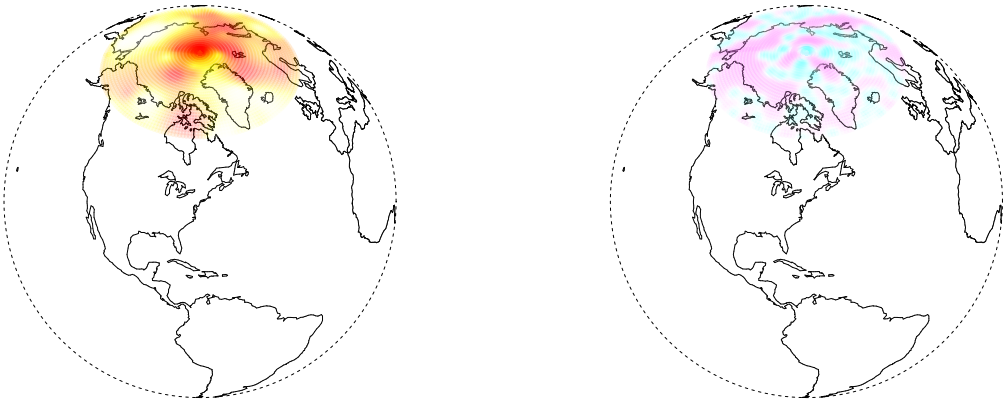

Figure 5: From left to right: prediction and associated MSE in the north pole area

References

Alegría, A., F. Cuevas, P. Diggle, and E. Porcu (2020). A family of covariance functions for random fields on spheres. *https://math.au.dk/forskning/publikationer/instituttets-serier/publication/publid/1126/*.

Baddeley, A., T. Lawrence, and E. Rubak (2017). *globe: Plot 2D and 3D Views of the Earth, Including Major Coastline*. R package version 1.2-0.

Bevilacqua, M. and C. Gaetan (2015). Comparing composite likelihood methods based on pairs for spatial Gaussian random fields. *Statistics and Computing 25*, 877–892.

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

Douglas Nychka, Reinhard Furrer, John Paige, and Stephan Sain (2015). fields: Tools for spatial data. R package version 9.0.

Gneiting, T. (2013). Strictly and non-strictly positive definite functions on spheres. *Bernoulli 19*, 1327–1349.

McIlroy, D., R. Brownrigg, T. Minka, and R. Bivand. (2017). *mapproj: Map Projections*. R package version 1.2-5.

Porcu, E., M. Bevilacqua, and M. Genton (2016). Spatio-temporal covariance and cross covariance functions of the great circle distance on a sphere. *Journal of the American Statistical Association 111*, 888–898.

Robotham, A. (2013). *sphereplot: Spherical plotting*. R package version 1.5.