01-meuse-INLA

Frances Lin

3/28/2022

The objective of this project is to get familiar with and compare kriging (uk) and spde (stochastic partial differential equation) via INLA.

Load packages

```
library(gstat) # data
library(INLA)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loading required package: parallel

## Loading required package: sp

## This is INLA_22.03.16 built 2022-03-16 13:24:07 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x purrr::accumulate() masks foreach::accumulate()
## x tidyr::expand()     masks Matrix::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x tidyr::pack()       masks Matrix::pack()
## x tidyr::unpack()     masks Matrix::unpack()
## x purrr::when()       masks foreach::when()
```

```
library(pander)
library(sp) # will retire
library(maptools) # for unionSpatialPolygons
```

```
## Checking rgeos availability: TRUE
## Please note that 'maptools' will be retired by the end of 2023,
## plan transition at your earliest convenience;
## some functionality will be moved to 'sp'.
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(here)
```

```
## here() starts at /Users/franceslinyc/INLA-with-Spatial-data-2022
```

Load data

```
data(meuse)
```

```
class(meuse)
```

```
## [1] "data.frame"
```

```
meuse %>% head(3) %>% pander
```

Table 1: Table continues below

x	y	cadmium	copper	lead	zinc	elev	dist	om
181072	333611	11.7	85	299	1022	7.909	0.001358	13.6
181025	333558	8.6	81	277	1141	6.983	0.01222	14
181165	333537	6.5	68	199	640	7.8	0.103	13

ffreq	soil	lime	landuse	dist.m
1	1	1	Ah	50
1	1	1	Ah	30
1	1	1	Ah	150

```
meuse %>% dim
```

```
## [1] 155  14
```

Kriging

Create a SpatialPointsDataFrame (the sp way) and assign CSR

```
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS("+init=epsg:28992")
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj = prefer_proj): Discar
##  but +towgs84= values preserved
```

```
class(meuse)
```

```
## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"
```

Do the same things for grid (prediction grid for meuse data)

```
data(meuse.grid)
coordinates(meuse.grid) = ~x+y
proj4string(meuse.grid) <- CRS("+init=epsg:28992")
gridded(meuse.grid) = TRUE # Not sure.
```

Compute the empirical variogram and fit a spherical variogram

The variogram and kriging sections for this project are meant to be brief. Instead, the focus will be on expanding the INLA section.

```
vgm <- variogram(log(zinc) ~ dist, meuse)
fit.vgm <- fit.variogram(vgm, vgm("Sph"))
fit.vgm
```

```
##   model      psill    range
## 1   Nug 0.07642515   0.0000
## 2   Sph 0.20529402 728.6969
```

Sill (variance) = mean of psill = 0.14086. Nugget (variance when distance = 0) = min of psill = 0.07643. Range (distance when the curve first to flattens out) = mean of range = 364.3.

```
vgm_summary <- summary(fit.vgm)
vgm_summary
```

```
##      model        psill             range           kappa            ang1
##  Nug    :1   Min.   :0.07643   Min.   :  0.0   Min.   :0.000   Min.   :0
##  Sph    :1   1st Qu.:0.10864   1st Qu.:182.2   1st Qu.:0.125   1st Qu.:0
##  Exp    :0   Median :0.14086   Median :364.3   Median :0.250   Median :0
##  Gau    :0   Mean   :0.14086   Mean   :364.3   Mean   :0.250   Mean   :0
##  Exc    :0   3rd Qu.:0.17308   3rd Qu.:546.5   3rd Qu.:0.375   3rd Qu.:0
##  Mat    :0   Max.   :0.20529   Max.   :728.7   Max.   :0.500   Max.   :0
##  (Other):0
##       ang2         ang3        anis1        anis2
##  Min.   :0   Min.   :0   Min.   :1   Min.   :1
##  1st Qu.:0   1st Qu.:0   1st Qu.:1   1st Qu.:1
##  Median :0   Median :0   Median :1   Median :1
##  Mean   :0   Mean   :0   Mean   :1   Mean   :1
##  3rd Qu.:0   3rd Qu.:0   3rd Qu.:1   3rd Qu.:1
##  Max.   :0   Max.   :0   Max.   :1   Max.   :1
##
```

```
# Write to the results folder
write_rds(fit.vgm, here("results", "fit.vgm.rds"))
write_rds(vgm_summary, here("results", "vgm_summary.rds"))
```

Fit the (universal) kriging model

```
krg <- krige(log(zinc) ~ dist, meuse, meuse.grid, model = fit.vgm)
```

```
## [using universal kriging]
```

```
krg %>% as.data.frame %>% head %>% pander
```

x	y	var1.pred	var1.var
181180	333740	6.737	0.2022
181140	333700	6.785	0.1766
181180	333700	6.699	0.1842
181220	333700	6.559	0.1922
181100	333660	6.841	0.1484
181140	333660	6.749	0.1567

```
#summary(krg)
```

Visualize the results of UK model

```
# Add estimates to meuse.grid
meuse.grid$zinc.krg <- krg$var1.pred
meuse.grid$zinc.krg.sd <- sqrt(krg$var1.var)
```

Results show that higher concentrations of (log) zinc in points closer to the Meuse river.

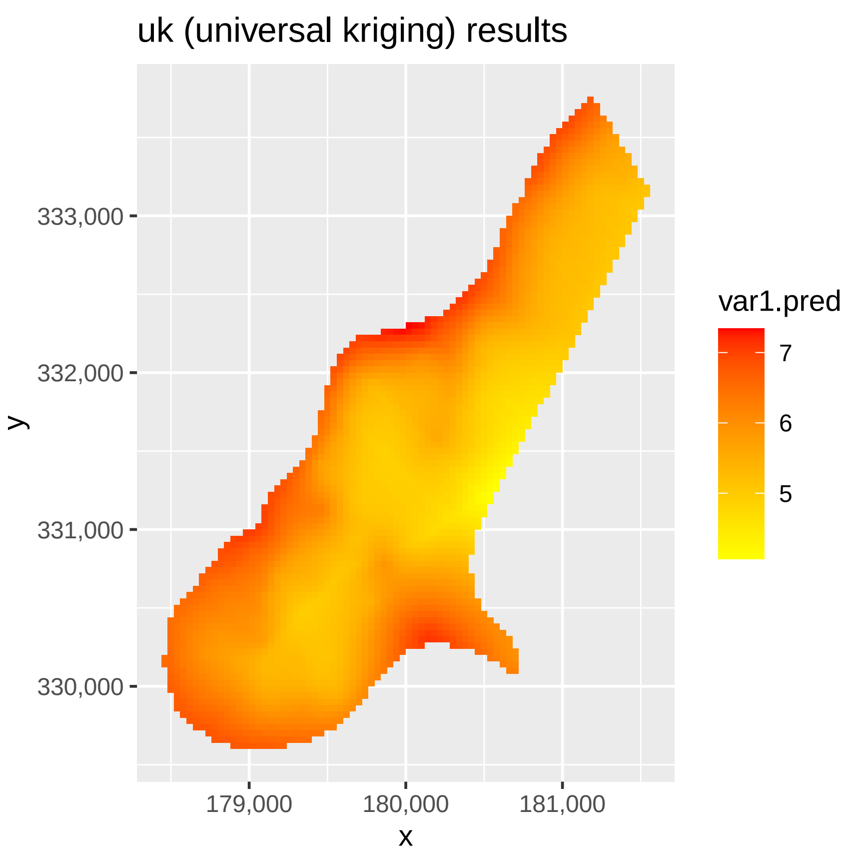
```r
# Visualize the results of uk
library(scales) # for comma
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
krg %>% as.data.frame %>%
  ggplot(aes(x=x, y=y)) + geom_tile(aes(fill=var1.pred)) + coord_equal() +
  scale_fill_gradient(low = "yellow", high="red") +
  scale_x_continuous(labels=comma) +
  scale_y_continuous(labels=comma) + # customise to add commma
  labs(title = "uk (universal kriging) results") -> p_uk
p_uk
```



Spatial Models using SPED (Stochastic Partial Differential Equations)

A spatial process with a Matérn covariance can be obtained as the weak solution to a stochastic partial differential equation (SPDE, Lindgren et al., 2011).

It involves the following steps:

1. Create a mash
2. Make the latent model
3. Make an A matrix
4. Organize the data
5. Estimate or predict

Preprocess data for INLA

A mesh needs to be defined over the study region and it will be used to compute the approximation to the solution (i.e., the spatial process).

Define the boundary of the study region

```
# Define the boundary
meuse.bdy <- unionSpatialPolygons(
  as(meuse.grid, "SpatialPolygons"), rep(1, length(meuse.grid))
)
```

1. Create a mash / define a two-dimensional mesh to define the set of basis functions

```
pts <- meuse.bdy@polygons[[1]]@Polygons[[1]]@coords
# Create a mesh
mesh <- inla.mesh.2d(loc.domain = pts,
                     max.edge = c(150, 500),
                     offset = c(100, 250) )
```

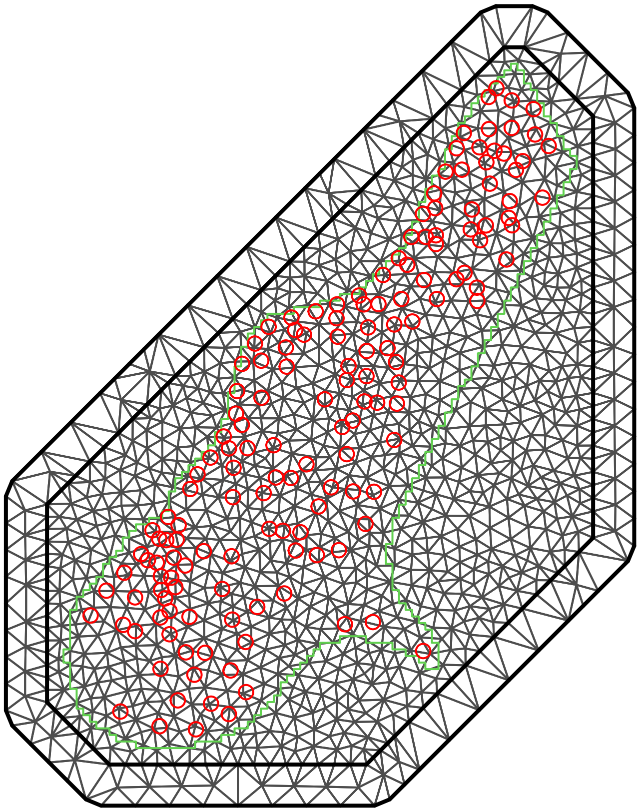
Plot it

```
# So that we can plot observations too!
coo <- coordinates(meuse)
```

```
par(mar = c(0, 0, 0, 0))
plot(mesh, asp = 1, main = "")
lines(pts, col = 3, with = 2)
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "with" is not a graphical
## parameter
```

```
points(coo, col = "red") # x & y in meuse
```



I am still not quite sure what it is. Let's break them into steps.

```
mesh$n # # of vertices
```

```
## [1] 1240
```

```
# par(mar = c(0, 0, 0, 0))
# plot(mesh, asp = 1, main = "")
```

Plot them side-by-side

```
# par(mfrow=c(2,2)) # This turns out to be really bad. Try ggplot2 instead.
# # Plot1
# bubble(meuse, "zinc", main = "zinc concentrations (ppm)")
# # Plot2
# plot(mesh, asp = 1, main = "")
# lines(pts, col = 3, with = 2)
# points(coo, col = "red") # x & y in meuse
```

Plot it using ggplot2

```
class(mesh)
```

```
## [1] "inla.mesh"
```

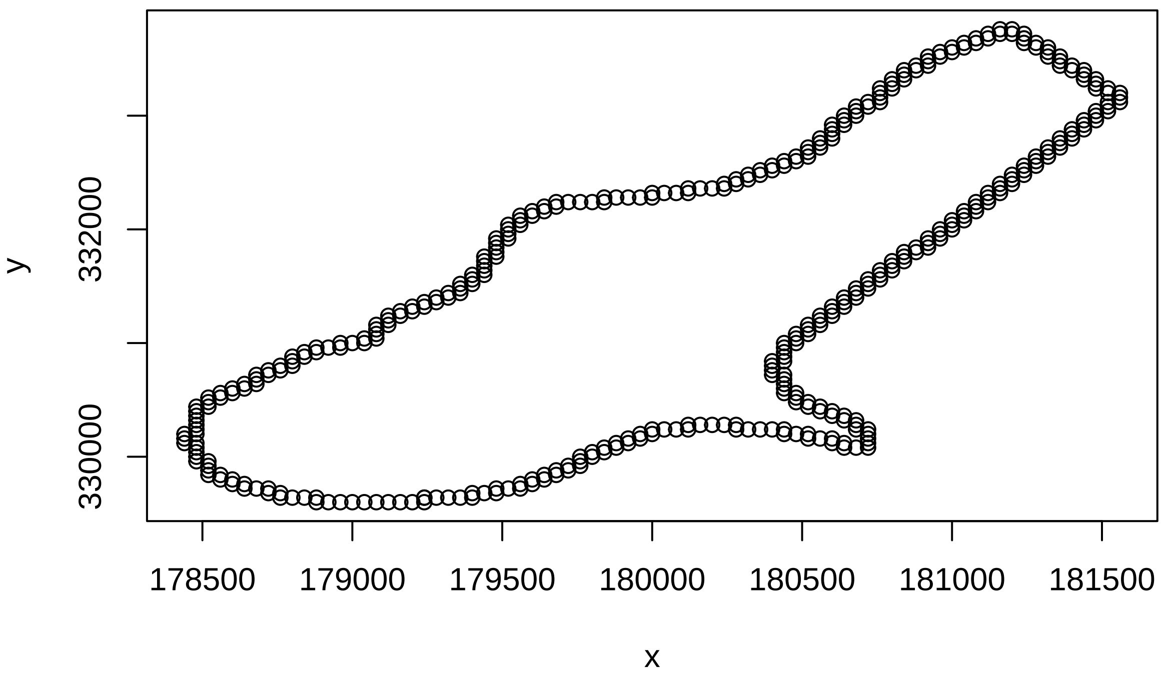
I am going to skip this for now but check this and this later.

```
meuse %>% as.data.frame %>%
  ggplot(aes(x, y)) + geom_point(aes(size=zinc), alpha=3/4) +
  ggtitle("Zinc Concentration (ppm)") +
  coord_equal() -> p
```

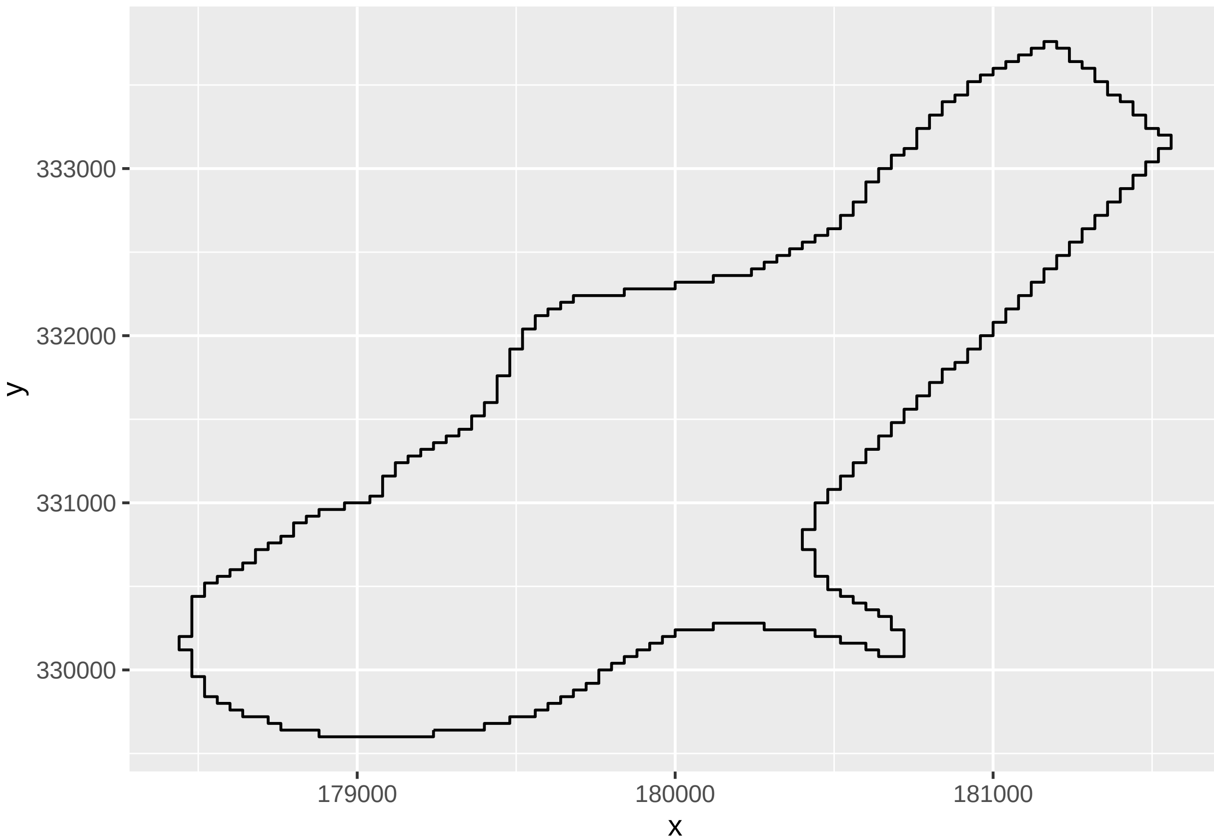
```
class(pts)
```

```
## [1] "matrix" "array"
```

```
plot(pts)
```

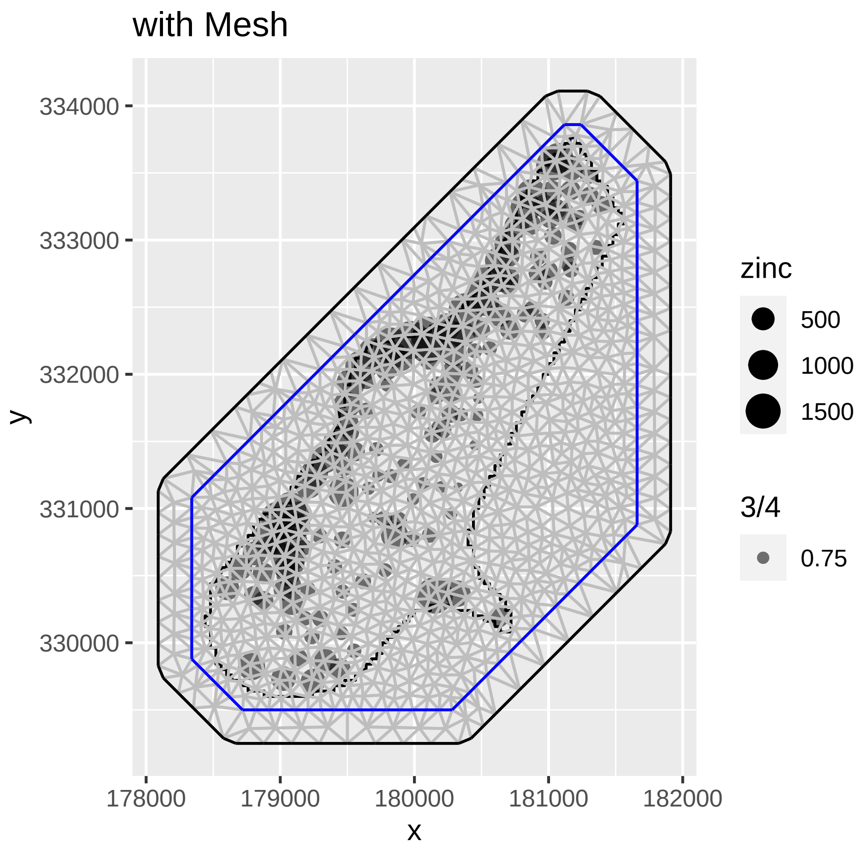



```
pts %>% as.data.frame %>%
  ggplot(aes(x, y)) + geom_path() #geo_line doesn't work.
```



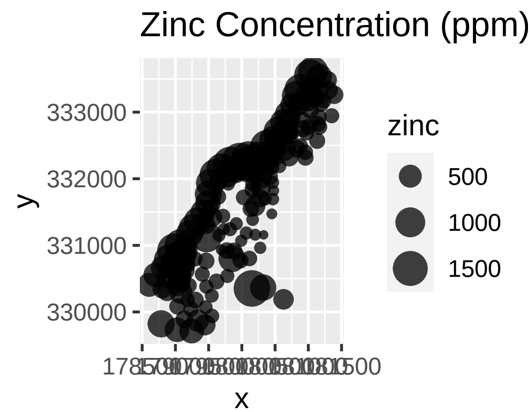

```r
# Convert to df for plotting
pts_df <- pts %>% as.data.frame
meuse_df <- meuse %>% as.data.frame
```

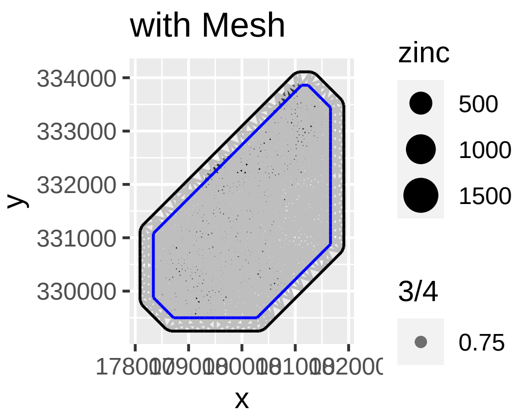
```r
library(inlabru)
# Use neither as the default
ggplot(NULL, aes(x, y)) +
  geom_point(data = meuse_df, aes(size=zinc, alpha=3/4)) +
  geom_path(data = pts_df) +
  gg(mesh) +
  ggtitle("with Mesh") +
  coord_equal() -> p_mesh
p_mesh
```




```
# library(inlabru)
# meuse %>% as.data.frame %>% # Boundary line left to add.
#   ggplot(aes(x, y)) + geom_point(aes(size=zinc), alpha=3/4) +
#   gg(mesh) +
#   ggtitle("with Mesh") +
#   coord_equal() -> p_mesh
# p_mesh
```

```
# Plot them
grid.arrange(p, p_mesh, nrow = 2)
```





2. Make the latent model / create a SPDE (stochastic partial differential equation) model

A spatial process with a Matérn covariance can be obtained as the weak solution to a stochastic partial differential equation (SPDE) (Lindgren et al., 2011).

```
# Build a SPDE model
meuse.spde <- inla.spde2.matern(mesh = mesh, alpha = 2) # alpha = smoothness parameter

# Generate the index set for the SPDE model
s.index <- inla.spde.make.index(name = "spatial.field",
                                n.spde = meuse.spde$n.spde)
```

3. Make an A matrix

```
# Construct a projection matrix A to project the GRF from the observations to the triangulation vertices
A.meuse <- inla.spde.make.A(mesh = mesh, loc = coordinates(meuse)) # Use meuse for estimation

# Construct another projection matrix A for prediction
A.pred <- inla.spde.make.A(mesh = mesh, loc = coordinates(meuse.grid)) # Use meuse.grid for prediction
```

4. Organize the data used for estimation (model fitting) or prediction

```
# Organize the data, projection matrices and effects for estimation
meuse.stack <- inla.stack(data = list(zinc = meuse$zinc),
                          A = list(A.meuse, 1),
                          effects = list(c(s.index, list(Intercept = 1)),
                                         list(dist = meuse$dist)),
                          tag = "meuse.data") # Change to meuse.est?

# Organize the data for prediction
meuse.stack.pred <- inla.stack(data = list(zinc = NA),
                               A = list(A.pred, 1),
                               effects = list(c(s.index, list (Intercept = 1)),
                                              list(dist = meuse.grid$dist)),
                               tag = "meuse.pred")
```

Join stacks of data into a single object

```
join.stack <- inla.stack(meuse.stack, meuse.stack.pred)
```

Fit the spatial model

Other models that can go into the `model =` argument in `f()` include, for example, `spde` (Matèrn correlation (continuous)), `matern2d` (Matèrn correlation (discrete)), `besag` (Intrinsic CAR), `besagproper` (Proper CAR), and `bym` (Convolution). We will explore this more in the future project.

```
# Specify the formula
form <- log(zinc) ~ -1 + Intercept + dist + f(spatial.field, model = spde) # - 1 removes intercept
```

```
# Fit the model
start.time <- Sys.time()
m1 <- inla(form,
           data = inla.stack.data(join.stack, spde = meuse.spde),
           family = "gaussian",
           control.predictor = list(A = inla.stack.A(join.stack), compute = TRUE),
           control.compute = list(cpo = TRUE, dic = TRUE))
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken # Time difference of 9.82935 secs
```

```
## Time difference of 10.06204 secs
```

According to this site, the main arguments of the `inla()` function are

- `formula`:
- `data`:
- `family`:
- `control.predictor`:
- `control.compute`: .

Print summary results

```
summary(m1)
```

```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts,
##    ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##    scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##    ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##    verbose, ", " lincomb = lincomb, selection = selection, control.compute
##    = control.compute, ", " control.predictor = control.predictor,
##    control.family = control.family, ", " control.inla = control.inla,
##    control.fixed = control.fixed, ", " control.mode = control.mode,
##    control.expert = control.expert, ", " control.hazard = control.hazard,
##    control.lincomb = control.lincomb, ", " control.update =
##    control.update, control.lp.scale = control.lp.scale, ", "
##    control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##    ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##    num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
##    working.directory = working.directory, ", " silent = silent, inla.mode
##    = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
##    .parent.frame)")
## Time used:
##     Pre = 3.22, Running = 6.3, Post = 0.362, Total = 9.88
## Fixed effects:
##             mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## Intercept  6.599 0.168      6.269    6.596      6.945  6.590   0
## dist      -2.778 0.413     -3.588   -2.781     -1.946 -2.787   0
##
## Random effects:
##   Name     Model
##     spatial.field SPDE2 model
##
## Model hyperparameters:
##                                          mean    sd 0.025quant 0.5quant
## Precision for the Gaussian observations 13.17 3.293       7.87    12.78
## Theta1 for spatial.field                 4.75 0.265       4.22     4.75
## Theta2 for spatial.field                -5.27 0.287      -5.83    -5.27
##                                         0.975quant  mode
## Precision for the Gaussian observations      20.71 12.03
## Theta1 for spatial.field                      5.27  4.75
## Theta2 for spatial.field                     -4.70 -5.27
##
## Deviance Information Criterion (DIC) ...............: 117.67
## Deviance Information Criterion (DIC, saturated) ....: 1942.26
## Effective number of parameters .....................: 69.56
##
## Marginal log-Likelihood:  -109.94
## CPO, PIT is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Extract summary of the fixed effects

```
m1$summary.fixed
```

```
##                mean        sd 0.025quant  0.5quant 0.975quant      mode
## Intercept  6.598907 0.1679872   6.268934  6.595992   6.944666  6.590340
## dist      -2.777897 0.4134087  -3.588160 -2.781450  -1.945716 -2.787004
##                    kld
## Intercept 3.022786e-10
## dist      5.406850e-11
```

Extract summary of the random effects

```
#m1$summary.random
```

Extract summary of the hyperparameters

```
m1$summary.hyperpar
```

```
##                                              mean        sd 0.025quant
## Precision for the Gaussian observations 13.172939 3.2934096   7.867830
## Theta1 for spatial.field                 4.747907 0.2654742   4.221468
## Theta2 for spatial.field                -5.265509 0.2873386  -5.827243
##                                          0.5quant 0.975quant      mode
## Precision for the Gaussian observations 12.778291  20.709107 12.028965
## Theta1 for spatial.field                 4.749203   5.268014  4.753995
## Theta2 for spatial.field                -5.266917  -4.698619 -5.271710
```

Extract results for comparision

```
# Compute statistics in terms or range and variance
spde.est <- inla.spde2.result(inla = m1,
                              name = "spatial.field",
                              spde = meuse.spde,
                              do.transf = TRUE)
```

Values differ a bit.

```
# Variance to compare to sill
variance <- inla.zmarginal(spde.est$marginals.variance.nominal[[1]])
```

```
## Mean            0.235154
## Stdev           0.0723044
## Quantile  0.025 0.124524
## Quantile  0.25  0.183101
## Quantile  0.5   0.224479
## Quantile  0.75  0.275397
## Quantile  0.975 0.406179
```

```r
variance
```

```
## $mean
## [1] 0.2351541
##
## $sd
## [1] 0.07230442
##
## $quant0.025
## [1] 0.1245239
##
## $quant0.25
## [1] 0.183101
##
## $quant0.5
## [1] 0.2244793
##
## $quant0.75
## [1] 0.2753969
##
## $quant0.975
## [1] 0.4061787
```

```r
# Range
range <- inla.zmarginal(spde.est$marginals.range.nominal[[1]])
```

```
## Mean            570.154
## Stdev           165.297
## Quantile  0.025 311.206
## Quantile  0.25  451.037
## Quantile  0.5   547.878
## Quantile  0.75  664.417
## Quantile  0.975 955.666
```

```r
range
```

```
## $mean
## [1] 570.1545
##
## $sd
## [1] 165.2974
##
## $quant0.025
## [1] 311.2056
##
## $quant0.25
## [1] 451.0375
##
## $quant0.5
## [1] 547.8777
##
## $quant0.75
```

```
## [1] 664.4169
##
## $quant0.975
## [1] 955.6665
```

```
# Write to the results folder
write_rds(variance, here("results", "variance.rds"))
write_rds(range, here("results", "range.rds"))
```

The `inlabru` package (Bachl et al., 2019) can simplify the way in which the model is defined and fit.

Plot it

```
class(krg)
```

```
## [1] "SpatialPixelsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
class(m1)
```

```
## [1] "inla"
```

```
#krg
```

```
#m1
```

This might not work but I am gonna try.

```
spde <- m1
```

```
#spde %>% as.data.frame # Does not work.
```

```
# Load spatial domain to interpolate over
data("meuse.grid") # What is this???
```

This section needs to be rewritten to be more organized.

```
# For krg
meuse.grid$zinc.krg <- krg$var1.pred
```

```
# For spde
# Obtain the indices of thr rows corresponding to the predictions
index.pred <- inla.stack.index(join.stack, tag = "meuse.pred")$data
# Create a variable zinc.spde with the posterior mean
meuse.grid$zinc.spde <- spde$summary.fitted.values[index.pred, "mean"]
```

```
meuse.grid$zinc.spde.ll <- spde$summary.fitted.values[index.pred, "0.025quant"] # lower limit of 95% cr
meuse.grid$zinc.spde.ul <- spde$summary.fitted.values[index.pred, "0.975quant"]
```

```
meuse.grid %>% head(3) %>% pander
```

Table 4: Table continues below

x	y	part.a	part.b	dist	soil	ffreq	zinc.krg
181180	333740	1	0	0	1	1	6.737
181140	333700	1	0	0	1	1	6.785
181180	333700	1	0	0.01222	1	1	6.699

zinc.spde	zinc.spde.ll	zinc.spde.ul
6.724	5.971	7.484
6.78	6.139	7.425
6.691	5.949	7.431

```
krg %>% as.data.frame %>%
  ggplot(aes(x=x, y=y)) + geom_tile(aes(fill=var1.pred)) + coord_equal() +
  scale_fill_gradient(low = "yellow", high="red") +
  scale_x_continuous(labels=comma) +
  scale_y_continuous(labels=comma) + # customize to add commma
  labs(title = "uk (universal kriging) results") -> p_uk
#p_uk
```

```
# Plot for krg
meuse.grid %>% as.data.frame %>%
  ggplot(aes(x=x, y=y)) + geom_tile(aes(fill=zinc.krg)) + coord_equal() +
  scale_fill_gradient(low = "yellow", high="red") +
  scale_x_continuous(labels=comma) +
  scale_y_continuous(labels=comma) +
  labs(title = "uk results") -> p_krg
#p_krg # This should match the plot from above.
```

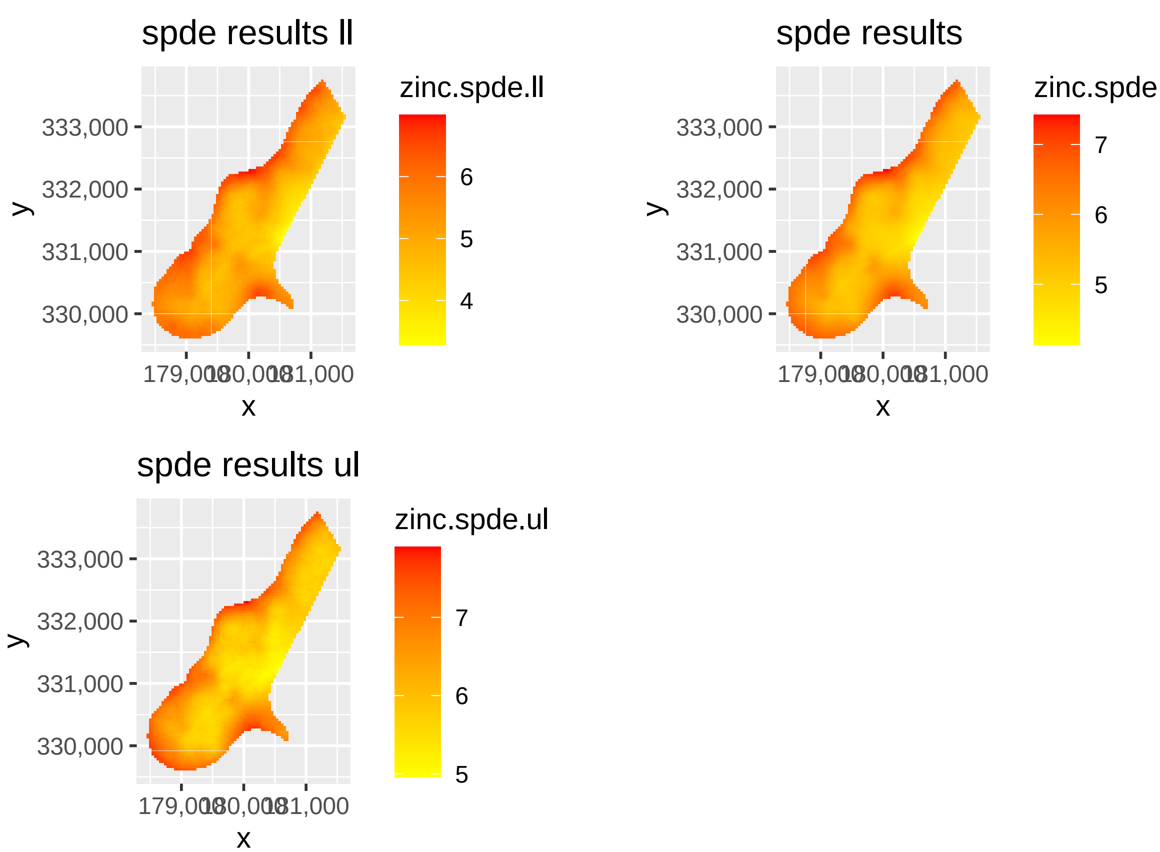
```
# Plot for spde
meuse.grid %>% as.data.frame %>%
  ggplot(aes(x=x, y=y)) + geom_tile(aes(fill=zinc.spde)) + coord_equal() +
  scale_fill_gradient(low = "yellow", high="red") +
  scale_x_continuous(labels=comma) +
  scale_y_continuous(labels=comma) +
  labs(title = "spde results") -> p_spde
#p_spde
```

Results of spde with credible intervals

```r
# For ll
meuse.grid %>% as.data.frame %>%
  ggplot(aes(x=x, y=y)) + geom_tile(aes(fill=zinc.spde.ll)) + coord_equal() +
  scale_fill_gradient(low = "yellow", high="red") +
  scale_x_continuous(labels=comma) +
  scale_y_continuous(labels=comma) +
  labs(title = "spde results ll") -> p_spde_ll

# For ul
meuse.grid %>% as.data.frame %>%
  ggplot(aes(x=x, y=y)) + geom_tile(aes(fill=zinc.spde.ul)) + coord_equal() +
  scale_fill_gradient(low = "yellow", high="red") +
  scale_x_continuous(labels=comma) +
  scale_y_continuous(labels=comma) +
  labs(title = "spde results ul") -> p_spde_ul
```

```r
grid.arrange(p_spde_ll, p_spde, p_spde_ul, ncol = 2)
```

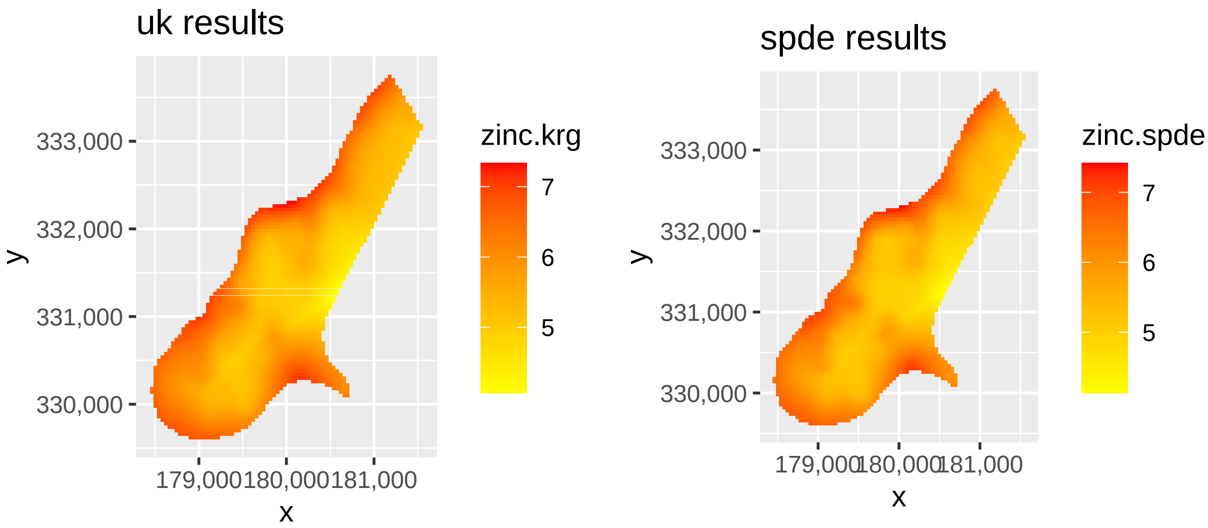


We will come back to explore more and add additional details.

Compare results of uk vs spde

Plot results of uk vs spde

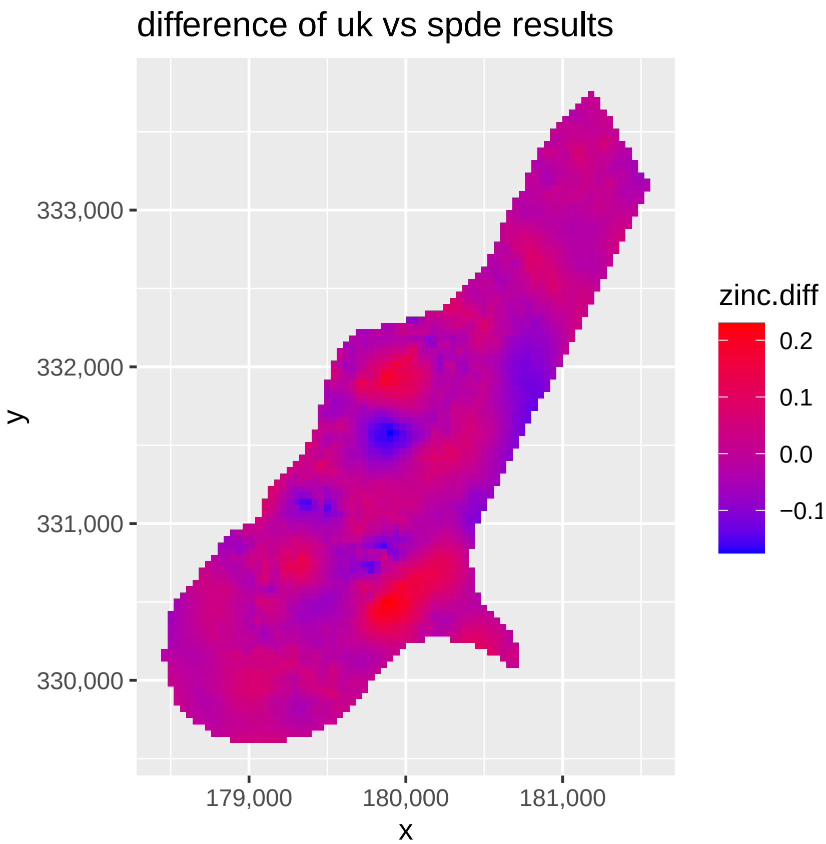
```
# Plot them side-by-side
grid.arrange(p_krg, p_spde, ncol = 2)
```



Plot difference plot of uk vs spde

Areas where it's colored in either red or blue are where `krg` results & `spde` results differ the most since we set `zinc.diff` = `krg` results - `spde` results.

```
p_diff <- meuse.grid %>% as.data.frame %>%
  mutate(
    zinc.diff = zinc.krg - zinc.spde
  ) %>%
  ggplot(aes(x=x, y=y)) + geom_tile(aes(fill=zinc.diff)) + coord_equal() +
  scale_fill_gradient(low = "blue", high="red") +
  scale_x_continuous(labels=comma) +
  scale_y_continuous(labels=comma) +
  labs(title = "difference of uk vs spde results")
p_diff
```




```
# Write to the results folder
write_rds(p_krg, here("results", "p_krg.rds"))
write_rds(p_spde, here("results", "p_spde.rds"))
write_rds(p_diff, here("results", "p_diff.rds"))
```

Reference

Gómez-Rubio, V. (2020). 7.3 Geostatistics. *Bayesian inference with INLA.* CRC Press.

Moraga, P. (2019). 8 Geostatistical data. *Geospatial health data: Modeling and visualization with R-INLA and shiny.* CRC Press.