

```

library(sp)
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##   filter
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(FRK)
# devtools::load_all("~/Wollongong/pkgs/FRK",
#                       export_all = FALSE)

opts_FRK$set("progress",FALSE)
opts_FRK$set("parallel",0L)
set.seed(1)

## Get data
data(meuse)
meuse$fs <- 1
coordinates(meuse) = ~x+y # change into an sp object

## Set up BAUs
data(meuse.grid)
gridded(meuse.grid) = ~x + y
HexPts <- spsample(meuse.grid,
                   type = "hexagonal",
                   cellsize = 50)
HexPols <- HexPoints2SpatialPolygons(HexPts)
HexPols_df <- SpatialPolygonsDataFrame(HexPols,
                                       cbind(over(HexPols,meuse.grid),
                                             coordinates(HexPts)))

HexPols_df$fs <- 1
#HexPols_df <- subset(HexPols_df,!is.na(dist))

# Generate observations with large spatial support
HexPts2 <- spsample(meuse.grid,
                   type = "hexagonal",
                   cellsize = 100)
HexPols2 <- HexPoints2SpatialPolygons(HexPts2)

```

```

HexPols_df2 <- SpatialPolygonsDataFrame(HexPols2,
                                         over(HexPols2,meuse) %>%
                                         select(zinc)) %>%

  subset(!is.na(zinc))

## Generate basis functions
G <- auto_basis(m = plane(),data=meuse,nres = 2,
               prune=10,type = "Gaussian")

## Loading required package: INLA
## Loading required package: Matrix
## Loading required package: splines
## Loading required package: splancs
##
## Spatial Point Pattern Analysis Code in S-Plus
##
## Version 2 - Spatial and Space-Time analysis

## [1] "Number of basis at resolution 1 = 6"
## [1] "Number of basis at resolution 2 = 27"

## Setup SRE model
f <- log(zinc) ~ 1
S <- SRE(f,list(meuse,HexPols_df2),BAUs = HexPols_df, G,est_error=T)

## [1] "Binned data in 1.286 seconds"

## Warning in map_data_to_BAUs(data[[i]], BAUs, av_var = av_var, variogram.formula
= f, : Not accounting for multiple data in the same grid box during
variogram estimation. Need to see how to do this with gstat

## [1] "sigma2e estimate = 0.0152413306239711"

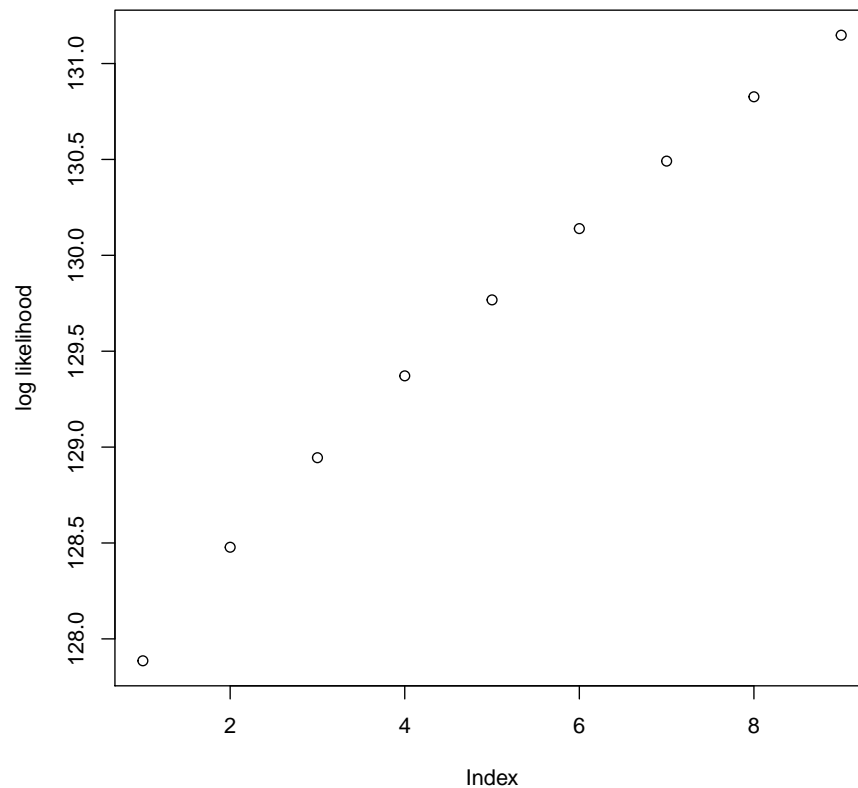
## Warning in map_data_to_BAUs(data[[i]], BAUs, av_var = av_var, variogram.formula
= f, : Not accounting for multiple data in the same grid box during
variogram estimation. Need to see how to do this with gstat

## [1] "sigma2e estimate = 0.00784995366538696"
## [1] "Averaging over polygons"

S <- SRE.fit(S,n_EM = 10,print_lik=T)

## [1] "Maximum EM iterations reached"

```



```
## [1] "Warning: Ignoring constants in log-likelihood computation"

## Point predict
HexPols_df <- SRE.predict(S,use_centroid = T)

## Warning in asMethod(object): as(., "dsCMatrix") is deprecated (since
2008); do use as(., "symmetricMatrix")

X <- SpatialPolygonsDataFrame_to_df(sp_polys = HexPols_df,
                                     vars = c("mu", "var"))

## Joining by: "id"

g1 <- EmptyTheme() +
  geom_polygon(data=X,aes(x,y,fill=mu,group=id),
              colour="light grey") +
  scale_fill_distiller(palette="Spectral",trans="reverse") +
```

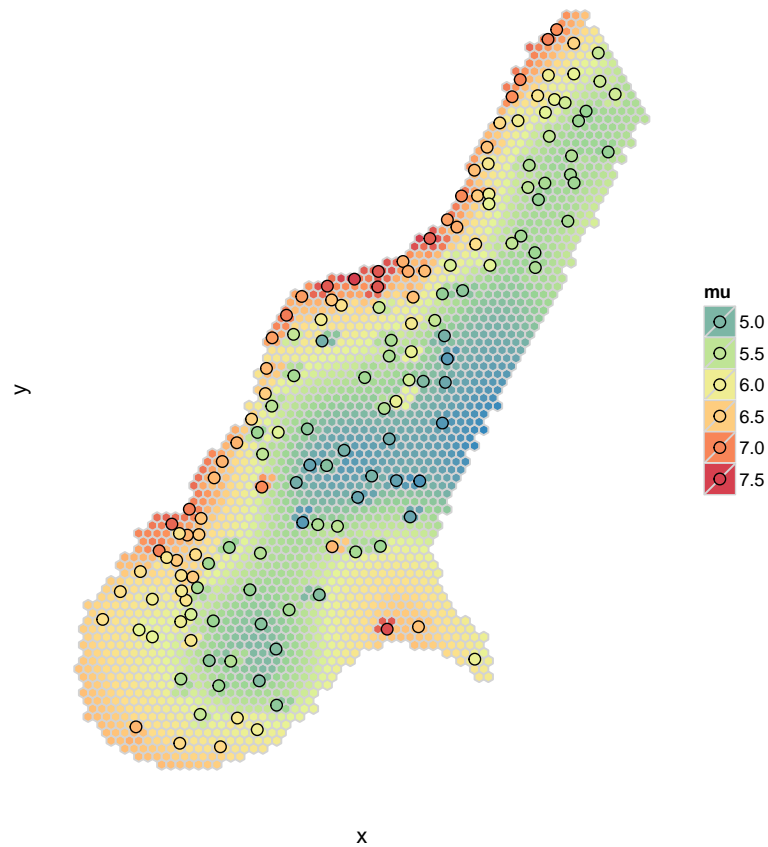
```

geom_point(data=data.frame(meuse),
           aes(x,y,fill=log(zinc)),
           colour="black",
           pch=21, size=3) +
coord_fixed()

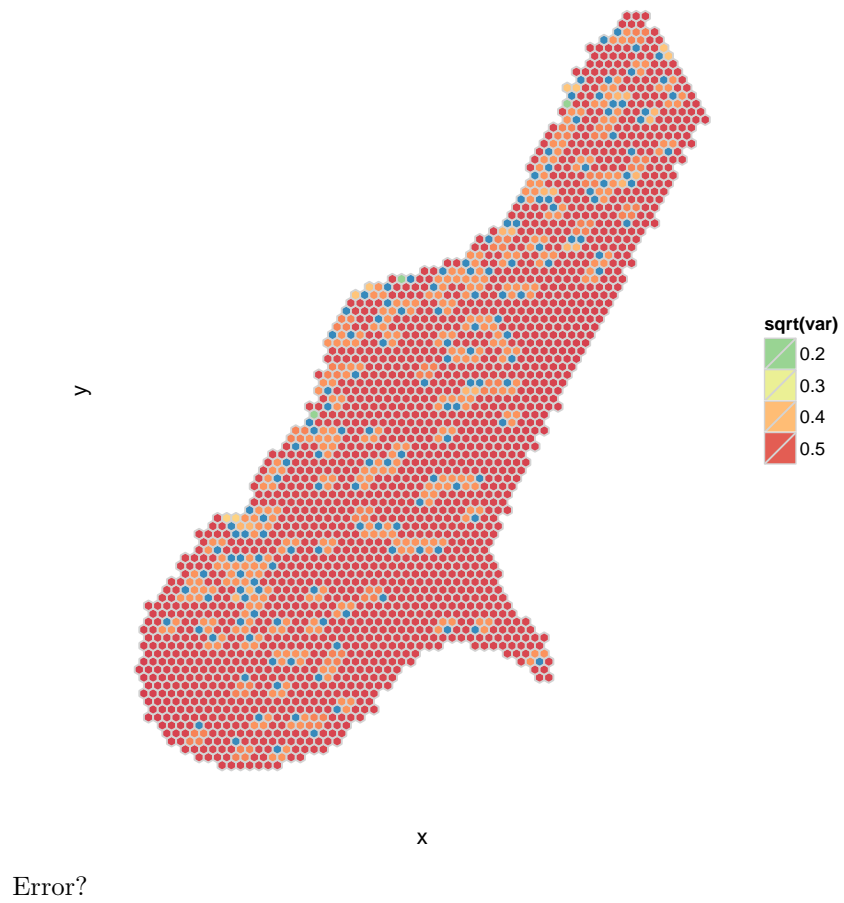
g2 <- EmptyTheme() +
  geom_polygon(data=X,aes(x,y,fill=sqrt(var),group=id),
              colour="light grey") +
  scale_fill_distiller(palette="Spectral",trans="reverse") +
  #geom_point(data=data.frame(meuse),
  #           aes(x,y,colour="black",pch=21, size=3) +
  coord_fixed()

print(g1)

```



```
print(g2)
```



Error?