Mapping spatial data

Zhen Zhang*

July 12, 2022

Abstract

In this document, we will provide an example for mapping spatial data (areal) with CAR model, and some quick estimation.

Contents

Simulated areal data for spatial mapping

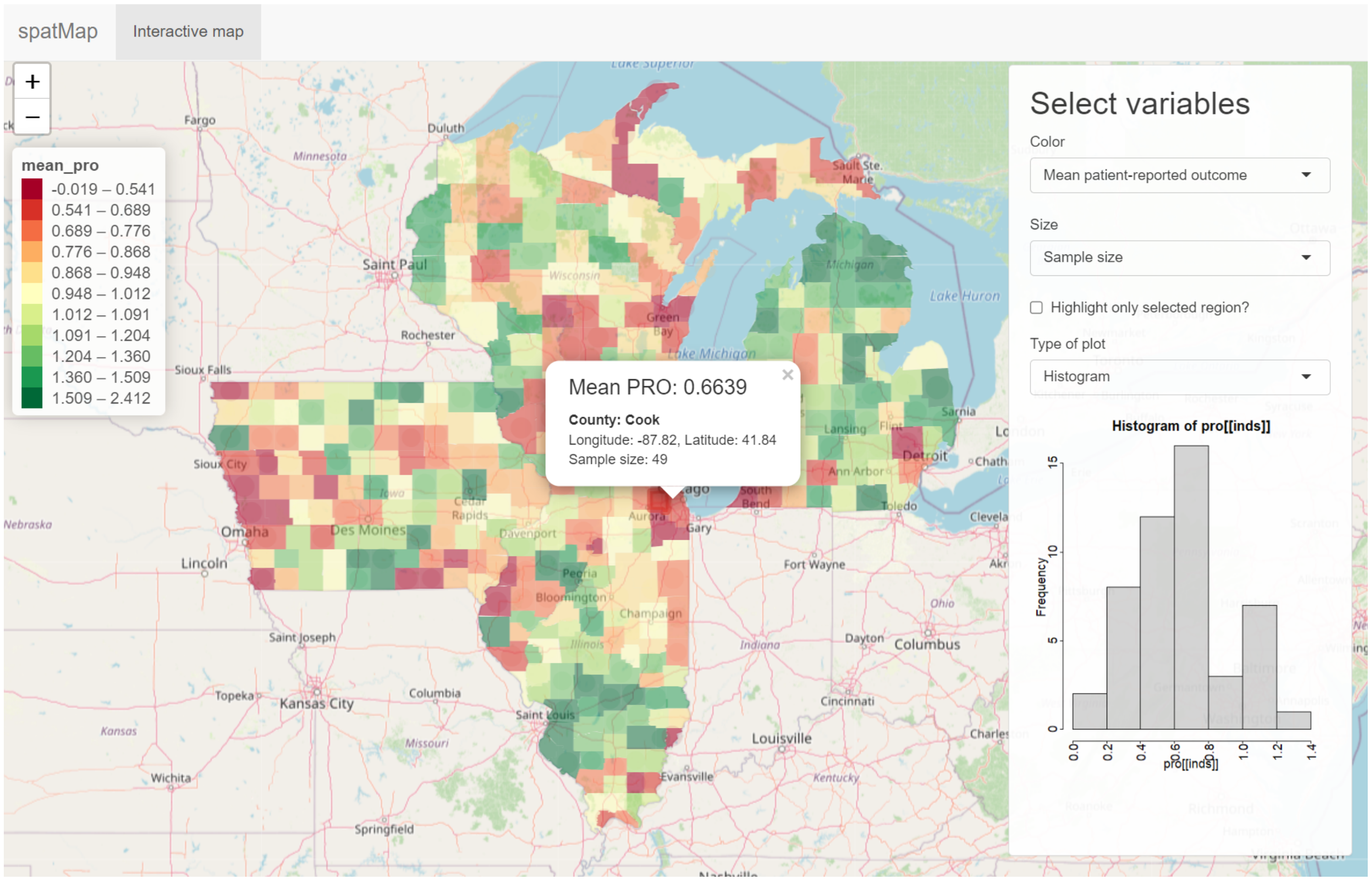
Here is the example R code.

```
rm(list=ls())
require(leaflet)

fileNam <- 'spatInfo.RData'
if(!file.exists(fileNam)){
  #-------------------- prepare the spatial info (e.g., .shp, Adj.csv from ArcGIS)
  dir0 <- './ArcGIS_out/'

  require(rgdal)
  map <- readOGR(dsn=paste0(dir0,'shapefile'),"Counties_for_analysis_region")
  map <- spTransform(map, CRS("+init=epsg:4267"))
  coords <- as.data.frame(coordinates(map)) #centers of the polygons
  names(coords) <- c('Longitude','Latitude')
  county <- map$COUNTY

  require(rgeos)
  map1 <- gSimplify(map, tol=0.01, topologyPreserve=TRUE)

  # need to save Adj csv from txt
  # match Adj.csv with Num.txt
```

*zhangquake1@outlook.com

```r
  W0 <- read.csv(paste0(dir0,'Adjacency/Adj.csv'), head=F)
  M0 <- read.csv(paste0(dir0,'Adjacency/Num.txt'), head=F)
  M <- M0$V1
  n <- nrow(W0); W <- matrix(0,n,n); for(i in 1:n) W[i,na.omit(as.numeric(W0[i,]))] <- 1
  all(W==t(W))
  all(rowSums(W)==M0$V1)
  cols <- rep('lightgreen',n);   k <- 125 #random check site ID
  cols[c(k,na.omit(as.numeric(W0[k,])))] <- 'blue'; cols[k] <- 'red'
  plot(map1, col=cols)  # shape file matches with W

  save(file=fileNam, map1, W, M, coords, county)
}else load(fileNam)

#-------------------- simulate areal data from Conditional AutoRegressive (CAR) model
# with precision matrix = (M-gamma*W)/tau^2
set.seed(1234)
n <- nrow(W)
mu <- 1
tau2 <- 0.5
gamm <- 0.9
Pre <- (diag(M)-gamm*W)/tau2
Z <- matrix(rnorm(n),n,1) #standard Normal
U <- chol(Pre) #so Pre = t(U)%*%U, Sigma=(Pre)^-1 = U^-1%*%t(U)^-1
# hence let Uy=Z, y=U^-1Z ~ N(0, U^-1%*%t(U)^-1 = Sigma) as desired
y0 <- solve(U,Z)  #spatial random effect
y <- mu + y0  #mu = X%*%beta to introduce site-specific covariates
# also for each site i, simulate 20-70 patients with reported outcomes around site mean y
sample_size <- 20 + sample.int(50, size=n, replace=TRUE)
pro <- as.list(rep(NA, n)) #simulate patient-reported outcomes
sig <- 0.3  #nugget effect
for(i in 1:n) pro[[i]] <- rnorm(sample_size[i], y[i], sig)
dat <- cbind(county, coords, sample_size, mean_pro=y)
# save(file='demo.RData', map1, dat, pro)

# customize color at: colorbrewer2.org
J <- length(cols <- c('#a50026','#d73027','#f46d43','#fdae61','#fee08b',
                      '#ffffbf','#d9ef8b','#a6d96a','#66bd63','#1a9850','#006837'))

# graphical control
sizeBy <- 'sample_size'
colorBy <- 'mean_pro'
fac <- 2000 #can be an option from the drag-down menu to control the size of centers
radius <- log(dat[[sizeBy]] / max(dat[[sizeBy]]) * 30000) *fac/2

# categorize the continuous variable
require(arules)
colorData <- dat[[colorBy]]
```

```
cuts <- discretize(colorData, method="frequency", breaks=length(cols), onlycuts=T)
y1 <- discretize(colorData, method='fixed', breaks=cuts)
pal <- colorBin(palette=cols, colorData, bins=cuts, pretty=FALSE)

# interactive spatial mapping
m <- leaflet() %>% addTiles() %>% setView(lng=-88.85, lat=42.45, zoom=5) %>%
  addPolygons(data=map1, stroke=FALSE, fillOpacity=0.5, smoothFactor=0.5, color=cols[y1])
m <- m %>%
  addCircles(data=dat, ~Longitude, ~Latitude, radius=radius, layerId=~county,
             stroke=FALSE, fillOpacity=0.1, fillColor=cols[y1]) %>%
  addLegend("bottomleft", pal=pal, opacity=1, values=colorData, title=colorBy,
            layerId="colorLegend")
# print(m)
knitr::include_graphics("demo.png")
```



Mean PRO: 0.6639

County: Cook

Longitude: -87.82, Latitude: 41.84

Sample size: 49

Select variables

Color: Mean patient-reported outcome

Size: Sample size

☐ Highlight only selected region?

Type of plot: Histogram

Figure 1: Spatial mapping

The results are shown in Figure 1.

Spatial prediction using conditional autoregressive (CAR) model

Estimate the spatial parameters for the areal data:

```r
logLik <- function(para){ #para = c('mu','tau2','gamm') #mean, variance, autocorrelation
  Pre <- (diag(M)-para[3]*W)/para[2]
  U <- chol(Pre) #so Pre = t(U)%*%U, t(y-mu)%*%Pre%*%(y-mu) = ||d||^2 where d=U(y-mu)
  -0.5*sum((U%*%(y-para[1]))^2) + sum(log(diag(U))) #quadratic term + log determinant
}
eps <- .Machine$double.eps
# get a small value for bounding the parameter space to avoid things such as log(0).
paraInit <- c(mu, tau2, gamm)*.8  #perturb the true parameters
# logLik(paraInit)
fit <- optim(paraInit, logLik, method="L-BFGS-B", lower=c(-Inf, eps, -1+eps),
             upper=c(Inf, Inf, .999), hessian=TRUE, control=list(fnscale=-1, trace=TRUE))
```

```
## iter   10 value -224.010423
## final  value -224.010423
## converged
```

```r
paraMLE <- fit$par  #MLE of the CAR model: maximizing the log likelihood
SE <- sqrt(diag(solve(-fit$hessian)))
est <- data.frame(est=paraMLE, SE=SE, lower=paraMLE-SE*qnorm(.975),
                  upper=paraMLE+SE*qnorm(.975), pval=2*(1-pnorm(abs(paraMLE)/SE)))
row.names(est) <- c('mu','tau2','gamm')
knitr::kable(est, row.names=TRUE, caption = "\\label{tab:t1}Parameter estimate.", digits=3)
```

Table 1: Parameter estimate.

	est	SE	lower	upper	pval
mu	1.005	0.060	0.888	1.121	0
tau2	0.480	0.038	0.407	0.554	0
gamm	0.932	0.034	0.865	0.998	0

The results are shown in Table 1.