R code for the statistical analysis of the paper: 'Taking advantage of sampling designs in Bayesian spatial small area survey studies'

Preparations

Libraries loading

```
if (!"pbugs" %in% installed.packages() | packageVersion("pbugs") < "0.1.4") {
  remotes::install_github(repo = "fisabio/pbugs")
}
library(pbugs)

library(spdep)
```

```
## Loading required package: sp

## Loading required package: spData

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`

## Loading required package: sf

## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1
```

```
library(RColorBrewer)
```

Data loading

```
load("data/survey.Rdata")
head(survey,n = 3)
```

```
##   neigh NHIC   age ownhouse
## 1    60    1 20-39       No
## 2    31    0 20-39       No
## 3    18    0 20-39       No
```

```
# data.frame not supplied due to confidenciality restrictions

# data corresponding to the Barcelona health survey questionnaires

# Columns:

# neigh: neighborhood

# NHIC: Non-high income origin country

# age: age group
```

```
# ownhouse: owner of his/her living house

load("data/neighs.Rdata")
head(neighs,n = 3)
```

```
##   gold.standard   n     N    p.0.19   p.20.39   p.40.59   p.60.79    p.80...
## 1     0.4586879 194 47359 0.1679301 0.3849743 0.2876116 0.1160920 0.04339196
## 2     0.2637934  46 15569 0.1109256 0.4253324 0.2786306 0.1318646 0.05324684
## 3     0.2023961  72 14941 0.1169266 0.3748745 0.2738103 0.1591594 0.07522923
##    p.NotOwn     p.Own
## 1 0.8043601 0.1956399
## 2 0.8180046 0.1819954
## 3 0.7428956 0.2571044
```

```
# (exhaustive) data on Barcelona neighborhoods corresponding to the city census

# Columns:

# gold.standard: percentage of people from non-high income origin country

# n: number of people in the neighborhood interviewed in the Barcelona health survey

# N: number of people in the neighborhood

# p.XX.YY: percentage of people in the neighborhood with age from XX to YY

# p.(Not)Own: Percentage of people in teh neighborhood owning their current home

load("data/geo.Rdata")
# Cartography (carto) and adjacencies list (adjacencies)
```

Simple Random Sampling (Case study 1)

Frequentist SRS estimate

```
# frequentist SRS estimate
pi.SRS<-tapply(survey$NHIC,survey$neigh,mean)

# Correlation with the gold standard
cor.SRS<-cor(pi.SRS,neighs$gold.standard)
cor.SRS
```

```
## [1] 0.6123257
```

```
# SQRT.MSE
MSE.SRS<-sqrt(mean((pi.SRS-neighs$gold.standard)^2))
MSE.SRS
```

```
## [1] 0.08322505
```

```
# CIs
sd.pi.SRS<-sqrt(pi.SRS*(1-pi.SRS)/neighs$n*(1-neighs$n/neighs$N))
CIl.SRS<-pi.SRS-qt(0.975,neighs$n)*sd.pi.SRS
CIu.SRS<-pi.SRS+qt(0.975,neighs$n)*sd.pi.SRS
# mean CI length
```

```
L.interval.SRS<-mean(2*qt(0.975,neighs$n)*sd.pi.SRS)
L.interval.SRS
```

```
## [1] 0.2707645
```

```
# Empirical coverage of the CIs
Cover.SRS<-mean(CIl.SRS<neighs$gold.standard & neighs$gold.standard<CIu.SRS)
Cover.SRS
```

```
## [1] 0.8082192
```

```
#  Moran's I
nbNeigh<-nb2listw(poly2nb(carto),style="W")
# Moran.s I for the gold.standard
I.true<-moran(neighs$gold.standard,nbNeigh,n=length(carto),Szero(nbNeigh))
# Moran's I for the SRS estimator
I.SRS<-moran(pi.SRS,nbNeigh,n=length(carto),Szero(nbNeigh))
c(I.true$I,I.SRS$I)
```

```
## [1] 0.2021324 0.1294423
```

```
# The gold standard shows larger spatial dependence than SRS

# Moran's I distribution under SRS
I.SRS.sample<-vector()
set.seed(1)
for(i in 1:1000){
I.SRS.sample[i]<-moran(pi.SRS+rt(73,df=neighs$n)*sd.pi.SRS,nbNeigh,n=length(carto),Szero(nbNeigh))$I
}
# Probability of Moran's I distribution under SRS to be lower than that of the gold standard
mean(I.SRS.sample<I.true)
```

```
## [1] 0.986
```

Bayesian SRS estimate

```
obs<-tapply(survey$NHIC,survey$neigh,sum)
# Pis' posterior distribution: dbeta(obs+1,n-obs+1)
# Posterior Mode: (alpha-1)/(alpha+beta-2)=obs/n coincides with pi.SRS.

# CIs
CIl.SRSbayes<-qbeta(0.025,obs+1,neighs$n-obs+1)
CIu.SRSbayes<-qbeta(0.975,obs+1,neighs$n-obs+1)
L.interval.SRSbayes<-mean(CIu.SRSbayes-CIl.SRSbayes)
c(L.interval.SRS,L.interval.SRSbayes)
```

```
## [1] 0.2707645 0.2516170
```

```
Cover.SRSbayes<-mean(CIl.SRSbayes<neighs$gold.standard & neighs$gold.standard<CIu.SRSbayes)
c(Cover.SRS,Cover.SRSbayes)
```

```
## [1] 0.8082192 0.9589041
```

```
# Avoiding the normality assumption improves the empirical coverage problems of the previous CIs

# Moran's I
I.SRSbayes<-I.SRS
# Moran's I for the bayes estimator coincides with the SRS since their estimates also coincide
```

```r
I.SRSbayes.sample<-vector()
set.seed(1)
for(i in 1:1000){
I.SRSbayes.sample[i]<-moran(rbeta(73,obs+1,neighs$n-obs+1)
      ,nbNeigh,n=length(carto),Szero(nbNeigh))$I
}
mean(I.SRSbayes.sample<I.true)
```

```
## [1] 0.969
```

```r
# The lack of spatial dependence of pi.SRS is not solved by the bayes estimates.
```

Spatially dependent Bayesian estimates

```r
# WinBUGS Model
modSRSsp <- function() {
  for (i in 1:nIndiv) {
    Y[i] ~ dbern(p[i])
    logit(p[i]) <- mu+sd.theta*theta[neigh[i]]
  }

  mu~dflat()
  sd.theta~dunif(0,10)

  # lCAR distribution for theta
  for (j in 1:nNeigh) {
    theta[j] ~ dnorm(media.theta[j], prec.theta[j])
    prec.theta[j]  <- (1 - lambda + lambda * nadj[j])
    media.theta[j] <- (lambda / (1 - lambda + lambda * nadj[j])) *
    sum(theta.map[(index[j] + 1):index[j + 1]])

    logit(pi[j])<-mu+sd.theta*theta[j]
  }
  for (k in 1:nadj.tot) {
    theta.map[k] <- theta[map[k]]
  }
  lambda~dunif(0,1)

}

# Data
dataSRSsp <- list(
  Y = survey$NHIC,
  neigh = survey$neigh,
  nNeigh = nrow(neighs),
  nIndiv = sum(neighs$n),
  nadj.tot = length(adjacencies$adj),
  nadj = adjacencies$num,
  map = adjacencies$adj,
  index = c(0, cumsum(adjacencies$num))
)
# Parameters to save
parametersSRSsp <- c( "mu", "lambda", "sd.theta", "pi")
```

```r
# Inits
initsSRSsp  <- function() {
  list(
    theta= rnorm(dataSRSsp$nNeigh, 0,1), mu=rnorm(1),sd.theta=runif(1)
  )
}

set.seed(1)
resultSRSsp <- pbugs(
  data               = dataSRSsp,
  inits              = initsSRSsp,
  parameters.to.save = parametersSRSsp,
  model.file         = modSRSsp,
  n.chains           = 3,
  n.iter             = 6000,
  n.burnin           = 1000,
  n.thin = 1,
  bugs.seed = 1
)
```

```r
# \hat{pi}^{spatial} estimates
Y.miss<-matrix(nrow=dim(resultSRSsp$sims.list$pi)[1],ncol=dim(resultSRSsp$sims.list$pi)[2])
for(i in 1:(dim(resultSRSsp$sims.list$p)[1])){
  Y.miss[i,]<-rbinom(dim(neighs)[1],neighs$N-neighs$n,resultSRSsp$sims.list$pi[i,])
}
Y.sample<-tapply(survey$NHIC,survey$neigh,sum)
p.spatial<-matrix(nrow=dim(resultSRSsp$sims.list$pi)[1],ncol=dim(resultSRSsp$sims.list$pi)[2])
for(i in 1:(dim(resultSRSsp$sims.list$p)[1])){
  p.spatial[i,]<-(neighs$n/neighs$N)*(Y.sample/neighs$n)+((neighs$N-neighs$n)/neighs$N)*Y.miss[i,]/(neig
}

# Correlation with the gold standard
cor(neighs$gold.standard,apply(p.spatial,2,mean))
```

```
## [1] 0.7468466
```

```r
cor(neighs$gold.standard,obs/neighs$n)
```

```
## [1] 0.6123257
```

```r
# SQRT.MSE
sqrt(mean((neighs$gold.standard-apply(p.spatial,2,mean))^2))
```

```
## [1] 0.04904433
```

```r
sqrt(mean((neighs$gold.standard-(obs/neighs$n))^2))
```

```
## [1] 0.08322505
```

```r
# CIs
CIl.SRSsp<-apply(p.spatial,2,quantile,0.025)
CIu.SRSsp<-apply(p.spatial,2,quantile,0.975)
L.interval.SRSsp<-mean(CIu.SRSsp-CIl.SRSsp)
c(L.interval.SRSbayes,L.interval.SRSsp)
```

```
## [1] 0.2516170 0.1721787
```

```
Cover.SRSsp<-mean(CIl.SRSsp<neighs$gold.standard & neighs$gold.standard<CIu.SRSsp)
c(Cover.SRS,Cover.SRSbayes,Cover.SRSsp)
```

```
## [1] 0.8082192 0.9589041 0.9452055
```

```
# Moran's I
I.SRSsp.sample<-apply(p.spatial,1,function(x){moran(x,nbNeigh,n=length(carto),Szero(nbNeigh))$I})
mean(I.SRSsp.sample)
```

```
## [1] 0.2328906
```

```
mean(I.SRSsp.sample<I.true)
```

```
## [1] 0.6742667
```

```
# The spatial dependence of this estimator matches that observed at population level.
```

```
# \hat{pi}^star estimates
# Correlation with the gold standard
cor(neighs$gold.standard,resultSRSsp$mean$pi)
```

```
## [1] 0.7469775
```

```
cor(neighs$gold.standard,obs/neighs$n)
```

```
## [1] 0.6123257
```

```
# SQRT.MSE
sqrt(mean((neighs$gold.standard-resultSRSsp$mean$pi)^2))
```

```
## [1] 0.04901569
```

```
sqrt(mean((neighs$gold.standard-(obs/neighs$n))^2))
```

```
## [1] 0.08322505
```

```
# CIs
CIl.SRSsp<-resultSRSsp$summary[grep("pi",rownames(resultSRSsp$summary)),3]
CIu.SRSsp<-resultSRSsp$summary[grep("pi",rownames(resultSRSsp$summary)),7]
L.interval.SRSsp<-mean(CIu.SRSsp-CIl.SRSsp)
c(L.interval.SRSbayes,L.interval.SRSsp)
```

```
## [1] 0.2516170 0.1719827
```

```
Cover.SRSsp<-mean(CIl.SRSsp<neighs$gold.standard & neighs$gold.standard<CIu.SRSsp)
c(Cover.SRS,Cover.SRSbayes,Cover.SRSsp)
```

```
## [1] 0.8082192 0.9589041 0.9452055
```

```
# Moran's I
I.SRSsp.sample<-apply(resultSRSsp$sims.list$pi,1,function(x){moran(x,nbNeigh,n=length(carto),Szero(nbNei
mean(I.SRSsp.sample)
```

```
## [1] 0.2333322
```

```
mean(I.SRSsp.sample<I.true)
```

```
## [1] 0.6712
```

```
# The spatial dependence of this estimator matches that observed at population level.
```

Auxiliary variables (Case study 2)

Stratified sampling estimator (Age)

```
# Age-specific probabilities of foreign population
pi.age<-cbind(
tapply(survey$NHIC[survey$age=="0-19"],factor(survey$neigh[survey$age=="0-19"],levels = 1:73),mean),
tapply(survey$NHIC[survey$age=="20-39"],factor(survey$neigh[survey$age=="20-39"],levels = 1:73),mean),
tapply(survey$NHIC[survey$age=="40-59"],factor(survey$neigh[survey$age=="40-59"],levels = 1:73),mean),
tapply(survey$NHIC[survey$age=="60-79"],factor(survey$neigh[survey$age=="60-79"],levels = 1:73),mean),
tapply(survey$NHIC[survey$age=="80-..."],factor(survey$neigh[survey$age=="80-..."],levels = 1:73),mean)
c(mean(is.na(pi.age)),mean(apply(is.na(pi.age),1,any)))
```

```
## [1] 0.06575342 0.17808219
```

```
# 6.6% of the age-neighborhood combinations are not represented in the sample, what will make a 17.8% o

# Proportion of age-groups-neighborhoods combinations with sample sizes between 1 and 5
(sum(0<table(factor(survey$neigh[survey$age=="0-19"],levels = 1:73)) & table(factor(survey$neigh[survey
sum(0<table(factor(survey$neigh[survey$age=="20-39"],levels = 1:73)) & table(factor(survey$neigh[survey
sum(0<table(factor(survey$neigh[survey$age=="40-59"],levels = 1:73)) & table(factor(survey$neigh[survey
sum(0<table(factor(survey$neigh[survey$age=="60-79"],levels = 1:73)) & table(factor(survey$neigh[survey
sum(0<table(factor(survey$neigh[survey$age=="80-..."],levels = 1:73)) & table(factor(survey$neigh[surve
```

```
## [1] 0.2849315
```

```
# stratified estimator
pop.age<-neighs[,c("p.0.19","p.20.39","p.40.59","p.60.79","p.80...")]
pi.str<-rowSums(pi.age*pop.age)
mean(is.na(pi.str))
```

```
## [1] 0.1780822
```

```
# Correlation with the gold.standard
cor.str<-cor(pi.str[!is.na(pi.str)],neighs$gold.standard[!is.na(pi.str)])
cor.str
```

```
## [1] 0.7237157
```

```
cor(pi.SRS[!is.na(pi.str)],neighs$gold.standard[!is.na(pi.str)])
```

```
## [1] 0.7405775
```

```
# The improvement of the stratified estimator over the SRS estimator in terms of correlation is mild

# SQRT.MSE
MSE.str<-sqrt(mean((pi.str[!is.na(pi.str)]-neighs$gold.standard[!is.na(pi.str)])^2))
MSE.str
```

```
## [1] 0.06156299
```

```
sqrt(mean(((obs/neighs$n)[!is.na(pi.str)]-neighs$gold.standard[!is.na(pi.str)])^2))
```

```
## [1] 0.06085416
```

```
# The improvement in terms of the MSE is also mild

# CIs
######
sd.pi.str<-sqrt(rowSums(pop.age*pop.age*pi.str*(1-pi.str)/(neighs$n-1)*(neighs$N-neighs$n)/neighs$N))
```

```r
L.interval.str<-mean(2*qt(0.975,neighs$n)*sd.pi.str,na.rm=T)
c(mean(CIu.SRS[!is.na(pi.str)]-CIl.SRS[!is.na(pi.str)]),L.interval.str)
```

```
## [1] 0.2055585 0.1009214
```

```r
# The stratified CIs are substantially narrower than the SRSs

Cover.str<-mean((pi.str-qt(0.975,neighs$n)*sd.pi.str)<neighs$gold.standard & neighs$gold.standard<(pi.s
Cover.str
```

```
## [1] 0.6
```

```r
# ICs Coverage for SRS wherever pi.str is not missing
mean(CIl.SRS[!is.na(pi.str)]<neighs$gold.standard[!is.na(pi.str)] & neighs$gold.standard[!is.na(pi.str)]
```

```
## [1] 0.8666667
```

```r
# The coverage of the ICs is also far lower than for SRS

# Moran's I
###########
I.str<-moran(pi.str,nbNeigh,n=length(carto),Szero(nbNeigh),NAOK = TRUE)$I
I.str
```

```
## [1] 0.04356376
```

```r
aux<-neighs$gold.standard
aux[is.na(pi.str)]<-NA
I.true.NA<-moran(aux,nbNeigh,n=length(carto),Szero(nbNeigh),NAOK = TRUE)$I
I.true.NA
```

```
## [1] 0.1409699
```

```r
# Moran's I for the stratified estimator is lower than for the gold standard

# Randomization
I.str.sample<-vector()
for(i in 1:1000){I.str.sample[i]<-moran(rnorm(length(carto),pi.str,sd.pi.str),nbNeigh,n=length(carto),S
mean(I.str.sample<I.true.NA)
```

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

```r
# Moran's I of the stratified estimator does not fit that of the gold.standard

# Moran's I for SRS excluding those outcomes where pi.str is missing
aux<-pi.SRS
aux[is.na(pi.str)]<-NA
I.SRS.NA<-moran(aux,nbNeigh,n=length(carto),Szero(nbNeigh),NAOK = TRUE)$I
I.SRS.NA
```

```
## [1] 0.04798209
```

```r
I.SRS.NA.sample<-vector()
for(i in 1:1000){I.SRS.NA.sample[i]<-moran(rnorm(length(carto),aux,sd.pi.SRS),nbNeigh,n=length(carto),S
mean(I.SRS.NA.sample<I.true.NA)
```

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

Ratio estimator (House owning)

```
pi.own<-cbind(
tapply(survey$NHIC[survey$ownhouse=="No"],factor(survey$neigh[survey$ownhouse=="No"],levels = 1:73),mean
tapply(survey$NHIC[survey$ownhouse=="Yes"],factor(survey$neigh[survey$ownhouse=="Yes"],levels = 1:73),me
c(mean(is.na(pi.own)),mean(apply(is.na(pi.own),1,any)))
```

```
## [1] 0.02739726 0.05479452
```

```
# 2.7% of the owning-neighborhood combinations are not represented in the sample, what will make a 5.5%

# Proportion of owning-groups-neighborhoods combinations with sample sizes between 1 and 5
(sum(0<table(factor(survey$neigh[survey$ownhouse=="No"],levels = 1:73)) & table(factor(survey$neigh[surv
sum(0<table(factor(survey$neigh[survey$ownhouse=="Yes"],levels = 1:73)) & table(factor(survey$neigh[surv
```

```
## [1] 0.0890411
```

```
# Ratio estimator including house owning information
pi.ratio<-rowSums(pi.own*cbind(neighs$p.NotOwn,neighs$p.Own))

# Correlation with the gold standard
cor.ratio<-cor(pi.ratio[!is.na(pi.ratio)],neighs$gold.standard[!is.na(pi.ratio)])
cor.ratio
```

```
## [1] 0.7272673
```

```
cor(pi.SRS[!is.na(pi.ratio)],neighs$gold.standard[!is.na(pi.ratio)])
```

```
## [1] 0.6990941
```

```
# The ratio estimator shows an evident improvement over the SRS estimator

# SQRT.MSE
sqrt(mean((pi.ratio[!is.na(pi.ratio)]-neighs$gold.standard[!is.na(pi.ratio)])^2))
```

```
## [1] 0.06227874
```

```
sqrt(mean((pi.SRS[!is.na(pi.ratio)]-neighs$gold.standard[!is.na(pi.ratio)])^2))
```

```
## [1] 0.06888978
```

```
# The improvement is also evident in terms of MSE

# CIs
sd.pi.ratio<-sqrt(rowSums(pop.age*pop.age*pi.ratio*(1-pi.ratio)/(neighs$n-1)*(neighs$N-neighs$n)/neighs$
L.interval.ratio<-mean(2*qt(0.975,neighs$n)*sd.pi.ratio,na.rm=T)
c(mean(CIu.SRS[!is.na(pi.ratio)]-CIl.SRS[!is.na(pi.ratio)]),L.interval.ratio)
```

```
## [1] 0.2316223 0.1151147
```

```
# The length of the ratio estimator is substantially lower for the ratio estimator

Cover.ratio<-mean((pi.ratio-qt(0.975,neighs$n)*sd.pi.ratio)<neighs$gold.standard & neighs$gold.standard
Cover.ratio
```

```
## [1] 0.6666667
```

```
mean(CIl.SRS[!is.na(pi.ratio)]<neighs$gold.standard[!is.na(pi.ratio)] & neighs$gold.standard[!is.na(pi.
```

```
## [1] 0.826087
```

```
# Anyway, the coverage of the CIs is lower than for SRSs, and lower than expected

# Moran,s I
I.ratio<-moran(pi.ratio,nbNeigh,n=length(carto),Szero(nbNeigh),NAOK = TRUE)$I
I.ratio
```

```
## [1] 0.1497097
```

```
aux<-neighs$gold.standard
aux[is.na(pi.ratio)]<-NA
I.true.NA<-moran(aux,nbNeigh,n=length(carto),Szero(nbNeigh),NAOK = TRUE)$I
c(I.ratio,I.true.NA)
```

```
## [1] 0.1497097 0.1788096
```

```
I.ratio.sample<-vector()
set.seed(1)
for(i in 1:1000){I.ratio.sample[i]<-moran(rnorm(length(carto),pi.ratio,sd.pi.ratio),nbNeigh,n=length(ca
mean(I.ratio.sample<I.true.NA)
```

```
## [1] 0.806
```

```
# I.SRS considering as missing values those values with missing pi.str
aux<-pi.SRS
aux[is.na(pi.ratio)]<-NA
I.ratio.NA<-moran(aux,nbNeigh,n=length(carto),Szero(nbNeigh),NAOK = TRUE)$I
I.ratio.NA
```

```
## [1] 0.1583015
```

```
I.ratio.NA.sample<-vector()
set.seed(1)
for(i in 1:1000){I.ratio.NA.sample[i]<-moran(rnorm(length(carto),aux,sd.pi.SRS),nbNeigh,n=length(carto)
mean(I.ratio.NA.sample<I.true.NA)
```

```
## [1] 0.838
```

Full (stratified-ratio-spatial) model

```
# Model
modFull <- function() {
  for (i in 1:nIndiv) {
    Y[i] ~ dbern(p[i])
    logit(p[i]) <- mu +
      beta.age[age[i]] +
      beta.own[own[i]] +
      sd.theta*theta[neigh[i]]
  }

  mu ~ dflat()
  sd.theta ~ dunif(0,10)

  beta.age[3] <- 0
  for (k in 1:2) {
    beta.age[k] ~ dflat()
  }
  for (k in 4:nAge) {
```

```
    beta.age[k] ~ dflat()
  }
  beta.own[1] <- 0
  beta.own[2] ~ dflat()

  lambda ~ dunif(0, 1)

  # lCAR distribution for theta
  for (k in 1:nNeigh) {
    theta[k]        ~ dnorm(media.theta[k], prec.theta[k])
    prec.theta[k]  <- (1 - lambda + lambda * nadj[k])
    media.theta[k] <- (lambda / (1 - lambda + lambda * nadj[k])) *
    sum(theta.map[(index[k] + 1):index[k + 1]])
  }
  for (k in 1:nadj.tot) {
    theta.map[k] <- theta[map[k]]
  }

  for(iNeigh in 1:nNeigh){
    for(iAge in 1:nAge){
      for(iOwn in 1:nOwn){
          logit(pi.pred[iAge,iOwn,iNeigh])<- mu +
              beta.age[iAge] +
              beta.own[iOwn] +
              sd.theta*theta[iNeigh]
      }
      pi.pred2[iAge,iNeigh]<-inprod2(pi.pred[iAge,,iNeigh],p.own[iNeigh,])
    }
    pi[iNeigh]<-inprod2(pi.pred2[,iNeigh],p.age[iNeigh,])
  }
}

# Data
dataFull <- list(
  Y = survey$NHIC,
  age = as.numeric(survey$age),
  own = as.numeric(survey$ownhouse),
  nOwn = length(unique(survey$ownhouse)),
  nAge = length(unique(survey$age)),
  neigh = survey$neigh,
  nNeigh = nrow(neighs),
  nadj.tot = length(adjacencies$adj),
  nadj = adjacencies$num,
  map = adjacencies$adj,
  index = c(0, cumsum(adjacencies$num)),
  nIndiv = sum(neighs$n),
  p.age = as.matrix(pop.age),
  p.own = cbind(neighs$p.NotOwn,neighs$p.Own)
)

# Inits and parameters
parametersFull <- c(
  "mu", "lambda", "sd.theta", "beta.age", "beta.own", "theta", "pi"
```

```
)

initsFull  <- function() {
  list(
    mu                = rnorm(1, 0, 0.1),
    beta.age         = c(rnorm(2), NA, rnorm(dataFull$nAge - 3)),
    beta.own     = c(NA, rnorm(1)),
    lambda            = runif(1),
    sd.theta          = runif(1),
    theta             = rnorm(dataFull$nNeigh, 0.1)
  )
}

# Model run
set.seed(1)
resultFull <- pbugs(
  data               = dataFull,
  inits              = initsFull,
  parameters.to.save = parametersFull,
  model.file         = modFull,
  n.chains           = 3,
  n.iter             = 6000,
  n.burnin           = 1000,
  DIC =FALSE,
  bugs.seed = 1
)
```

```
# Correlation
cor(neighs$gold.standard,resultFull$mean$pi)
```

```
## [1] 0.7962962
```

```
cor(neighs$gold.standard,resultSRSsp$mean$pi)
```

```
## [1] 0.7469775
```

```
cor(obs/neighs$n,neighs$gold.standard)
```

```
## [1] 0.6123257
```

```
# SQRT.MSE
sqrt(mean((neighs$gold.standard-resultFull$mean$pi)^2))
```

```
## [1] 0.04336361
```

```
sqrt(mean((neighs$gold.standard-resultSRSsp$mean$pi)^2))
```

```
## [1] 0.04901569
```

```
sqrt(mean((neighs$gold.standard-(obs/neighs$n))^2))
```

```
## [1] 0.08322505
```

```
# CIs
CIl.Full<-resultFull$summary[grep("pi",rownames(resultFull$summary)),3]
CIu.Full<-resultFull$summary[grep("pi",rownames(resultFull$summary)),7]
L.interval.Full<-mean(CIu.Full-CIl.Full)
c(L.interval.SRSbayes,L.interval.SRSsp,L.interval.Full)
```

```
## [1] 0.2516170 0.1719827 0.1412398
```

```
Cover.Full<-mean(CIl.Full<neighs$gold.standard & neighs$gold.standard<CIu.Full)
c(Cover.SRS,Cover.SRSbayes,Cover.SRSsp,Cover.Full)
```

```
## [1] 0.8082192 0.9589041 0.9452055 0.9589041
```

```
# Moran's I
I.full<-moran(resultFull$mean$pi,nbNeigh,n=length(carto),Szero(nbNeigh))$I
I.full
```

```
## [1] 0.3050834
```

```
I.Full.sample<-apply(resultFull$sims.list$pi,1,function(x){moran(x,nbNeigh,n=length(carto),Szero(nbNeig
mean(I.Full.sample<I.true)
```

```
## [1] 0.6437126
```

```
# The spatial dependence observed in the gold standard matches that observed in the full estimator.
```

Figure 1

```
dev.new()

par(mfrow=c(2,2), mar=c(0,0,2,0))
pi.SRS.cut<-cut(pi.SRS,c(-0.1,0.05,0.10,0.15,0.20,0.25,1.1))
cols.SRS<-brewer.pal(name = "Blues",n=6)[pi.SRS.cut]
plot(carto,col=cols.SRS)
legend(x="bottomright",legend=c("<5%","5%-9%","10%-14%","15%-19%","20%-24%",">=25%"),fill=brewer.pal(na
title("SRS")

pi.spatial.cut<-cut(resultSRSsp$mean$pi,c(-0.1,0.05,0.10,0.15,0.20,0.25,1.1))
cols.spatial<-brewer.pal(name = "Blues",n=6)[pi.spatial.cut]
plot(carto,col=cols.spatial)
title("Spatial SRS")

pi.Full.cut<-cut(resultFull$mean$pi,c(-0.1,0.05,0.10,0.15,0.20,0.25,1.1))
cols.Full<-brewer.pal(name = "Blues",n=6)[pi.Full.cut]
plot(carto,col=cols.Full)
title("Full estimator")

pi.gold<-cut(neighs$gold.standard,c(-0.1,0.05,0.10,0.15,0.20,0.25,1.1))
cols.gold<-brewer.pal(name = "Blues",n=6)[pi.gold]
plot(carto,col=cols.gold)
title("Gold standard")
```