

Spatial Epi HW 3

David Coomes

Due February 24, 2020

Question 1

- (a) The posterior median of β is -0.0292 (95% CI: -0.0784-0.0138). The total variance of the random effects is 0.128 (95% CI: 0.082-0.128) and the proportion of the total variance that we can attribute to the spatial random effects is 0.0747 (95% CI: 0.0045-0.4967).
- (b) There is a relatively large difference between the RR estimates from the spatially smoothed model as compared to the direct estimates, particularly for those estimates that had large and small SMR estimates. This makes us believe that the SMRs at the highest and lowest ends were driven by fairly large uncertainty. The non-spatially smoothed estimates and the spatially smoothed estimates were quite similar. This is unsurprising as only about 7.5% of the random effects were attributed to spatial variation.

Figure 1: Map of relative risk estimates for cancer in Ohio using an ICAR spatially smoothed model.

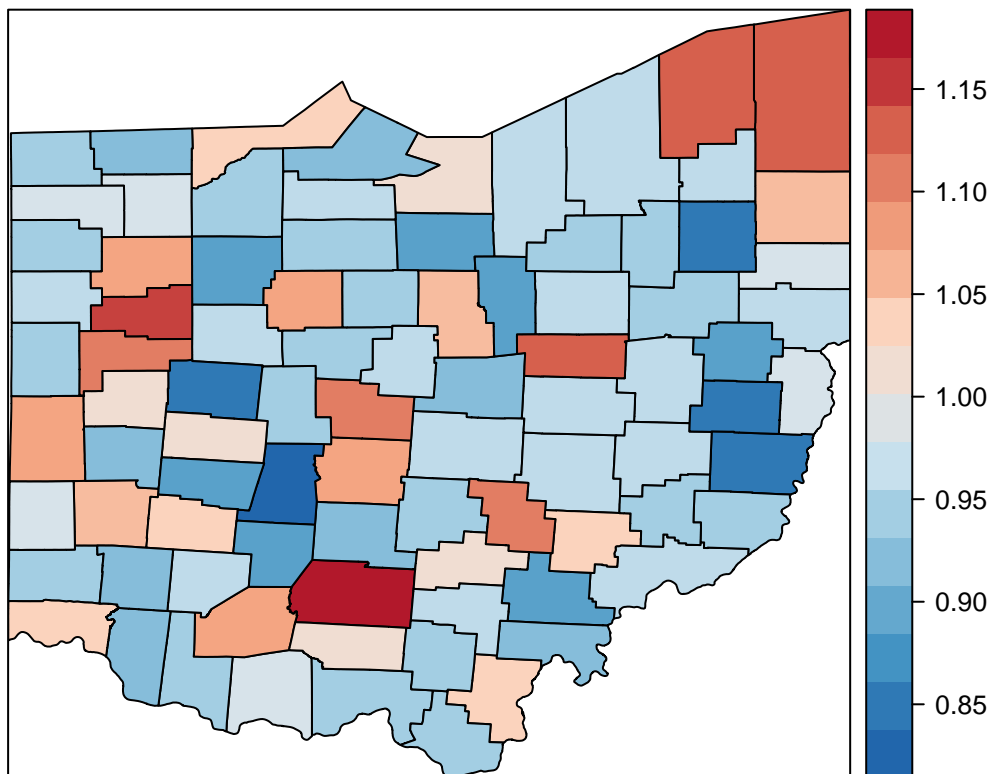


Figure 2: Plot of direct estimates (SMR) and spatially fitted estimates

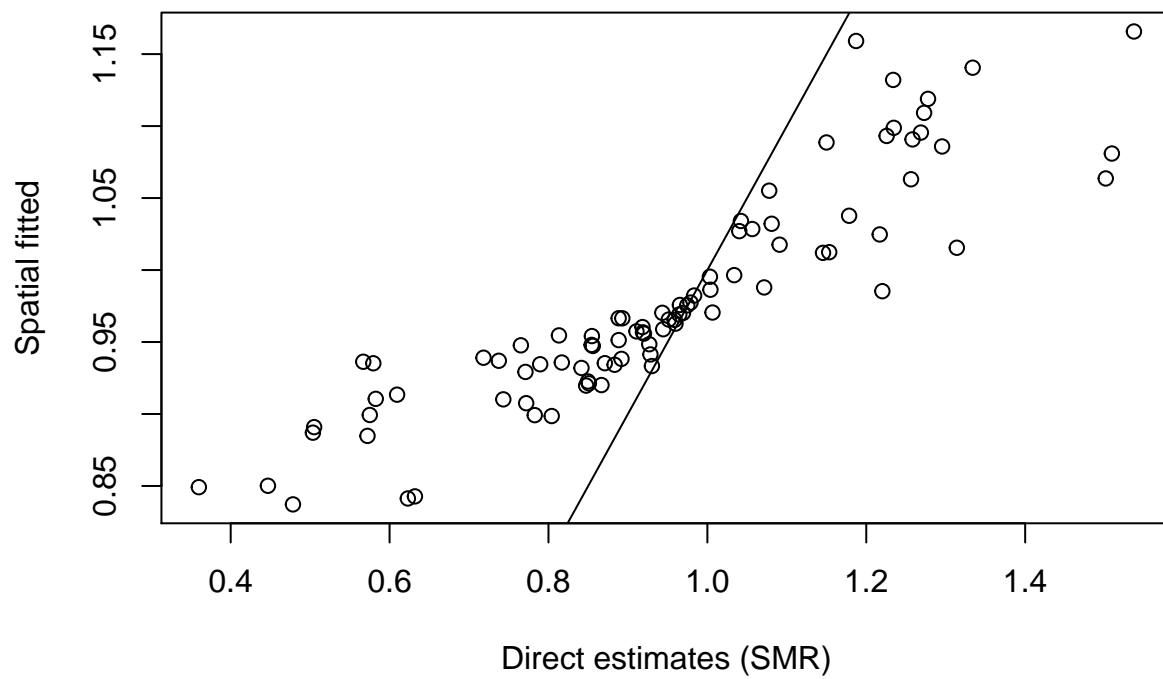
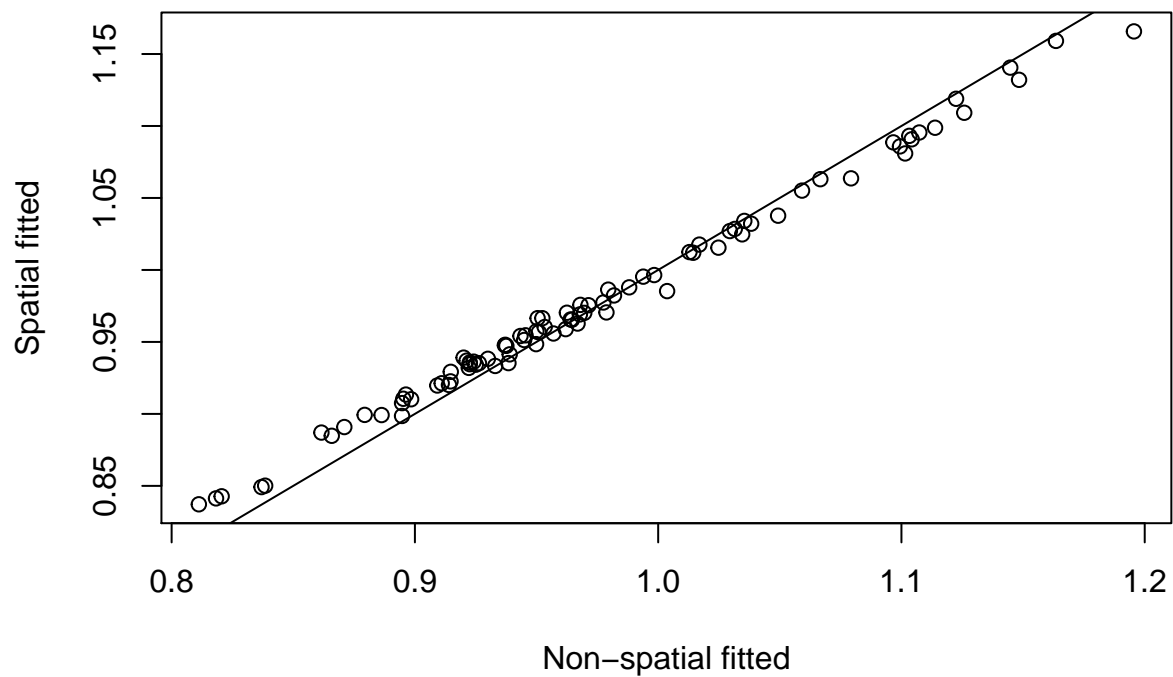


Figure 3: Plot of non-spatially fitted estimates and spatially fitted estimates



Question 2

- (b) In the two panels comparing smoothed and unsmoothed estimates (1 & 3), the smoothed estimates are shifted slightly towards the mean, particularly those that are at the extreme low and high ends. This is what we would expect from smoothing. For the two panels that compare weighted to unweighted estimates (2 & 4), the unweighted estimates are larger than the weighted estimates, with only a few exceptions. This means that this survey oversampled populations that were more likely to smoke (or it could have oversampled smokers directly). Figure 5 shows the comparison of the direct estimates to those that are weighted and smoothed.

Figure 4: Comparison of smoothed and weighted estimates.

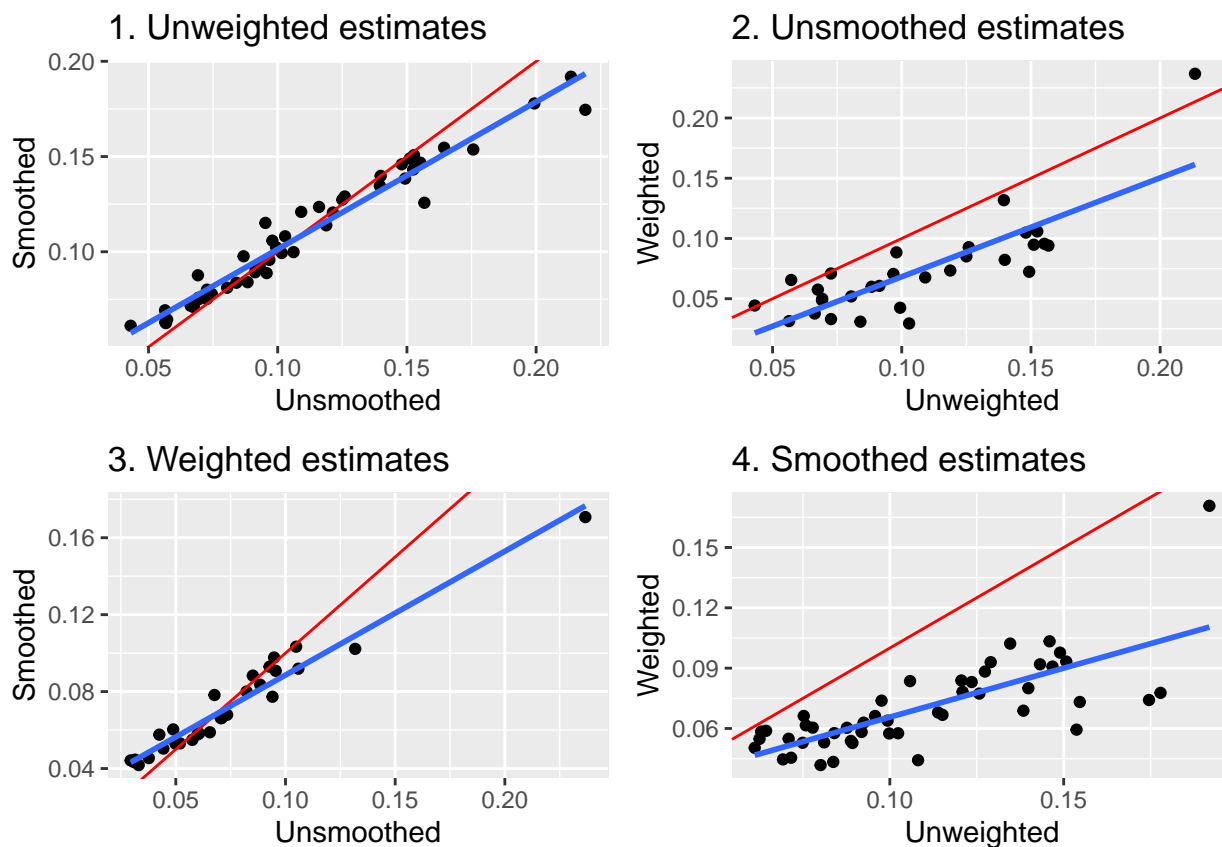
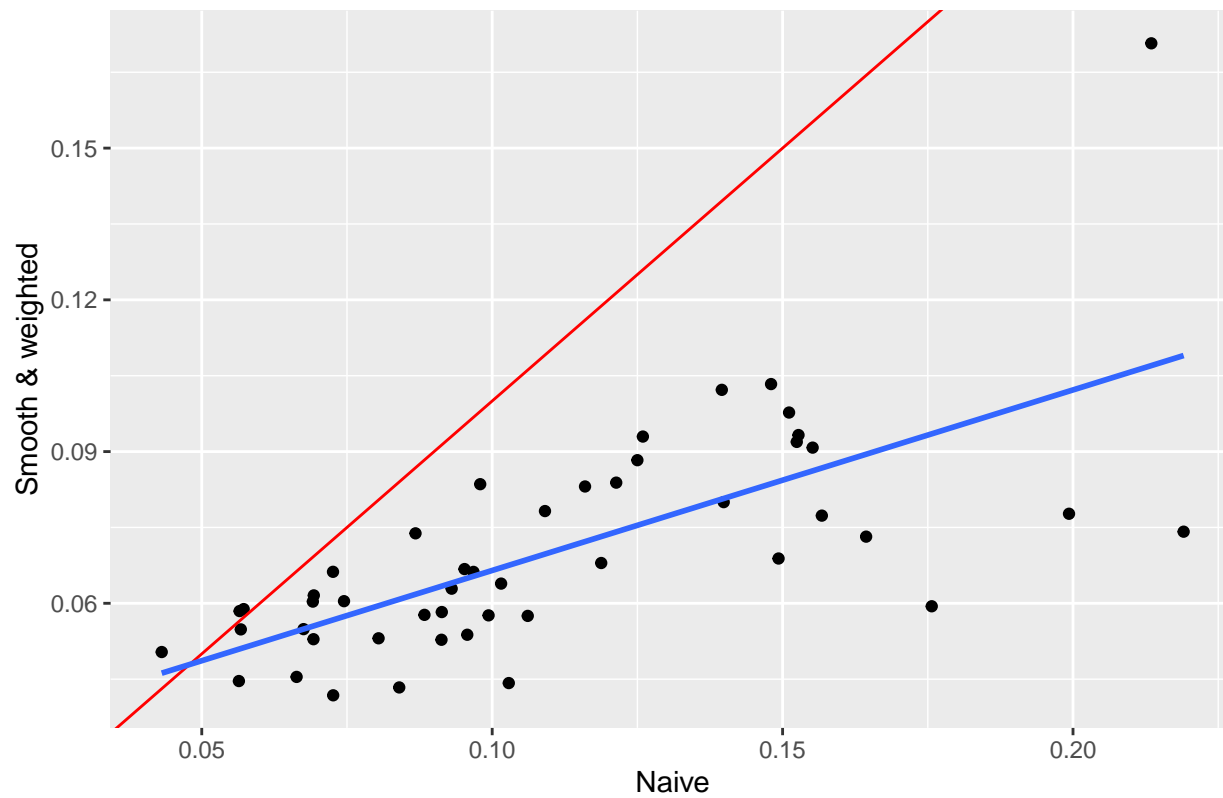


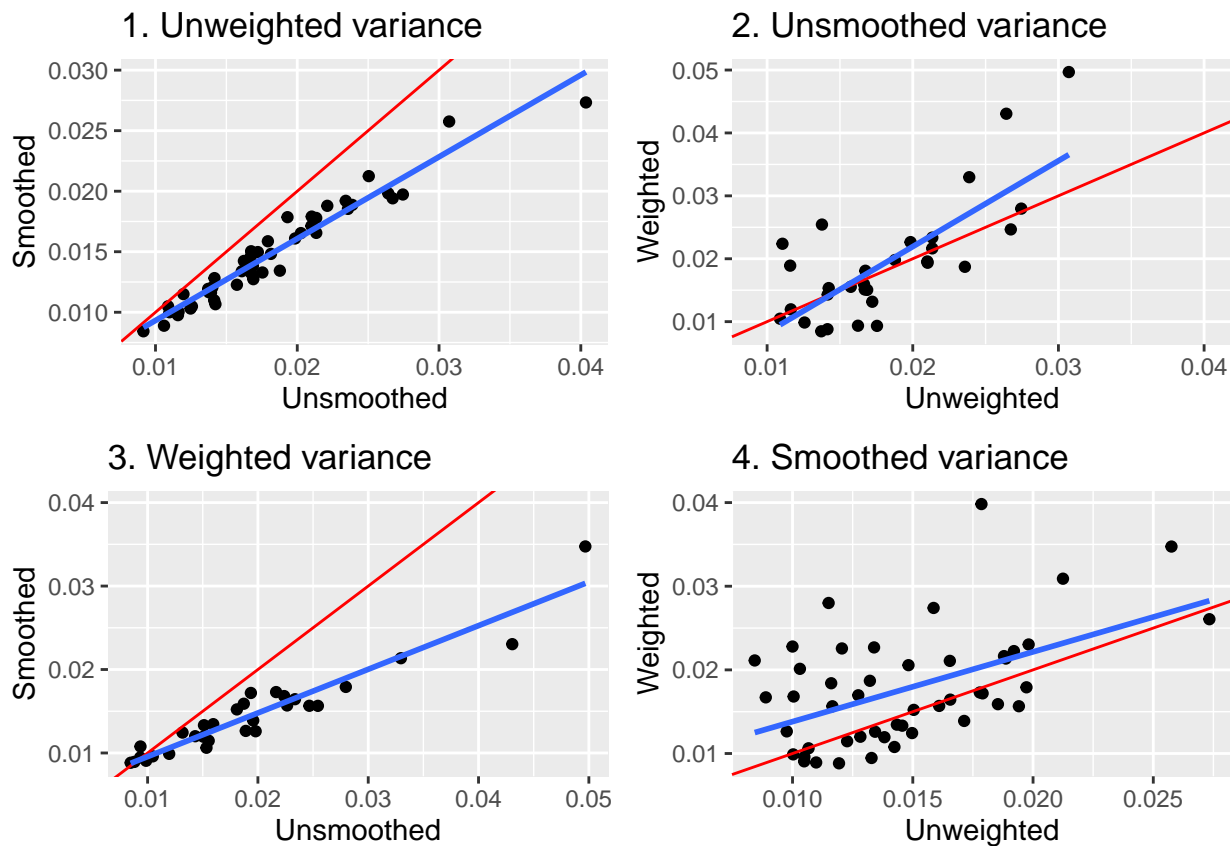
Figure 5: Comparison of direct estimates to smoothed and weighed estimates.

Naive compared to smoothed & weighted



- (c) Comparing the smoothed to unsmoothed standard deviations and standard errors (1 & 3), the smoothed observations are lower in general than the unsmoothed observations. The smoothed and unsmoothed are similar for lower estimates, however, as the standard deviation or standard error gets larger the smoothed estimates become further away from the unsmoothed estimates. Comparing the weighted and unweighted standard deviations and standard errors (2 & 4), there is no pattern or marked difference.

Figure 6: Comparison of variance from smoothed and weighted estimates.

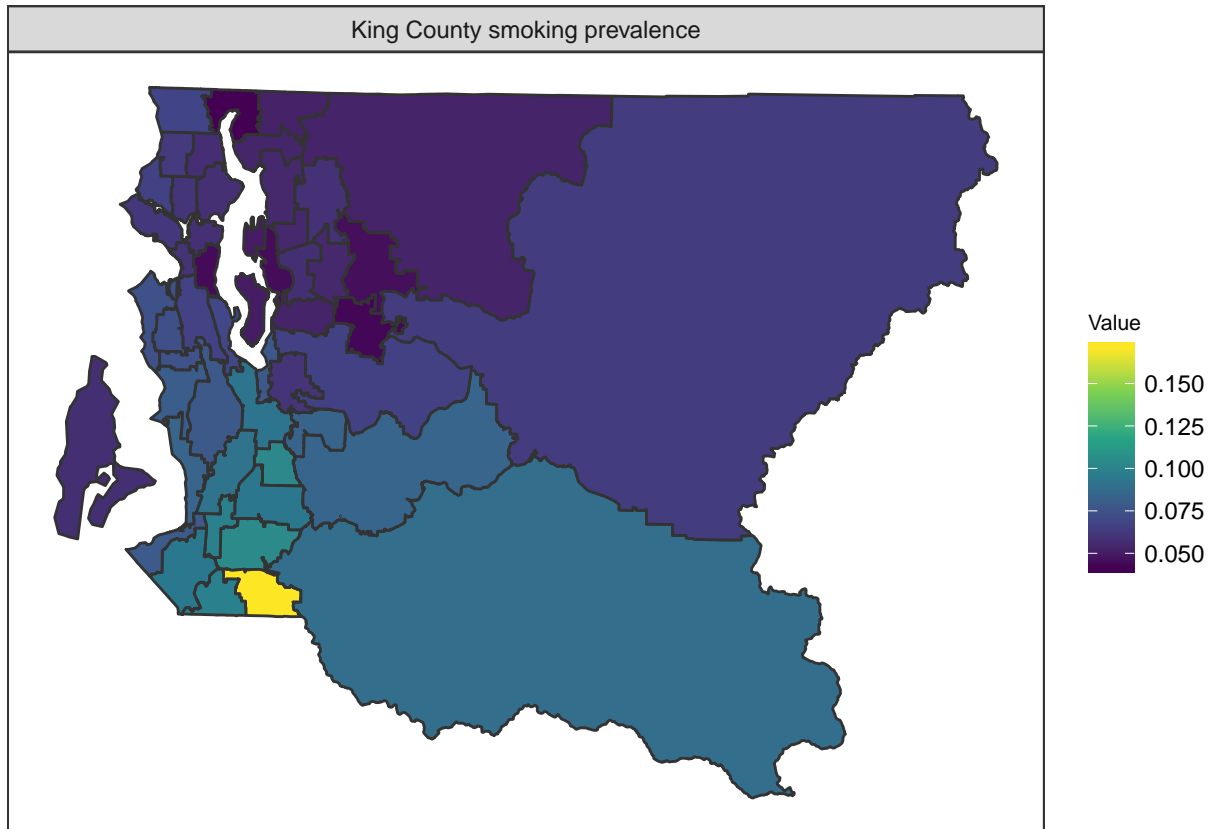


- (d) Since our data set is weighted, we must use a weighted estimate otherwise our estimates will be biased. This leaves us with the choice of using either a smoothed model or an unsmoothed model (direct estimates). Panel 3 of the two prior figures illustrates the comparison of the estimates and the variance between smoothed and unsmoothed weighted estimates. By smoothing the weighted estimates, we gain some precision and pull some of the outlying areas towards the overall mean. The model that we should use (and is represented in Figure 7) is the weighted and smoothed estimates of smoking prevalence in King County by area.

Figure 7: Estimates of the smoking prevalence by area in King County using a spatially smoothed model.

```
## SpP is invalid
```

```
## Using region as id variables
```



Appendix

```
knitr::opts_chunk$set(echo = FALSE, warning=FALSE)

if (!isTRUE(requireNamespace("INLA", quietly = TRUE))) {
  install.packages("INLA", repos = c(getOption("repos"),
    INLA = "https://inla.r-inla-download.org/R/stable"),
    dep=TRUE)
}

if (!require(SUMMER)) install.packages("SUMMER", repos = "http://cran.us.r-project.org")
if (!require(foreign)) install.packages("foreign", repos = "http://cran.us.r-project.org")
if (!require(haven)) install.packages("haven")
if (!require(rgeos)) install.packages("rgeos")
if (!require(rgdal)) install.packages("rgdal")
if (!require(maptools)) install.packages("maptools")
if (!require(sp)) install.packages("sp")
if (!require(spdep)) install.packages("spdep")
if (!require(SpatialEpi)) install.packages("SpatialEpi")
if (!require(RColorBrewer)) install.packages("RColorBrewer")
if (!require(ggplot2)) install.packages("ggplot2")
if (!require(maps)) install.packages("maps")
if (!require(broom)) install.packages("broom")
if (!require(raster)) install.packages("raster")
if (!require(leaflet)) install.packages("leaflet")
if (!require(dplyr)) install.packages("dplyr")
if (!require(tidyverse)) install.packages("tidyverse")
if (!require(SUMMER)) install.packages("SUMMER")

rm(list=ls())
library(foreign)
library(haven)
library(INLA)
library(rgdal)
library(maptools)
library(sp)
library(spdep)
library(SpatialEpi)
library(RColorBrewer)
library(ggplot2)
library(maps)
library(broom)
library(raster)
library(leaflet)
library(dplyr)
library(tidyverse)
library(SUMMER)
library(rgeos)
library(gridExtra)
```

```

#Cancer data
#link = "https://github.com/dmccoomes/Spatial_epi/raw/master/HW%203/Data/ohio_data_ascending_fips.txt"
#ohio_canc <- read.table(url(link), header=TRUE)

#Cancer data
#on pc at home
#ohio_canc <- read.table("/Users/david/Documents/GitHub/Spatial_epi/HW 3/Data/ohio_2019_version.txt", h
#on laptop
#ohio_canc <- read.table("/Users/david/Documents/GitHub/Spatial_epi/HW 3/Data/ohio_2019_version.txt", h

#using github
link = "https://github.com/dmccoomes/Spatial_epi/raw/master/HW%203/Data/ohio_2019_version.txt"
ohio_canc <- read.table(url(link), header=TRUE)

#Map data
#zip_oh_map <- "https://github.com/dmccoomes/Spatial_epi/raw/master/HW%203/Map%20data.zip"

#library(utils)
#temp=tempfile()
#download.file(zip_oh_map, temp)
#unzip(temp)

#(maps=list.files(pattern = 'shp'))

#ohmap <- readOGR(url(zip_oh_map),stringsAsFactors=F)

#maplink <- "https://github.com/dmccoomes/Spatial_epi/raw/master/HW%203/Map%20data/"
#ohmap <- readOGR(dsn = maplink, layer="ohio_map")

#pc in office
#ohmap <- readOGR(dsn="C:\\Users\\dcoomes\\Dropbox\\Classes\\Spatial modeling\\HW 3\\Data\\Map data", l

#laptop
#ohmap <- readOGR(dsn="/Users/david/Documents/GitHub/Spatial_epi/HW 3/Map data", layer="ohio_map")

#computer in library
#ohmap <- readOGR(dsn="C:\\Users\\dcoomes\\Documents\\GitHub\\Spatial_epi\\HW 3\\Map data", layer="ohio

#home pc
ohmap <- readOGR(dsn="/Users/david/Documents/GitHub/Spatial_epi/HW 3/Map data", layer="ohio_map")

#ordering of regions is not the same among the data sets - how do we align these two?
#summary(ohmap)
#ohmap$COUNTYFPO0

#summary(ohio_canc)
#ohio_canc$fips
#View(ohio_canc)

#creating smr

```



```

ohio_canc$smr <- ohio_canc$Obs/ohio_canc$Exp
#ohmap <- readOGR(dsn="C:\\Users\\dcoomes\\Documents\\GitHub\\Spatial_epi\\HW 3\\Data\\Map data", #layer="ohmap")

#When using PC in office
#ohmap <- readOGR(dsn="C:\\Users\\dcoomes\\Desktop\\Epi PhD\\Classes\\Spatial_epi\\Map data", layer="ohmap")

#ordering of regions is not the same among the data sets - how do we align these two?
summary(ohmap)
ohmap$CNTYIDFP00

#summary(ohio_canc)
#ohio_canc$fips
#View(ohio_canc)

#creating RR
ohio_canc$Z <- log(ohio_canc$smr)
ohio_canc$varZ <- 1/(ohio_canc*ohio_canc$smr)
ohio_canc$precZ <- 1/ohio_canc$varZ

ohmap$fips <- as.numeric(as.character(ohmap$CNTYIDFP00))
ohmap <- ohmap[order(ohmap$fips),] #ordering fips so that counties align in data sets
ohmap <- merge(ohmap,ohio_canc,by="fips")

#creating graph file (what is this doing?)
nb.map <- poly2nb(ohmap)
nb2INLA("ohmap.graph", nb.map)

#creating region variable in the cancer data
#ohio_canc <- ohio_canc[order(ohio_canc$fips),] #not sure if I need to do this?
ohio_canc$Region <- 1:nrow(ohmap)

#I get twice the number of regions as there are (176 as opposed to 88)
#ohmap$Region <- ohmap$fips

#for this we are fitting a spatially smoothed ICAR model using the INLA function

#Don't need to define pc.prec for this question
#pc.prec <- list(theta = list(prior = "pc.prec",
#                             param = c(1, 0.05)))

#ohmap2 <- as.data.frame(ohmap)

#setting up the inla model
formula <- Obs ~ 1 +
  f(Region, model="bym2", graph="ohmap.graph", scale.model=T, constr=T,
    hyper=list(phi=list(prior="pc",
                        param=c(0.5, 0.5), initial=1), prec=list(prior="pc.prec",
                                                                    param=c(0.3,0.01), initial=5)))

#fit the inla model

```

```

ohio.fit1 <- inla(formula, data=ohio_canc, family="poisson",
                 E=Exp,
                 control.predictor=list(compute=TRUE))

#ohio.fit1 <- inla(formula, data=ohio_canc,
#                 family="gaussian",
#                 control.predictor=list(compute=TRUE),
#                 control.family = list(hyper = list(prec = list(initial = log(1), fixed=TRUE))),scale=pr

#get estimates
summary(ohio.fit1)
post.med.1 <- ohio.fit1$summary.fixed[4]           #posterior med
post.med.1_low <- ohio.fit1$summary.fixed[3]        #posterior med low
post.med.1_high <- ohio.fit1$summary.fixed[5]       #posterior med high

var.ran <- 1/sqrt(ohio.fit1$summary.hyperpar$`0.5quant`[1])      #total variance of random effects
var.ran_low <- 1/sqrt(ohio.fit1$summary.hyperpar$`0.025quant`[1])
var.ran_high <- 1/sqrt(ohio.fit1$summary.hyperpar$`0.975quant`[1])

prop.1 <- ohio.fit1$summary.hyperpar$`0.5quant`[2]              #proportion of total variance due to random

#diff is the non-spatial random effects
diff <- ohio.fit1$summary.random$Region[1:88, 2] -
        ohio.fit1$summary.random$Region[89:176, 2]

REsnonspat <- exp(diff)
REsspat <- exp(ohio.fit1$summary.random$Region[89:176, 5])

ohmap$REsnonspat <- REsnonspat
ohmap$REsspat <- REsspat

#generating RR estimates
ohmap$postRR <- ohio.fit1$summary.fitted.values$`0.5quant`

spplot(ohmap, c("postRR"), col.regions=colorRampPalette(rev(brewer.pal(8, "RdBu")))(50))

#spplot(ohmap, c("REsnonspat"), col.regions=colorRampPalette(rev(brewer.pal(8, "RdBu")))(50))
#spplot(ohmap, c("REsspat"), col.regions=colorRampPalette(rev(brewer.pal(8, "RdBu")))(50))

#setting priors
pcprec <- list(theta = list(prior = "pc.prec",
                             param = c(1, 0.05)))

#fit non-spatial map
ohio.fit2 <- inla(Obs ~ 1 + f(Region,

```

```

    model = "iid", hyper= pcprec), data = ohio_canc,
    family="poisson", E = Exp, control.predictor = list(compute=TRUE))

summary(ohio.fit2)

expbeta0med <- ohio.fit2$summary.fixed[4]           #post median
sdmed <- 1/sqrt(ohio.fit2$summary.hyperpar[4])      #post sd

expbeta0med
sdmed

ohio.fit2$summary.fixed

lnorminter.2 <- ohio.fit2$summary.fixed[4]
lnormREs.2 <- exp(ohio.fit2$summary.random$Region[5])

lnormRRs.2 <- as.double(exp(lnorminter.2)) * lnormREs.2[,1]

ohmap$RRlnorm.2 <- lnormRRs.2
spplot(ohmap, c("RRlnorm.2"), col.regions=colorRampPalette(rev(brewer.pal(8, "RdBu")))(50))

spplot(ohmap, c("smr"), at = c(0,0.2,0.4,0.6, 0.8, 1, 1.2, 1.4, 1.6), col.regions = rev(brewer.pal(8, "RdBu"))(50))

plot(ohmap$smr, ohio.fit1$summary.fitted.values[,4],
     xlab="Gamma fitted", ylab="Spatial fitted")
abline(0,1)

plot(ohio_canc$smr, ohmap$postRR,
     xlab="Direct estimates (SMR)", ylab="Spatial fitted")
abline(0,1)

plot(ohmap$RRlnorm.2, ohmap$postRR,
     xlab="Non-spatial fitted", ylab="Spatial fitted")
abline(0,1)

#reading in BRFSS data
data(BRFSS)
#summary(BRFSS)

#removing obs with missing hracode and smoking status
brfss <- subset(BRFSS, !is.na(BRFSS$hracode))

```

```

brfss <- subset(brfss, !is.na(brfss$smoker1))

#load spatial data
data(KingCounty)
nb.r <- poly2nb(KingCounty, queen=F,
                row.names = KingCounty$HRA2010v2_)
mat <- nb2mat(nb.r, style="B", zero.policy = TRUE)
colnames(mat) <- rownames(mat)
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
mat[1:2, 1:2]

#calculating weighted non-smoothed estimates
library(survey)
design <- svydesign(ids=~1, weights = ~rwt_llcp,
                  strata = ~strata, data=brfss)
direct <- svyby(~smoker1, ~hrcode, design, svymean)
head(direct, n=2)

#smoothed, non-weighted estimates and non-smoothed, non-weighted estimates
smoothed <- fitGeneric(data=brfss, geo=KingCounty,
                       Amat=mat, responseType = "binary", responseVar="smoker1",
                       strataVar=NULL, weightVar=NULL, regionVar="hrcode",
                       clusterVar=NULL, CI=0.95)
head(smoothed$HT, n=2)

#The smooth estimates are the smoothed non-weighted estimates and the HT estimates are the non-smoothed
head(smoothed$smooth, n=1)
#simple smoothed binomial probabilities
head(smoothed$HT, n=1)

if (!require(gpclib)) install.packages("gpclib", type="source", repos = "http://cran.us.r-project.org")
gpclibPermit()

#mapping smoothed, non-weighted estimates
toplot <- smoothed$smooth
mapPlot(data = toplot, geo=KingCounty,
        variables=c("mean.original"),
        labels=c("Posterior Mean"), by.data="region",
        by.geo="HRA2010v2_")

svysmoothed <- fitGeneric(data=brfss, geo=KingCounty,
                          Amat = mat, responseType="binary", responseVar="diab2",
                          strataVar="strata", weightVar="rwt_llcp", regionVar="hrcode",
                          clusterVar=~1, CI=0.95)

est <- data.frame(naive=smoothed$HT$HT.est.original,

```

```

weighted=svysmoothed$HT$HT.est.original, smooth=smoothed$smooth$mean.original,
weightedsmooth=svysmoothed$smooth$mean.original)

var <- data.frame(naive=smoothed$HT$HT.variance.original,
weighted=svysmoothed$HT$HT.variance.original,
smooth=smoothed$smooth$variance.original, weightedsmooth=svysmoothed$smooth)

se.prop <- sqrt(smoothed$HT$HT.variance.original)           #naive se
se.weighted <- sqrt(svysmoothed$HT$HT.variance.original)    #weighted se
sd.post.med <- sqrt(smoothed$smooth$variance.original)      #smoothed sd
sd.smooth <- sqrt(svysmoothed$smooth$variance.original)      #smoothed and weighted sd

estimates <- data.frame(HRA=smoothed$HT$region, "Simple Proportion"=smoothed$HT$HT.est.original, "Posterior
"Smoothed Direct Estimates"=svysmoothed$smooth$median.original)

estimates[1:48,]

# est <- data.frame(naive = smoothed$HT$HT.est.original,
#                   weighted = svysmoothed$HT$HT.est.original,
#                   smooth = smoothed$smooth$mean.original,
#                   weightedsmooth = svysmoothed$smooth$mean.original)

l1 <- range(est)

total <- ggplot(est, aes(x=naive, y=weightedsmooth)) + geom_point() + geom_abline(slope=1, intercept=0,
unweighted <- ggplot(est, aes(x = naive, y = smooth)) + geom_point() + geom_abline(slope = 1, intercept
unsmoothed <- ggplot(est, aes(x = naive, y = weighted)) + geom_point() + geom_abline(slope = 1, intercept
weighted <- ggplot(est, aes(x = weighted, y = weightedsmooth)) + geom_point() + geom_abline(slope = 1,
smoothed <- ggplot(est, aes(x = smooth, y = weightedsmooth)) + geom_point() + geom_abline(slope = 1, in

# print(unweighted)
# print(unsmoothed)
# print(weighted)
# print(smoothed)

# print(total)

grid.arrange(grobs = list(unweighted, unsmoothed, weighted, smoothed), ncol = 2)

print(total)

```

```

unweighted_var <- ggplot(est, aes(x=se.prop, y=sd.post.med)) + geom_point() + geom_abline(slope=1, inte
unsmoothed_var <- ggplot(est, aes(x=se.prop, y=se.weighted)) + geom_point() + geom_abline(slope=1, inte
weighted_var <- ggplot(est, aes(x=se.weighted, y=sd.smooth)) + geom_point() + geom_abline(slope=1, inte
smoothed_var <- ggplot(est, aes(x=sd.post.med, y=sd.smooth)) + geom_point() + geom_abline(slope=1, inte

grid.arrange(grobs = list(unweighted_var, unsmoothed_var, weighted_var, smoothed_var), ncol = 2)

#print(g1)

#mapping smoothed, non-weighted estimates
toplot <- svysmoothed$smooth

mapPlot(data = toplot, geo=KingCounty,
        variables=c("mean.original"),
        labels=c("King County smoking prevalence"), by.data="region",
        by.geo="HRA2010v2_")

```