CSSS 554: Assignment 2

Walker Azam
Due: February 6th 2023

Assignment 2 Overview

For this problem we will use disease mapping for lung cancer mortality data for men in the Valencia region of Spain from 1991-2000.

The data files that will be used contains, for each subarea: observed deaths and expected deaths (adjusted for reference rate only), polygon files, and a graph file for INLA.

We I will load libraries to be used:

```r
# Loading in libraries
library(SpatialEpi)
library(RColorBrewer)
library(ggplot2)
library(ggridges)
library(INLA)
library(sf)
```

Question 0: Loading in Data

The following code will be used to read in the data for this assignment:

```r
# Set your working directory to where HW2data.Rdata and VR.graph are
# located
setwd("./Data")
load("HW2data.Rdata")
# Expected counts are stored in Exp.mv3[,'Lung'] Observed counts are
# stored in Obs.mv3[,'Lung'] The shapefile for plotting is VR.cart
```

Question 1

Let Y_i and E_i, $i = 1, ..., n$, denote the observed and expected counts in region i, $i = 1, ..., n$. Then consider the model

$$\mathrm{Y}_i|E_i\ Poisson(E_i\theta_i)$$

Question 1(a): Provide a map of the observed counts Yi

First let's convert our data to a spatial object:

```r
# Reading in the expected and observed lung cancer vectors
expected <- Exp.mv3[, "Lung"]
cases <- Obs.mv3[, "Lung"]
# Adding them as columns on VR.cart
VR.cart$expected <- expected
VR.cart$cases <- cases
# Saving it as a dataframe object
data <- VR.cart@data
# Adding SMR column
data$SMR <- data$cases/data$expected
```

We can convert `dmap` to an `sfc` object, and add it as geometry column to the dataframe, then convert into an `sf` object:

```r
data$geometry <- st_as_sfc(VR.cart)
dmap <- st_as_sf(data)
```

Now we can map Y_i, which is the observed count of cases for Lung Cancer:

```r
pal = function(n) brewer.pal(n, "Oranges")
plot(dmap["cases"], nbreaks = 8, pal = pal, breaks = "equal", main = "Observed Lung Cancer Cases",
    sub = "Yi for Valencia")
```

Observations (Figure 1):

- It seems that just plotting observed cases doesn't tell us anything too revealing. There is one region, in the mid-east, that appears like a hot spot with over 7500 cases, but most of the areas are all around, or less, than 1000 total cases

Question 1(b): Provide a map of the expected counts Ei

Using the same spatial object dataframe, we can map E_i, which is the Expected count of cases for Lung Cancer:

```r
pal = function(n) brewer.pal(n, "Blues")
plot(dmap["expected"], nbreaks = 8, pal = pal, breaks = "equal", main = "Expected Lung Cancer Cases",
    sub = "Ei for Valencia")
```

Observations (Figure 2):

- Again, it seems that just plotting expected cases alone doesn't tell us anything too revealing.
- The hot spot regions seem to align in severity of expected and actual cases, with most areas all being well below 1000 cases.
- These plots demonstrate that there may be extreme variance in cases between regions, ranging from under 10 to well over 7500.

The variance of the estimate in area i is

$$\text{var}(\text{SMR}_i) = \frac{\text{SMR}_i}{E_i},$$

which will be large if E_i is small. So from that we can see the summary of expected:

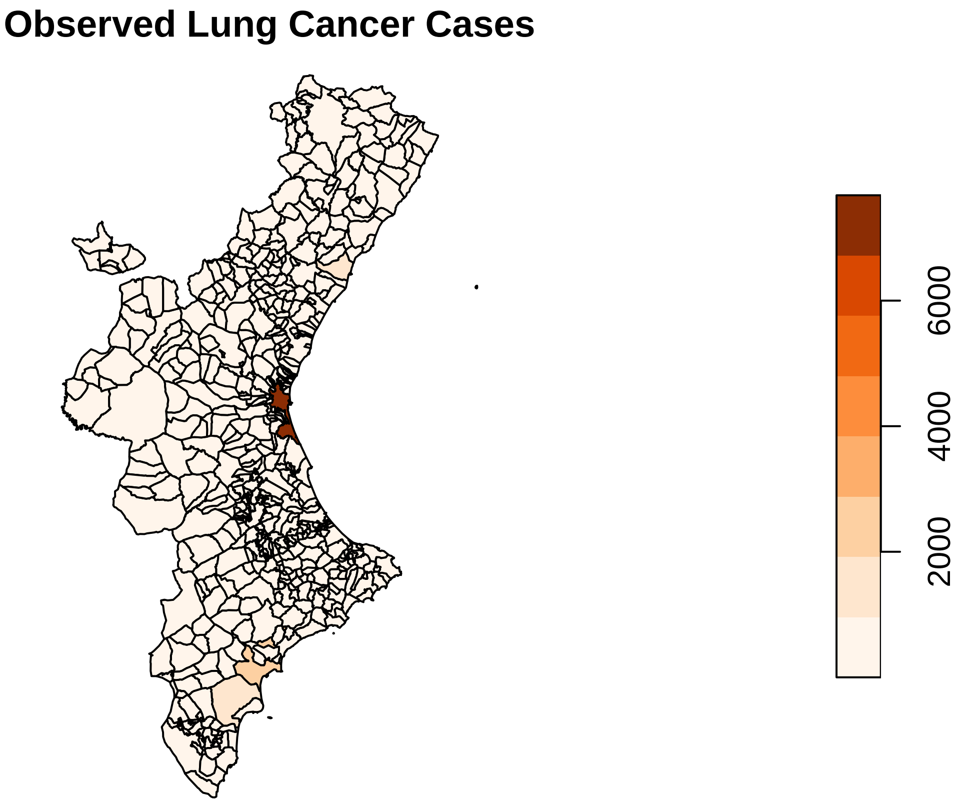


Figure 1: Observed Cases for Lung Cancer Data In Valencia

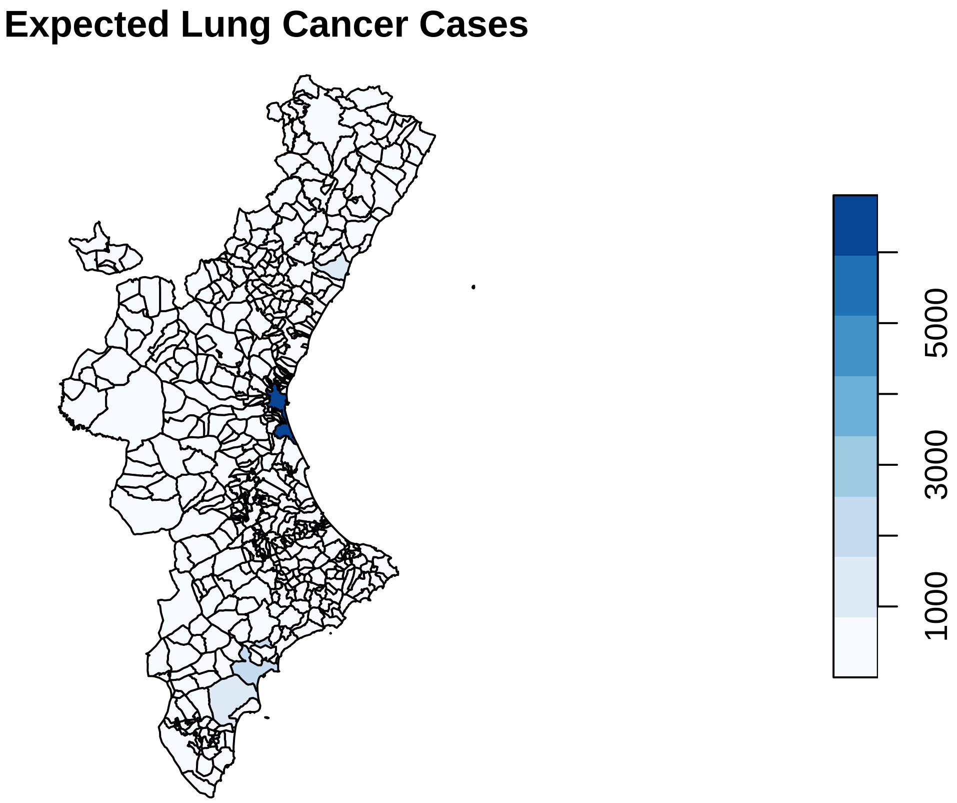


Figure 2: Expected Cases for Lung Cancer Data In Valencia

```
summary(dmap$expected)
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.228    6.892   17.169   73.000   49.951 6804.364
```

For the lung cancer data the expected numbers are highly variable, with range 0.228–6804.364.

From this range we can expect that extreme SMRs may be based on small Expected cases (possibly due to rural areas being underpopulated).

Question 1(c): Provide a map of the SMRs

SMRs are defined as

$$\mathrm{SMR}_i = (\widehat{\theta}_i) = \frac{Y_i}{E_i},$$

for $i = 1, ..., n$.

We already calculated the SMRs previously, so we can plot them below:

```
pal = function(n) brewer.pal(n, "Purples")
plot(dmap["SMR"], nbreaks = 8, pal = pal, breaks = "equal", main = "SMRs for Lung Cancer Cases")
```

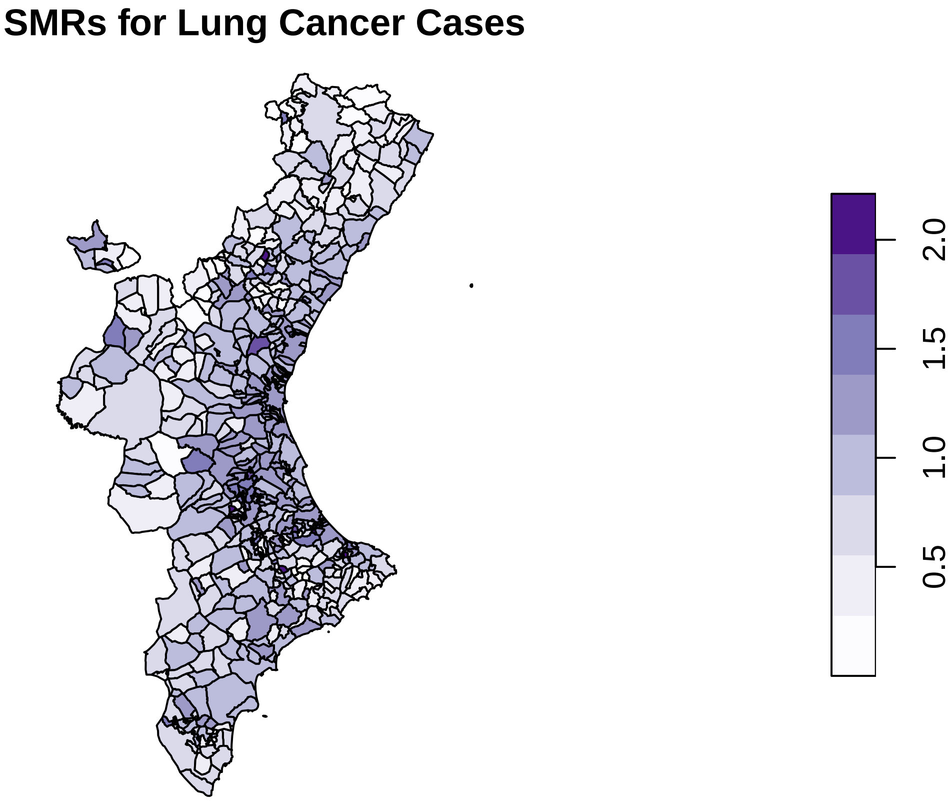


Figure 3: SMRs Lung Cancer Data In Valencia

```
summary(dmap$SMR)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.5893  0.8704  0.8480  1.1137  2.2108
```

Observations:

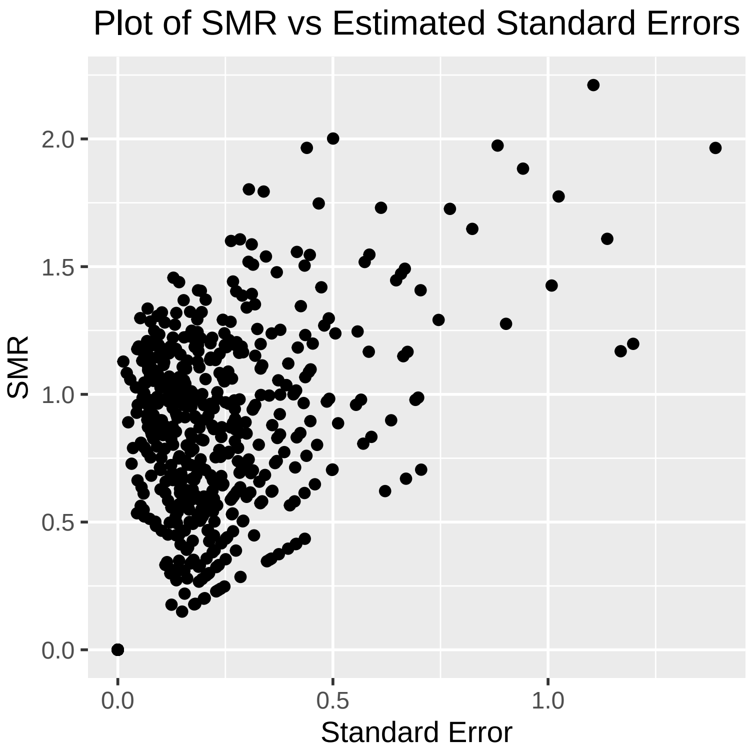
- The SMRs don't seem to have too large a spread, ranging from 0.0 to 2.21.
- We can see that the larger regions towards the West (and further North/South) have a lower SMR. Whereas near the East boundary (and towards the center of the area) slightly higher SMRs can be seen.
- From our plot of expected cases, we saw a large range, so possibly the larger SMRs may be due to sparseness. Moreover, how much does this results are from sampling variation, rather than true variation, is yet to be seen?

Question 1(d): Plot the SMRs versus the estimated standard errors

The standard errors are given by $\sqrt{\widehat{\theta}_i/E_i}$.

We can use ggplot to help plot:

```
ggplot(data.frame(se = sqrt(data$SMR/data$expected), SMR = data$SMR),
    aes(x = se, y = SMR)) + geom_point() + labs(y = "SMR", x = "Standard Error",
    title = "Plot of SMR vs Estimated Standard Errors") + theme(plot.title = element_text(hjust = 0.5))
```



- The highest SMRs certainly have the highest Standard Errors
- Some Standard Errors are 0, which arise from 0 SMR values (0 estimates). This may be considered for 0.5 adjustments, but for the purposes of this assignment, will be left as 0.

Question 2: Smoothing SMRs using Disease Mapping Poisson-Lognormal Models

In this question I will smooth the SMRs using the disease mapping Poisson-Lognormal model as given:

$$\begin{aligned} Y_i|\beta_0, e_i &\sim_{ind} \text{Poisson}(E_i \mathrm{e}^{\beta_0} \mathrm{e}^{e_i}), \\ e_i|\sigma_e^2 &\sim_{iid} \text{N}(0, \sigma_e^2) \end{aligned}$$

for i, $i = 1, ..., n$.

Question 2(a): Using the inla function in R fit this model using the default priors for B_0 and σ_e.

(Report the posterior medians and 95% intervals for β_0 and for σ_e.)

I will fit the Poisson-Lognormal model in INLA to the lung cancer data, with no prior specifications given (default prior):

```r
# Fit Poisson-Lognormal model in INLA:
model.fit0 <- inla(cases ~ 1 + f(MUNI_ID, model = "iid"), data = dmap,
    family = "poisson", E = expected)
```

To fit the model above, we used MUNI_ID as an index to assign random effects to each observation, since it functions as a Region Identifier. The Poisson family was also specified, with default priors. We can see what the summary (posterior median and 95% intervals) are for both β_0 and σ_e:

```r
# Results for the \beta_0:
model.fit0$summary.fixed[, 1:5]
##                   mean         sd 0.025quant   0.5quant 0.975quant
## (Intercept) -0.1337818 0.01603758 -0.1657331 -0.1336088 -0.1028069
```

- For β_0 the posterior median is -0.1336088, and the interval is -0.1657331 to -0.1028069

```r
# Results for the precision:
model.fit0$summary.hyper[, 1:5]
##                             mean       sd 0.025quant 0.5quant 0.975quant
## Precision for MUNI_ID 15.76215 1.858711   12.45121  15.6481   19.72706
```

```r
# Results for the sigma_e:
sigma <- 1/sqrt(model.fit0$summary.hyper[, 4])
lower <- 1/sqrt(model.fit0$summary.hyper[, 3])
upper <- 1/sqrt(model.fit0$summary.hyper[, 5])
print(c(sigma, lower, upper))
## [1] 0.2527954 0.2833964 0.2251484
```

- For σ_e the posterior median is 0.2527954, and the interval is 0.2251484 to 0.2833964

Question 2(b): Extract the posterior medians of the relative risk (RR) estimates and provide a map of these.

We can map the posterior medians of the relative risks:

```r
# Extracting the posterior medians using our fitted model
dmap$fit0fitted <- model.fit0$summary.fitted.values$`0.5quant`
# Mapping our results
pal = function(n) brewer.pal(n, "Purples")
plot(dmap["fit0fitted"], pal = pal, nbreaks = 8, breaks = "equal", main = "Map of Posterior Medians")
```

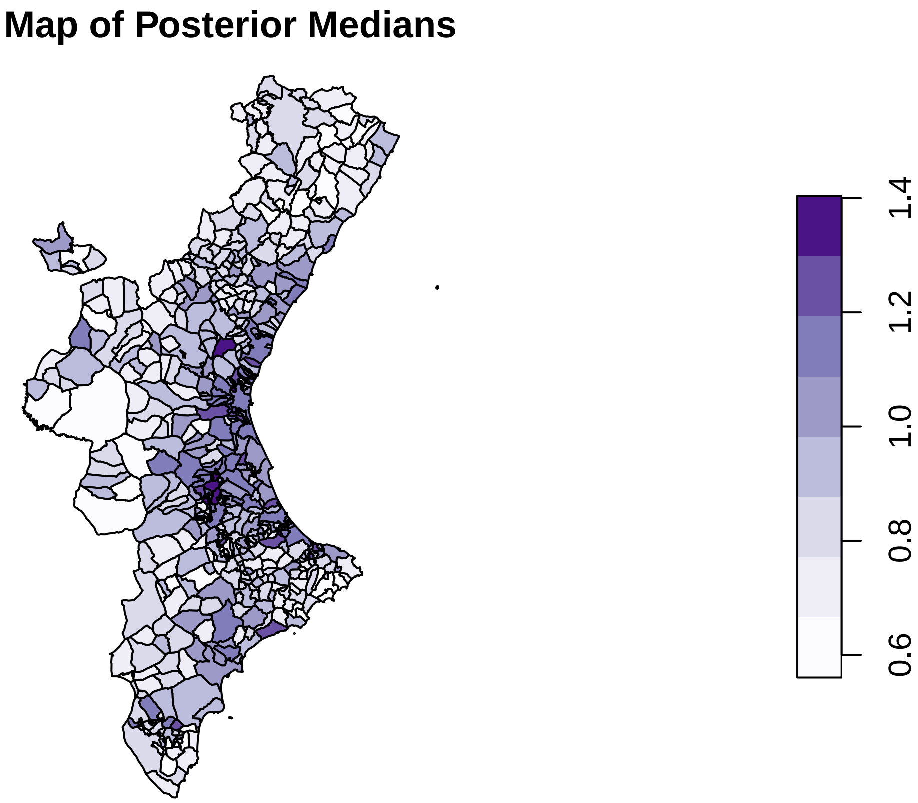
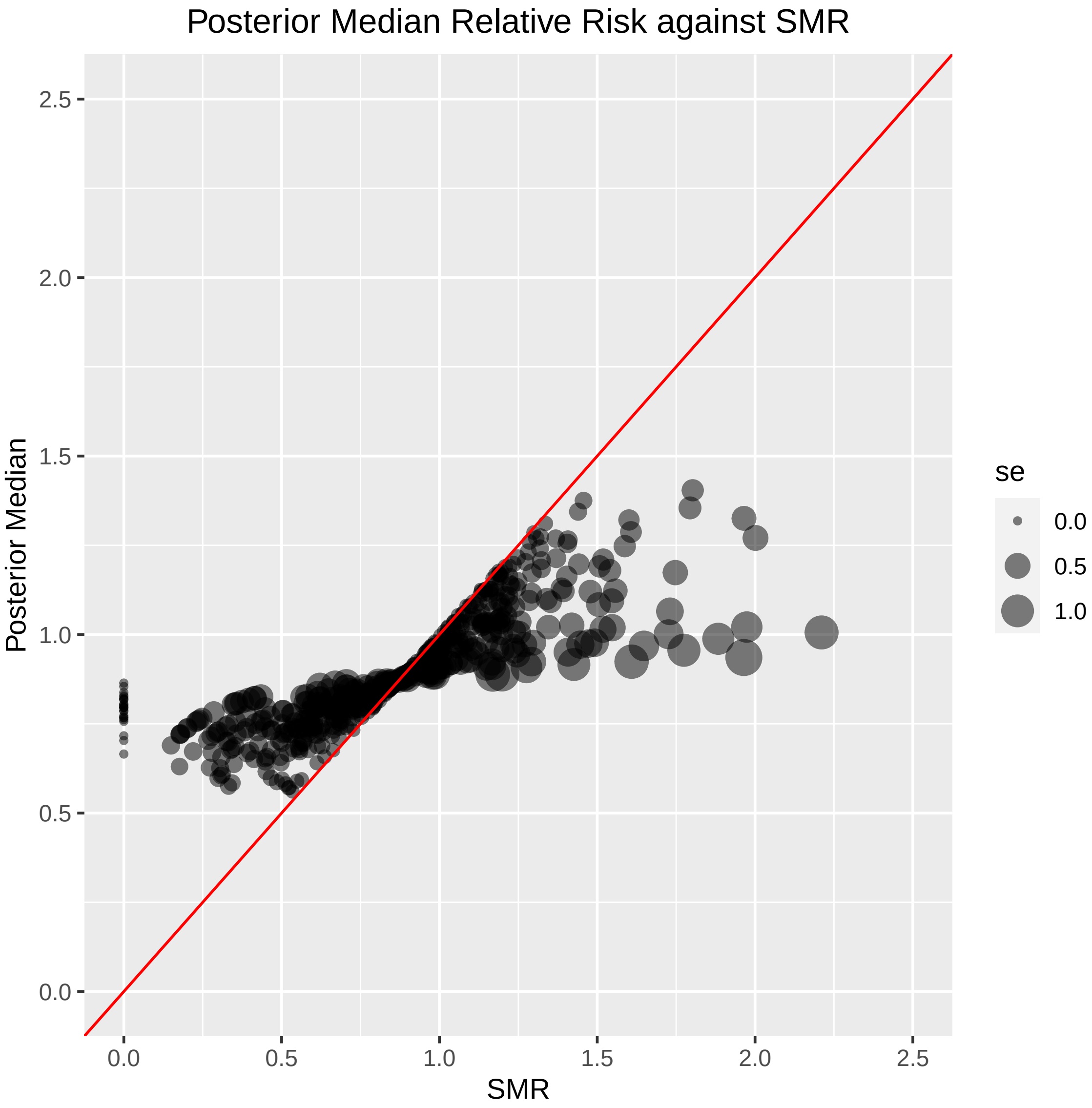


Figure 4: Posterior Medians for Relative Risk with default priors

Question 2(c): Plot these posterior RR estimates against the SMRs, and comment.

```r
# Find the Standard Errors
se <- sqrt(dmap$SMR/dmap$expected)
# Plotting Posterior Median vs SMR
ggplot(data.frame(pmedian = model.fit0$summary.fitted.values$`0.5quant`,
    SMR = dmap$SMR), aes(y = pmedian, x = SMR, size = se)) + geom_point(alpha = 0.5) +
    labs(y = "Posterior Median", x = "SMR") + labs(y = "Posterior Median",
    x = "SMR", title = "Posterior Median Relative Risk against SMR") +
    theme(plot.title = element_text(hjust = 0.5)) + geom_abline(intercept = 0,
    slope = 1, color = "red") + xlim(0, 2.5) + ylim(0, 2.5)
```

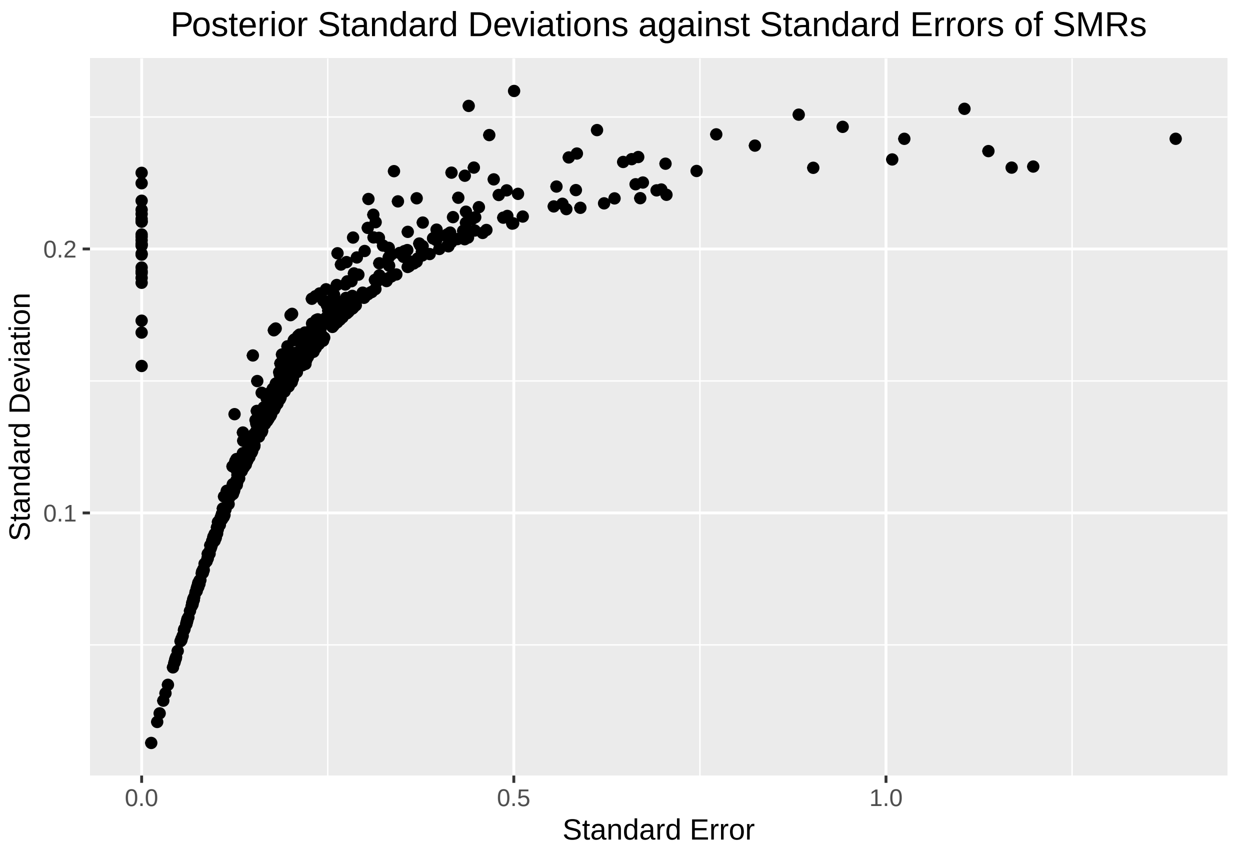


Observations:

- We can see shrinkage at the extreme SMR values the most. The posterior median range is much smaller than the original SMR values (from 0 to 2.2)
- High SMRs also have much higher standard errors than SMR values around or less than 1.
- Points with higher variability (higher Standard Error) are seeing more shrinkage (high SMR values).
- Points close to the red line all have smaller standard errors!

Question 2(d): Plot the posterior standard deviations of the RRs against the standard errors of the SMRs and comment.

```r
# Find the Standard Errors
se <- sqrt(dmap$SMR/dmap$expected)
# Plotting standard deviations vs standard errors
ggplot(data.frame(psd = model.fit0$summary.fitted.values$sd, se = se),
    aes(y = psd, x = se)) + geom_point() + labs(y = "Standard Deviation",
    x = "Standard Error", title = "Posterior Standard Deviations against Standard Errors of SMRs") +
    theme(plot.title = element_text(hjust = 0.5))  #+ geom_abline(intercept=0,slope=1,color='red') + xl
```



Observations:

- We can see that the posterior Standard Deviation is smaller than the Standard Errors of the SMRs.
- Since the standard deviations of the posterior is an analog to the standard errors, we can note a reduction in uncertainty from the INLA model, due to smoothing.

Question 3: Smoothing SMRs using Disease Mapping Poisson-Lognormal-Spatial Model

I will smooth the SMRs for the Valencian lung cancer data using the disease mapping Poisson-Lognormal-Spatial model in this question.

Question 3(a): INLA with BYM2 model

I will fit a INLA function using the bym2 model, with default β_0 priors and given prior for spatial and non-spatial effects as follows:

```
# Assigning a region column with a numeric identifier
dmap$region <- 1:nrow(data)
# Running INLA with given specifications
formula <- cases ~ 1 + f(region, model = "bym2", graph = "./Data/VR.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)))

model.fit1 <- inla(formula, data = dmap, family = "poisson", E = expected,
    control.predictor = list(compute = TRUE), control.compute = list(return.marginals.predictor = TRUE,
        config = TRUE))

model.fit1$summary.fixed[, 1:5]
##                   mean         sd 0.025quant   0.5quant 0.975quant
## (Intercept) -0.1526172 0.01261572  -0.177555 -0.1525523 -0.1280541
model.fit1$summary.hyper[, 1:5]
##                            mean         sd 0.025quant  0.5quant 0.975quant
## Precision for region 15.3874780 2.15072473 11.5623791 15.248802 20.0202752
## Phi for region        0.9528119 0.03878178  0.8514022  0.963306  0.9954758
```

```
1/sqrt(model.fit1$summary.hyper[1, 4])
## [1] 0.2560838
model.fit1$summary.hyper[2, 4]
## [1] 0.963306
```

Report both the posterior medians and 95% intervals for B0, the total variance of the random effects, and the proportion of the total variance attributed to the spatial random effect:

For β_0 the posterior median is -0.1525523 and the 95% interval is -0.177555 to -0.1280541.

The posterior median of the total standard variance of random effects is: 0.2560838

The posterior median for the proportion of the residual variation that is spatial ϕ is: 0.963306.

Question 3(b): Map Relative Risk Estimates

We can map the fitted relative risk estimates below:

```
dmap$model1fitted <- model.fit1$summary.fitted.values$`0.5quant`
plot(dmap["model1fitted"], pal = pal, nbreaks = 8, breaks = "equal", main = "Map of Posterior Medians (S
```

Comparing them to the Possion-Lognormal model (IID model from Q2, plotted below for convenience) I can make these observations:

- We can see that the BYM2 random effects ‘borrow’ from neighbors. So we can see less stark differences in neighboring region’s relative risks.
- For example, in the IID model, we can see higher hot spots for the Posterior Medians (around 1.4) and some spots that are much ‘whiter’. In contrast the bym2 model has less extreme differences in the color range, specially for neighboring areas.
- This is clear in the southern half of the bym2 map where most values range around 0.8, but in the IID map, the same region has 0.6-1.0.

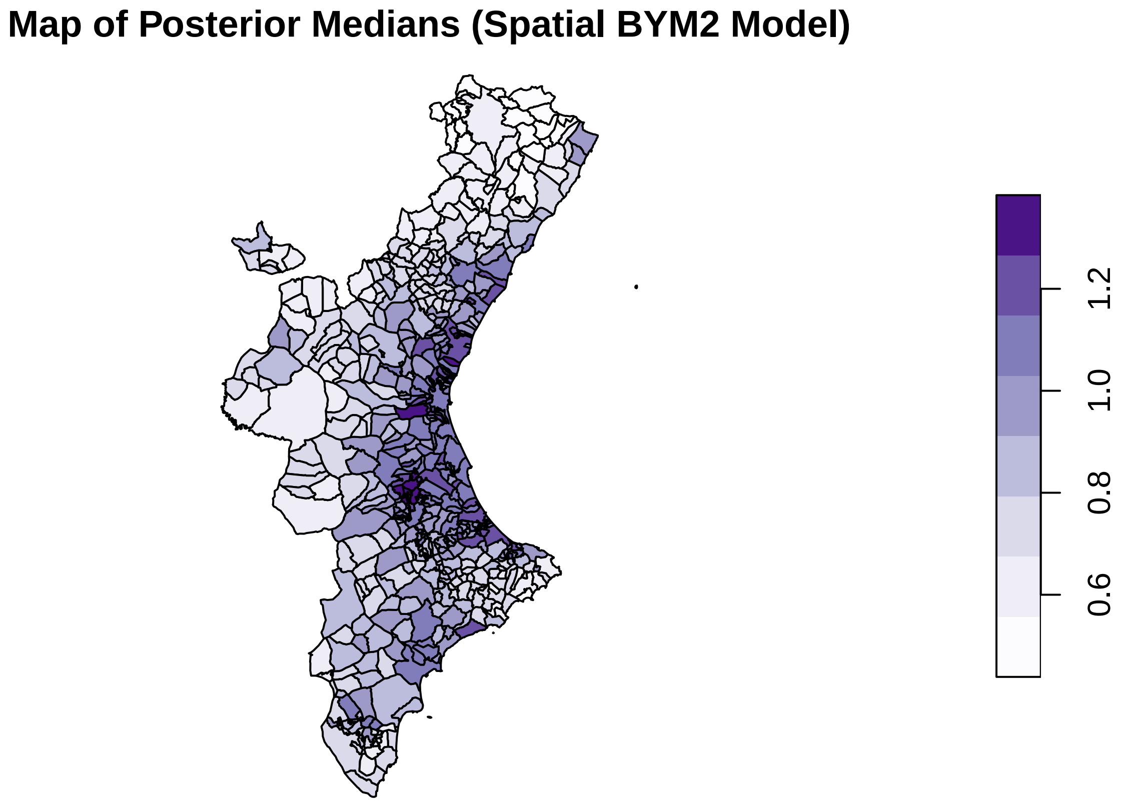


Figure 5: Posterior Medians for Poisson-Lognormal-Spatial Model

```
# Extracting the posterior medians using our fitted model
dmap$fit0fitted <- model.fit0$summary.fitted.values$`0.5quant`
# Mapping our results
pal = function(n) brewer.pal(n, "Purples")
plot(dmap["fit0fitted"], pal = pal, nbreaks = 8, breaks = "equal", main = "REFERENCE MAP FROM Q2: Map o
```

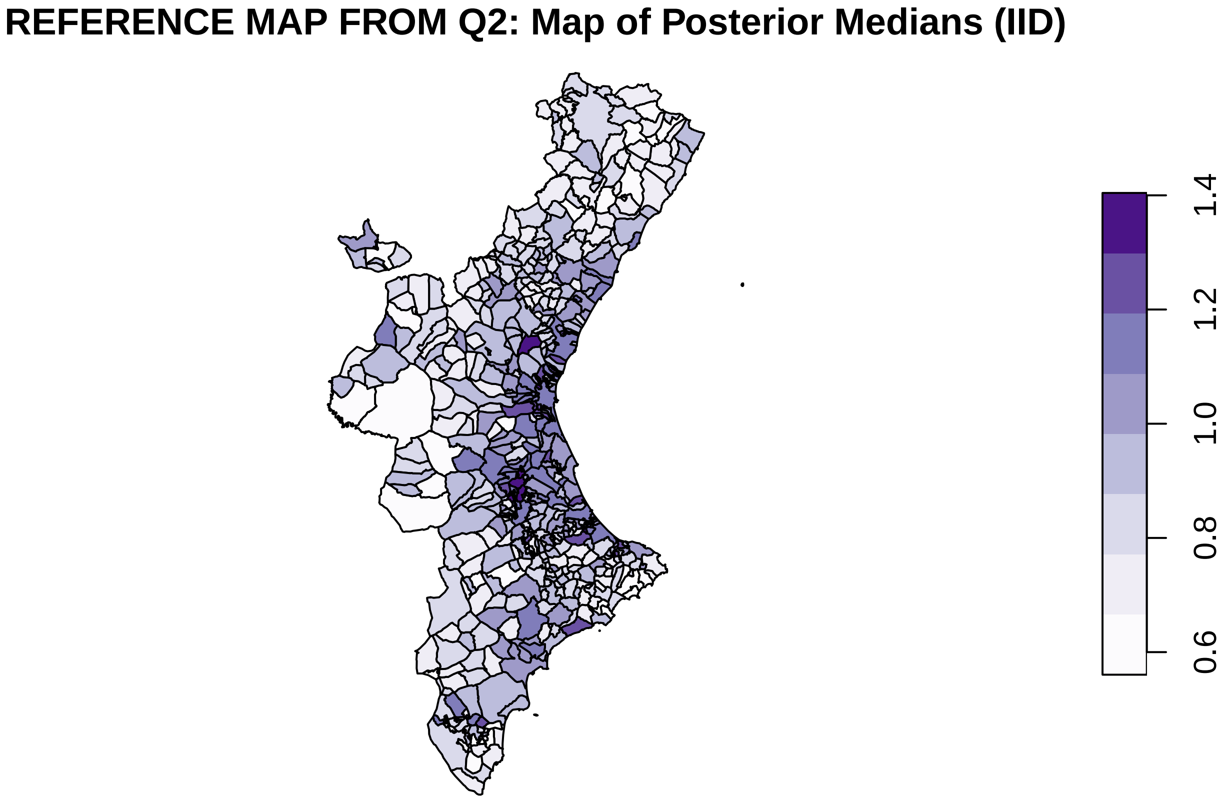


Figure 6: REFERENCE MAP FROM Q2: Posterior Medians for Relative Risk with default priors

Appendix: R-Code Provided

Below I am providing the R code used in this assignment as an Appendix:

```
# Setting up RMarkdown
library(knitr)
opts_chunk$set(message = FALSE, collapse = TRUE, fig.align = "center",
    tidy = TRUE, tidy.opts = list(blank = TRUE, width.cutoff = 70, strip.white = TRUE),
    warning = FALSE, cache = FALSE)
# Loading in libraries
library(SpatialEpi)
library(RColorBrewer)
library(ggplot2)
library(ggridges)
library(INLA)
```

```r
library(sf)
# Set your working directory to where HW2data.Rdata and VR.graph are
# located
setwd("./Data")
load("HW2data.Rdata")
# Expected counts are stored in Exp.mv3[,'Lung'] Observed counts are
# stored in Obs.mv3[,'Lung'] The shapefile for plotting is VR.cart
# Reading in the expected and observed lung cancer vectors
expected <- Exp.mv3[, "Lung"]
cases <- Obs.mv3[, "Lung"]
# Adding them as columns on VR.cart
VR.cart$expected <- expected
VR.cart$cases <- cases
# Saving it as a dataframe object
data <- VR.cart@data
# Adding SMR column
data$SMR <- data$cases/data$expected
data$geometry <- st_as_sfc(VR.cart)
dmap <- st_as_sf(data)
pal = function(n) brewer.pal(n, "Oranges")
plot(dmap["cases"], nbreaks = 8, pal = pal, breaks = "equal", main = "Observed Lung Cancer Cases",
    sub = "Yi for Valencia")
pal = function(n) brewer.pal(n, "Blues")
plot(dmap["expected"], nbreaks = 8, pal = pal, breaks = "equal", main = "Expected Lung Cancer Cases",
    sub = "Ei for Valencia")
summary(dmap$expected)
pal = function(n) brewer.pal(n, "Purples")
plot(dmap["SMR"], nbreaks = 8, pal = pal, breaks = "equal", main = "SMRs for Lung Cancer Cases")
summary(dmap$SMR)
ggplot(data.frame(se = sqrt(data$SMR/data$expected), SMR = data$SMR), aes(x = se,
    y = SMR)) + geom_point() + labs(y = "SMR", x = "Standard Error", title = "Plot of SMR vs Estimated S
    theme(plot.title = element_text(hjust = 0.5))
# Fit Poisson-Lognormal model in INLA:
model.fit0 <- inla(cases ~ 1 + f(MUNI_ID, model = "iid"), data = dmap,
    family = "poisson", E = expected)
# Results for the \beta_0:
model.fit0$summary.fixed[, 1:5]
# Results for the precision:
model.fit0$summary.hyper[, 1:5]
# Results for the sigma_e:
sigma <- 1/sqrt(model.fit0$summary.hyper[, 4])
lower <- 1/sqrt(model.fit0$summary.hyper[, 3])
upper <- 1/sqrt(model.fit0$summary.hyper[, 5])
print(c(sigma, lower, upper))
# Extracting the posterior medians using our fitted model
dmap$fit0fitted <- model.fit0$summary.fitted.values$`0.5quant`
# Mapping our results
pal = function(n) brewer.pal(n, "Purples")
plot(dmap["fit0fitted"], pal = pal, nbreaks = 8, breaks = "equal", main = "Map of Posterior Medians")
# Find the Standard Errors
se <- sqrt(dmap$SMR/dmap$expected)
# Plotting Posterior Median vs SMR
ggplot(data.frame(pmedian = model.fit0$summary.fitted.values$`0.5quant`,
    SMR = dmap$SMR), aes(y = pmedian, x = SMR, size = se)) + geom_point(alpha = 0.5) +
    labs(y = "Posterior Median", x = "SMR") + labs(y = "Posterior Median",
    x = "SMR", title = "Posterior Median Relative Risk against SMR") +
    theme(plot.title = element_text(hjust = 0.5)) + geom_abline(intercept = 0,
    slope = 1, color = "red") + xlim(0, 2.5) + ylim(0, 2.5)
# Find the Standard Errors
se <- sqrt(dmap$SMR/dmap$expected)
# Plotting standard deviations vs standard errors
ggplot(data.frame(psd = model.fit0$summary.fitted.values$sd, se = se),
    aes(y = psd, x = se)) + geom_point() + labs(y = "Standard Deviation",
    x = "Standard Error", title = "Posterior Standard Deviations against Standard Errors of SMRs") +
    theme(plot.title = element_text(hjust = 0.5))  #+ geom_abline(intercept=0,slope=1,color='red') + xl
# Assigning a region column with a numeric identifier
dmap$region <- 1:nrow(data)
# Running INLA with given specifications
formula <- cases ~ 1 + f(region, model = "bym2", graph = "./Data/VR.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)))

model.fit1 <- inla(formula, data = dmap, family = "poisson", E = expected,
    control.predictor = list(compute = TRUE), control.compute = list(return.marginals.predictor = TRUE,
        config = TRUE))

model.fit1$summary.fixed[, 1:5]
model.fit1$summary.hyper[, 1:5]
1/sqrt(model.fit1$summary.hyper[1, 4])
model.fit1$summary.hyper[2, 4]
dmap$model1fitted <- model.fit1$summary.fitted.values$`0.5quant`
plot(dmap["model1fitted"], pal = pal, nbreaks = 8, breaks = "equal", main = "Map of Posterior Medians (
# Extracting the posterior medians using our fitted model
dmap$fit0fitted <- model.fit0$summary.fitted.values$`0.5quant`
# Mapping our results
pal = function(n) brewer.pal(n, "Purples")
plot(dmap["fit0fitted"], pal = pal, nbreaks = 8, breaks = "equal", main = "REFERENCE MAP FROM Q2: Map o
```