Tarea entregable

Disease mapping:Modelo logistico - INLA - Reproducibilidad

Santiago Moncalero, Diego Fernández y Jorge Guitart

Introduccion

En el contexto del Máster de Bioestadística de la Universitat de València, la reproducibilidad y la replicabilidad se reconocen como pilares fundamentales de la investigación científica. Este principio asegura que los análisis estadísticos puedan ser verificados por terceros y reutilizados en nuevos contextos, fortaleciendo la confianza en los resultados obtenidos y su validez empírica

En esta práctica, se aborda un análisis reproducible basado en el mapeo de enfermedades (disease mapping), utilizando dos enfoques metodológicos: WingBUGS e INLA. Ambos modelos fueron implementados para estimar tasas de mortalidad y probabilidades ajustadas espacialmente en la región de Aragón. La comparación entre los resultados busca evaluar la consistencia entre ambos métodos y destacar diferencias clave en sus predicciones, empleando herramientas de reproducibilidad como RMarkdown, proyectos de R y rutas relativas

Carga de Datos

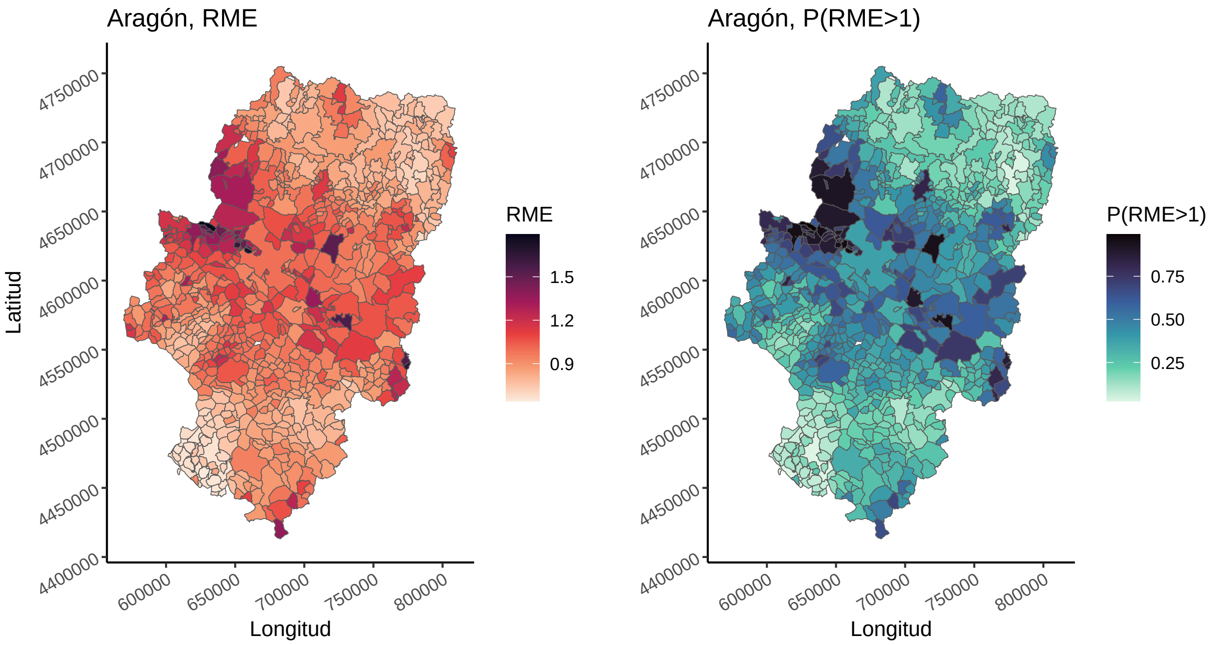
Para llevar a cabo este análisis reproducible, se utilizaron datos correspondientes a la región de Aragón, los cuales incluyen información geoespacial y de mortalidad. Los pasos de carga y preparación de los datos se llevaron a cabo utilizando rutas relativas y herramientas de R para garantizar la reproducibilidad del trabajo, independientemente de la máquina o el directorio de ejecución

```r
# Programamos el modelo
mod1 <- function() {
  for (i in 1:nObs) {
    O[i] ~ dpois(mu[i])
    log(mu[i]) <- log(E[i]) + m + het[i] + sp[i]
    het[i] ~ dnorm(0, prechet)
    R[i] <- exp(m + het[i] + sp[i])
  }

  sp[1:nObs] ~ car.normal(adj[], w[], num[], precsp)
  m ~ dflat()
  prechet <- pow(sdhet, -2); precsp <- pow(sdsp, -2)
  sdhet ~ dunif(0, 10); sdsp ~ dunif(0, 10)

  for (j in 1:nObs) {
    p.R[j] <- step(R[j] - 1) # Calcula la probabilidad de que RME sea mayor que 1
  }
}

# Ajustamos las distribuciones previas
inits.mod1 <- function() {list(m = rnorm(1),
                               sdhet = runif(1),
                               sdsp = runif(1))}

# Cargamos los datos
datos.mod1 <- list(O = aragon.spdf$O,
                   E = aragon.spdf$E,
                   adj = vecinos$adj,
                   num = vecinos$num,
                   w = vecinos$weights,
                   nObs = length(aragon.spdf))

# Pedimos los resultados
params.mod1 <- c("R", "p.R")

set.seed(123)
# Lanzamos el modelo
res.mod1 <- bugs(data = datos.mod1, model = mod1,
                 param = params.mod1, inits = inits.mod1)
```

Recuperamos el objeto sf para graficarlo mejor.

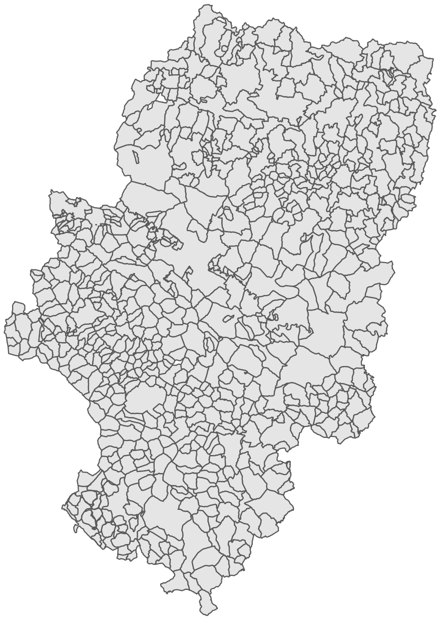
```
# Añadimos al data.frame las columnas RME y p.RME
aragon.sf$RME <- res.mod1$mean$R
aragon.sf$p.RME <- res.mod1$mean$p.R
```

```
rme.plot <- ggplot(aragon.sf) +
  geom_sf(aes(fill = RME)) +
  scale_fill_viridis_c(option = "rocket",direction = -1) +
  ggtitle("Aragón, RME") + xlab("Longitud") + ylab("Latitud") +
  theme_classic() +
  theme(axis.text = element_text(hjust = 1, angle = 30))

p.rme.plot <- ggplot(aragon.sf) +
  geom_sf(aes(fill = p.RME)) +
  scale_fill_viridis_c(option = "mako",direction=-1) +
  ggtitle("Aragón, P(RME>1)") + xlab("Longitud") + ylab("") +
  theme_classic() +
  labs(fill = "P(RME>1)") +
  theme(axis.text = element_text(hjust = 1, angle = 30))

wrap_plots(plotlist = list(rme.plot, p.rme.plot))
```



```
#'
### --- 0. Loading the packages --- ####
library(sf)
library(spdep)
library(lattice)
library(latticeExtra)
library(viridis)
library(gridExtra)
library(RColorBrewer)
library(INLA)
library(sf)
library(ggthemes)


#BiocManager::install(c("graph", "Rgraphviz"), dep=TRUE)
library(Rgraphviz)
library(graph)
library(leaflet)
library(ggplot2)
library(patchwork)

### --- 1. Loading the data --- ####
#Dataset
load("material/Aragon.Rdata") #datos
data <- Aragon.df
```

```r
Nareas <- length(data[,1]) #número de áreas

# sf
aragon_shape <- read_sf("material/aragon.shp")

aragon_shape <- st_set_crs(aragon_shape, 27700)

ggplot(aragon_shape) +
  geom_sf() +
  theme_void()
```

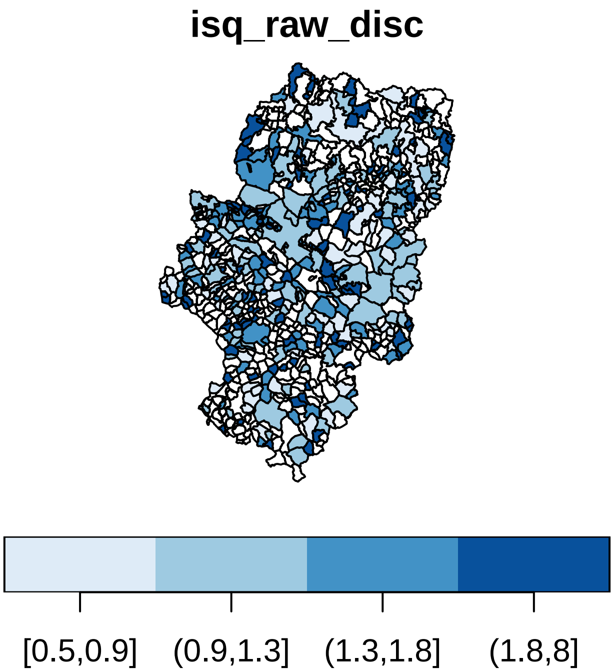


```r
### --- 2. Checking if the data of the sp and the data.frame match --- ####
# The order of the areas needs to be the same between
# the data and the spatial polygon object obtained importing
# the shapefile, so we re-order the data.
data$CODMUNI == aragon_shape$CODMUNI
```

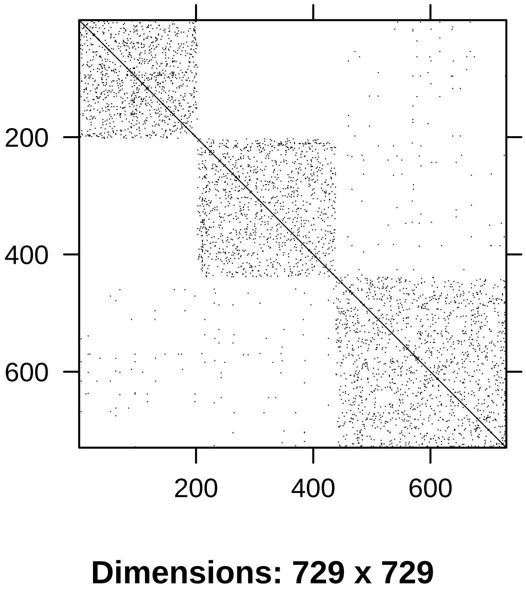
```
  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
 [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
[133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[457] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[469] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[481] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[493] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[505] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[517] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[529] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[541] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[553] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[565] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[577] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[589] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[601] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[613] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[637] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[673] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[685] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[697] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[709] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[721] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

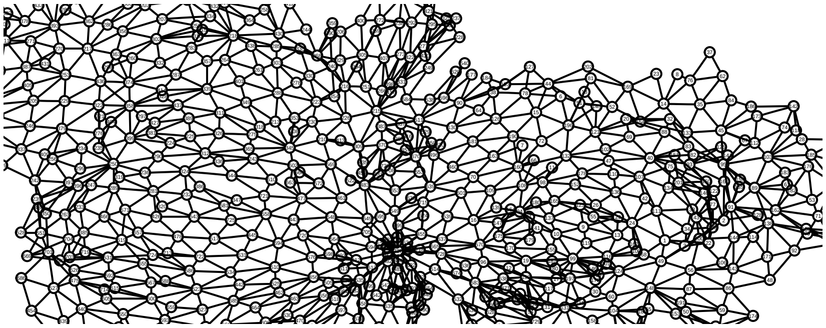
```
aragon_shape <- aragon_shape[ order(aragon_shape$CODMUNI), ]


### ----- 2.1. Plotting the data --- ####
### ------- 2.1.1. ISQ --- ####
aragon_shape$isq_raw <- data$O/data$E
isq_raw.cutoff<- c(0.5, 0.9, 1.3, 8,  1.8)
isq_raw_disc = cut(aragon_shape$isq_raw,
                   breaks         = isq_raw.cutoff,
                   include.lowest = TRUE)

aragon_shape$isq_raw_disc <- isq_raw_disc

plot(aragon_shape["isq_raw_disc"], max.plot = 1,
     pal =  brewer.pal(9,'Blues')[c(2,4,6,8)],
     key.pos = 1)
```



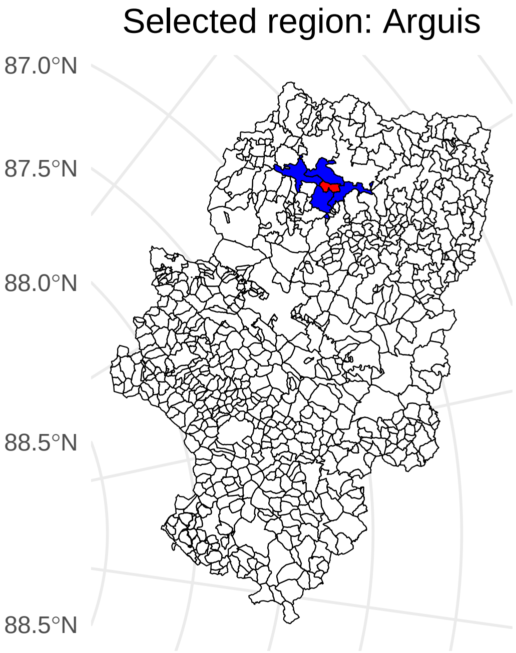
```
### --- 3. Defining neighbor relation --- ####
temp <- poly2nb(aragon_shape)

#This create a file called ``ARG.graph'' with the graph for INLA
nb2INLA("ARG.graph", temp)

### ----- 3.1. Plotting the generated graph --- ####
H <- inla.read.graph(filename="ARG.graph")
image(inla.graph2matrix(H),xlab="",ylab="") #identifica los vecinos de cada barrio, el barrio 1 tiene
```



```
### ----- 3.2. More plotting --- ####
plot(H)
```

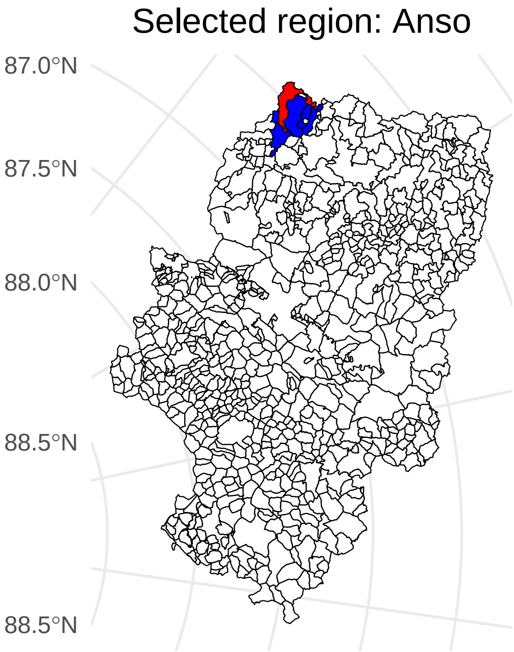


```
### ----- 3.3. Plotting the neighbors --- ####
plot_map_neig_ggplot <- function(neig, aragon_shape, temp) {
  # Base map of London
  p <- ggplot() +
    geom_sf(data = aragon_shape, fill = "white", color = "black") +

    # Highlight the selected region in red
    geom_sf(data = aragon_shape[neig, ], fill = "red", color = "black") +

    # Highlight the neighbors in blue
    geom_sf(data = aragon_shape[temp[[neig]], ], fill = "blue", color = "black") +

    # Set the theme
    theme_minimal() +
    ggtitle(paste("Selected region:", aragon_shape$NOMBRE[neig])) +
    theme(plot.title = element_text(hjust = 0.5))

  # Print the plot
  print(p)

  # Print information about the selected region and its neighbors
  cat("You have selected", aragon_shape$NOMBRE[neig], "and its neighbors are:", "\n")
  cat(aragon_shape$NOMBRE[temp[[neig]]], "\n")
}
#esta funcion identifica los vecinos de cada vecindario en concreto

plot_map_neig_ggplot(neig = 30, aragon_shape, temp = temp)
```



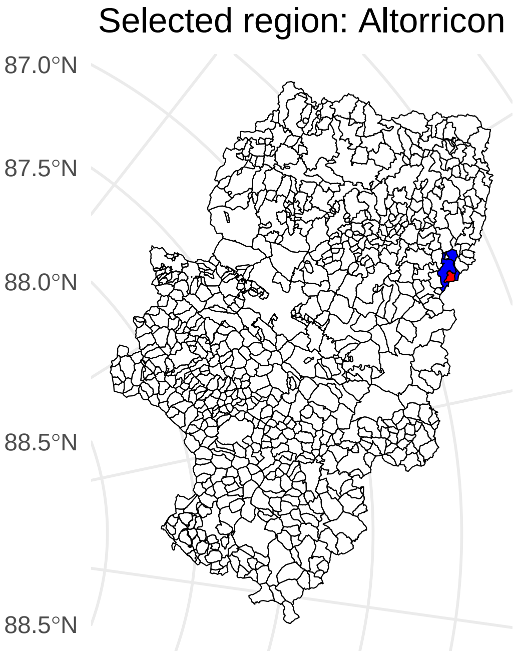
```
You have selected Arguis and its neighbors are:
Caldearenas Nueno Las Penas de Riglos La Sotonera
```

```
plot_map_neig_ggplot(neig = 25, aragon_shape, temp = temp)
```



```
You have selected Anso and its neighbors are:
Aisa Aragues del Puerto Canal de Berdun Fago Valle de Hecho
```

```
plot_map_neig_ggplot(neig = 23, aragon_shape, temp = temp)
```

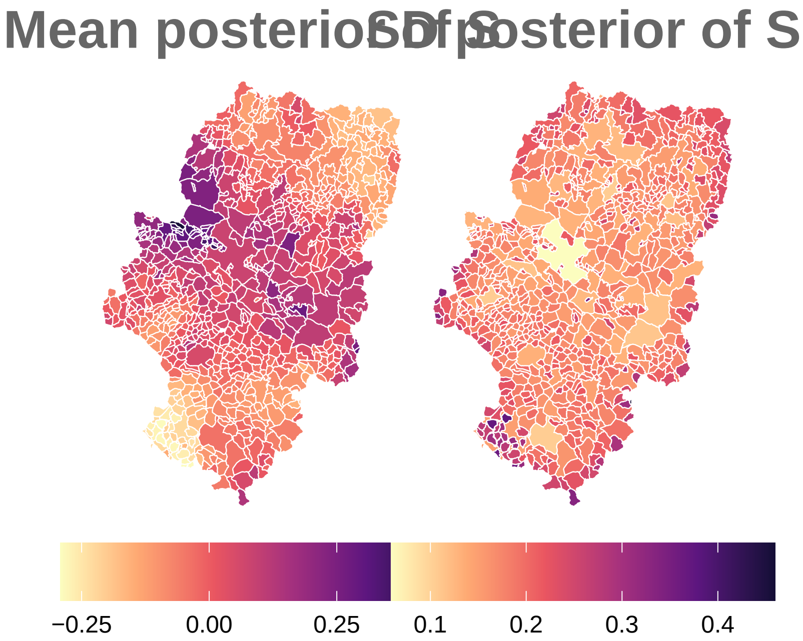


```
You have selected Altorricon and its neighbors are:
Alcampell Tamarite de Litera
```

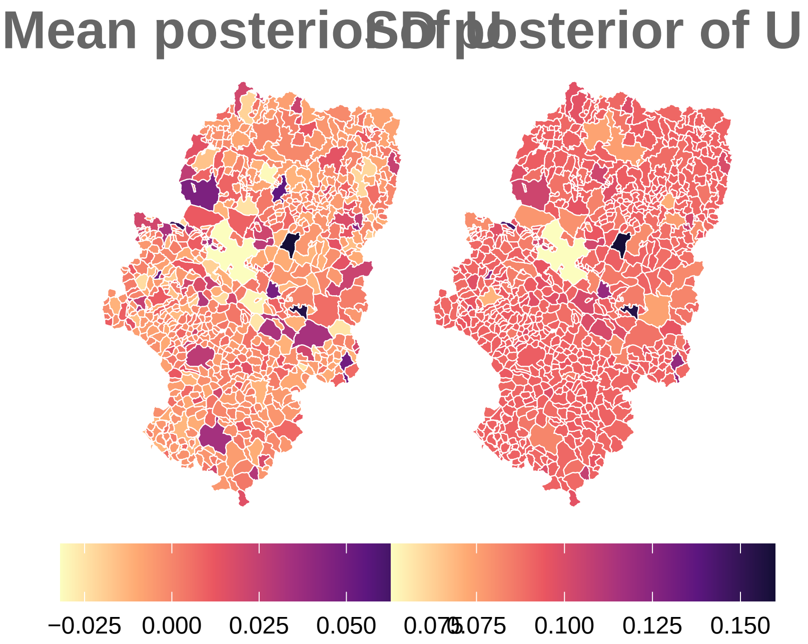
```r
### --- 4. Fitting a model with bym effect --- ####
### ----- 4.0. Adding ids for the random effects --- ####
S <- U <- seq(1,729) # crea dos veces el mismo índice, uno para el efecto aleatorio espacial y otro p
data <- cbind(data, S, U)


### ----- 4.1. Formula --- ####
formula <- O ~ 1 + f(S,
                     model       = "besag",
                     graph       = H,
                     scale.model = TRUE,
                     hyper       =
                       list(prec = list(prior="loggamma",param = c(1,0.001)))) +
  f(U,
    model       = "iid",
    hyper       =
      list(prec = list(prior="loggamma",param = c(1,0.001))))


### ----- 4.2. Model --- ####
set.seed(123)
mod.isq <- inla(formula,
                family          = "poisson",
                data            = data,
                E               = E, #es el offset, se puede poner como offset o como esperados
                control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE),
                control.predictor = list(compute=TRUE, cdf=c(log(1)))) #cdf = cumulative density func

summary(mod.isq)
```

```
Time used:
    Pre = 0.477, Running = 0.958, Post = 0.124, Total = 1.56
Fixed effects:
              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
(Intercept) -0.066 0.036     -0.139   -0.066      0.003 -0.066   0

Random effects:
  Name    Model
    S Besags ICAR model
   U IID model

Model hyperparameters:
                  mean     sd 0.025quant 0.5quant 0.975quant  mode
Precision for S  19.02  11.38       6.37    16.16      49.07 11.95
Precision for U 705.16 934.70      13.17   373.67    3199.56 18.41

Deviance Information Criterion (DIC) ...............: 1897.94
Deviance Information Criterion (DIC, saturated) ....: 825.54
Effective number of parameters .....................: 53.62

Watanabe-Akaike information criterion (WAIC) ...: 1902.53
Effective number of parameters .................: 52.37

Marginal log-Likelihood:  -1246.57
CPO, PIT is computed
Posterior summaries for the linear predictor and the fitted values are computed
(Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

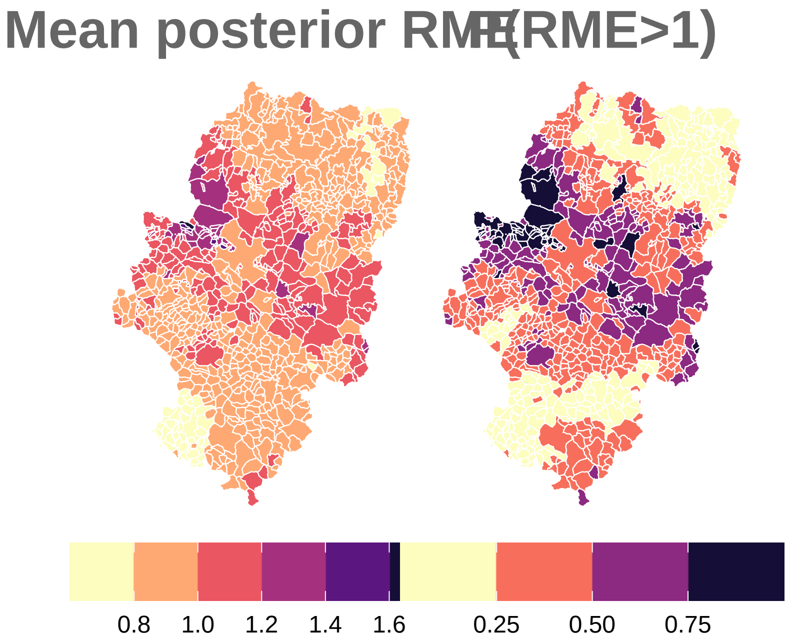
```r
### ----- 4.3. Posterior distribution of the random effects --- ####
aragon_shape$SPmean <- round(mod.isq$summary.random$S[["mean"]], 4)
aragon_shape$SPsd <- round(mod.isq$summary.random$S[["sd"]],5)

#Mean posterior distribution
a <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = SPmean), color = "white") +
  scale_fill_viridis_c(option = "magma",begin = 0.1, direction = -1) +
  theme_void() +
  theme(legend.position="bottom",
        plot.title = element_text(hjust = 0.5,
                                  color = "Gray40",
                                  size = 20,
                                  face = "bold"),
        legend.title = element_blank(),
        plot.subtitle = element_text(color = "blue"),
        plot.caption = element_text(color = "Gray60")) +
  guides(fill = guide_colorbar(barwidth = 10, barheight = 1.5)) +  # Adjusting the legend size
  ggtitle("Mean posterior of S")

#Sd posterior distribution
b <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = SPsd), color = "white") +
  scale_fill_viridis_c(option = "magma",begin = 0.1, direction = -1) +
  theme_void() +
  theme(legend.position="bottom",
        plot.title = element_text(hjust = 0.5,
                                  color = "Gray40",
                                  size = 20,
                                  face = "bold"),
        legend.title = element_blank(),
        plot.subtitle = element_text(color = "blue"),
        plot.caption = element_text(color = "Gray60")) +
  guides(fill = guide_colorbar(barwidth = 10, barheight = 1.5)) +  # Adjusting the legend size
  ggtitle("SD posterior of S")

a | b
```



```
aragon_shape$UPmean <- round(mod.isq$summary.random$U[["mean"]], 4)
aragon_shape$UPsd <- round(mod.isq$summary.random$U[["sd"]],5)

#Mean posterior distribution
c <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = UPmean), color = "white") +
  scale_fill_viridis_c(option = "magma",begin = 0.1, direction = -1) +
  theme_void() +
  theme(legend.position="bottom",
        plot.title = element_text(hjust = 0.5,
                                  color = "Gray40",
                                  size = 20,
                                  face = "bold"),
        legend.title = element_blank(),
        plot.subtitle = element_text(color = "blue"),
        plot.caption = element_text(color = "Gray60")) +
  guides(fill = guide_colorbar(barwidth = 10, barheight = 1.5)) +  # Adjusting the legend size
  ggtitle("Mean posterior of U")

#Sd posterior distribution
d <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = UPsd), color = "white") +
  scale_fill_viridis_c(option = "magma",begin = 0.1, direction = -1) +
  theme_void() +
  theme(legend.position="bottom",
        plot.title = element_text(hjust = 0.5,
                                  color = "Gray40",
                                  size = 20,
                                  face = "bold"),
        legend.title = element_blank(),
        plot.subtitle = element_text(color = "blue"),
        plot.caption = element_text(color = "Gray60")) +
  guides(fill = guide_colorbar(barwidth = 10, barheight = 1.5)) +  # Adjusting the legend size
  ggtitle("SD posterior of U")

c | d
```

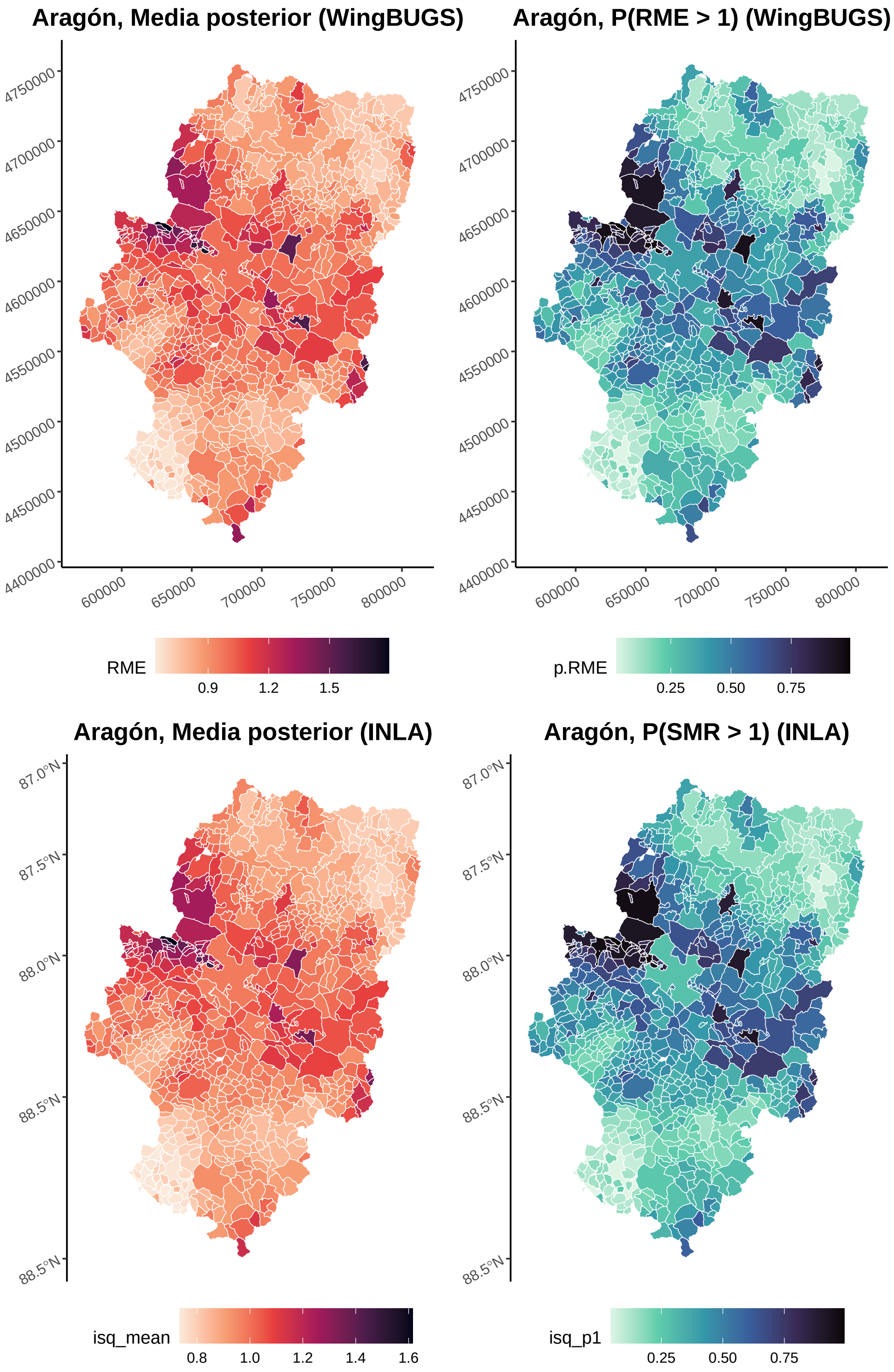


```
### --- 4.4. Posterior distribution of suicides mortality --- ####
aragon_shape$isq_mean <- mod.isq$summary.fitted.values$mean # mean
aragon_shape$isq_sd <- mod.isq$summary.fitted.values$sd #s
aragon_shape$isq_median <- mod.isq$summary.fitted.values$`0.5quant` # median
aragon_shape$isq_q025 <- mod.isq$summary.fitted.values$`0.025quant` # quantile
aragon_shape$isq_q975 <- mod.isq$summary.fitted.values$`0.975quant` # quantile
aragon_shape$isq_p1 <- 1 - mod.isq$summary.fitted.values$`1cdf` # probability to be greater than 1

### --- 4.5. Posterior distribution of suicides SMR with cutoff--- ####
## Also, the probability for SMR to be greater than 1.

isq_disc1 <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = isq_mean), color = "white") +
  scale_fill_viridis_b(option = "magma", begin = 0.1, direction = -1) +
  theme_void() +
  theme(
    legend.position = "bottom",
    plot.title = element_text(
      hjust = 0.5,
      color = "Gray40",
      size = 20,
      face = "bold"
    ),
    legend.title = element_blank(),
    plot.subtitle = element_text(color = "blue"),
    plot.caption = element_text(color = "Gray60")
  ) +
  guides(fill = guide_colourbar(barwidth = 10, barheight = 1.5)) +
  ggtitle("Mean posterior RME")


isq_disc2 <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = isq_p1), color = "white") +
  scale_fill_viridis_b(option = "magma", begin = 0.1, direction = -1) +
  theme_void() +
  theme(
    legend.position = "bottom",
    plot.title = element_text(
      hjust = 0.5,
      color = "Gray40",
      size = 20,
      face = "bold"
    ),
    legend.title = element_blank(),
    plot.subtitle = element_text(color = "blue"),
    plot.caption = element_text(color = "Gray60")
  ) +
  guides(fill = guide_colourbar(barwidth = 10, barheight = 1.5)) +
  ggtitle("P(RME>1)")


isq_disc1 | isq_disc2
```



```
library(ggplot2)
library(patchwork)

# Gráfico INLA: Media posterior
inla_mean_plot <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = isq_mean), color = "white") +
  scale_fill_viridis_c(option = "rocket",direction = -1) +
  ggtitle("Aragón, Media posterior (INLA)") +
  theme_classic() +
  theme(
    legend.position = "bottom",
    axis.text = element_text(hjust = 1, angle = 30),
    plot.title = element_text(hjust = 0.5, size = 15, face = "bold")
  ) +
  guides(fill = guide_colourbar(barwidth = 10, barheight = 1.5))

# Gráfico INLA: Probabilidad (SMR > 1)
inla_p_plot <- ggplot(data = aragon_shape) +
  geom_sf(aes(fill = isq_p1), color = "white") +
  scale_fill_viridis_c(option = "mako", direction=-1) +
  ggtitle("Aragón, P(SMR > 1) (INLA)") +
  theme_classic() +
  theme(
    legend.position = "bottom",
    axis.text = element_text(hjust = 1, angle = 30),
    plot.title = element_text(hjust = 0.5, size = 15, face = "bold")
  ) +
  guides(fill = guide_colourbar(barwidth = 10, barheight = 1.5))
```

```r
# Gráfico WingBUGS: Media (RME)
rme_plot <- ggplot(data = aragon.sf) +
  geom_sf(aes(fill = RME), color = "white") +
  scale_fill_viridis_c(option = "rocket",direction = -1) +
  ggtitle("Aragón, Media posterior (WingBUGS)") +
  theme_classic() +
  theme(
    legend.position = "bottom",
    axis.text = element_text(hjust = 1, angle = 30),
    plot.title = element_text(hjust = 0.5, size = 15, face = "bold")
  ) +
  guides(fill = guide_colourbar(barwidth = 10, barheight = 1.5))

# Gráfico WingBUGS: Probabilidad (RME > 1)
p_rme_plot <- ggplot(data = aragon.sf) +
  geom_sf(aes(fill = p.RME), color = "white") +
  scale_fill_viridis_c(option = "mako",direction = -1) +
  ggtitle("Aragón, P(RME > 1) (WingBUGS)") +
  theme_classic() +
  theme(
    legend.position = "bottom",
    axis.text = element_text(hjust = 1, angle = 30),
    plot.title = element_text(hjust = 0.5, size = 15, face = "bold")
  ) +
  guides(fill = guide_colourbar(barwidth = 10, barheight = 1.5))

# Organización de los gráficos en una cuadrícula 2x2
final_plot <- (rme_plot | p_rme_plot) /
              (inla_mean_plot | inla_p_plot)

# Mostrar los gráficos
final_plot
```



Aragón, Media posterior (WingBUGS)

Aragón, P(RME > 1) (WingBUGS)

Aragón, Media posterior (INLA)

Aragón, P(SMR > 1) (INLA)

```
library(dplyr)
library(ggplot2)
```

```r
library(gridExtra)

## 1. Resumen descriptivo
descriptive_stats <- aragon_shape %>%
  summarise(
    mean_INLA_RME = mean(isq_mean, na.rm = TRUE),
    sd_INLA_RME = sd(isq_mean, na.rm = TRUE),
    mean_WingBUGS_RME = mean(aragon.sf$RME, na.rm = TRUE),
    sd_WingBUGS_RME = sd(aragon.sf$RME, na.rm = TRUE),
    mean_INLA_p = mean(isq_p1, na.rm = TRUE),
    sd_INLA_p = sd(isq_p1, na.rm = TRUE),
    mean_WingBUGS_p = mean(aragon.sf$p.RME, na.rm = TRUE),
    sd_WingBUGS_p = sd(aragon.sf$p.RME, na.rm = TRUE)
  )
descriptive_stats
```

```
Simple feature collection with 1 feature and 8 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 569301 ymin: 4413136 xmax: 810739 ymax: 4755089
Projected CRS: OSGB36 / British National Grid
# A tibble: 1 x 9
  mean_INLA_RME sd_INLA_RME mean_WingBUGS_RME sd_WingBUGS_RME mean_INLA_p
          <dbl>       <dbl>             <dbl>           <dbl>       <dbl>
1         0.968       0.123             0.956           0.167       0.399
# i 4 more variables: sd_INLA_p <dbl>, mean_WingBUGS_p <dbl>,
#   sd_WingBUGS_p <dbl>, geometry <POLYGON [m]>
```

```r
## 2. Correlación
cor_RME <- cor(aragon.sf$RME, aragon_shape$isq_mean, use = "complete.obs")
cor_p <- cor(aragon.sf$p.RME, aragon_shape$isq_p1, use = "complete.obs")

cat("Correlación entre RME (WingBUGS e INLA):", cor_RME, "\n")
```

```
Correlación entre RME (WingBUGS e INLA): 0.9786599
```

```r
cat("Correlación entre P(RME>1) (WingBUGS e INLA):", cor_p, "\n")
```

```
Correlación entre P(RME>1) (WingBUGS e INLA): 0.9830765
```

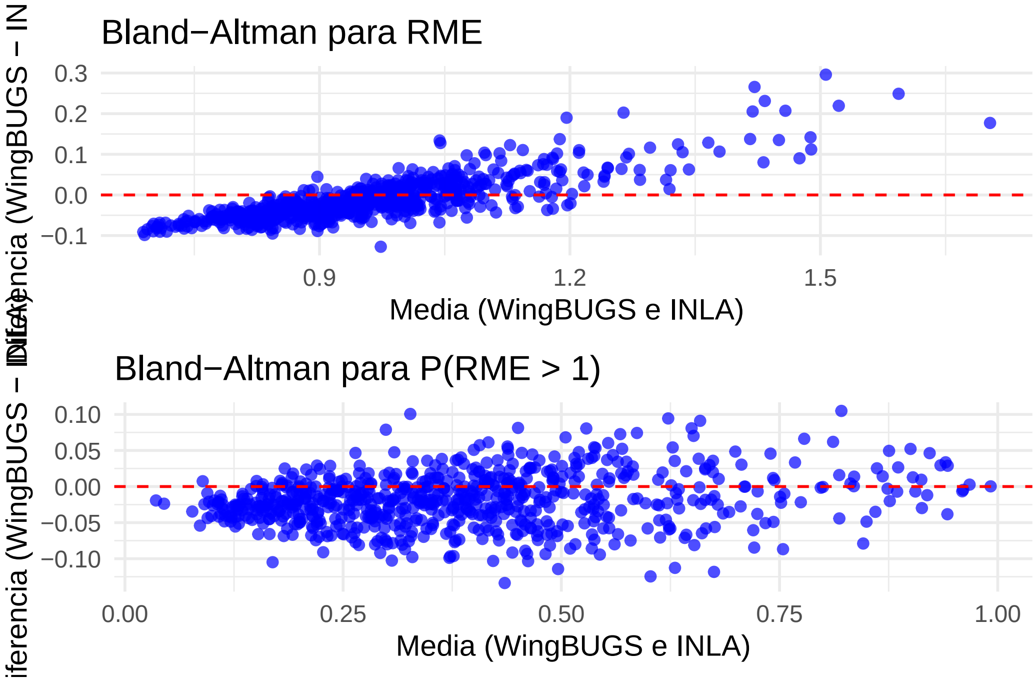
```r
## 3. Test de diferencias estadísticas
t_test_RME <- t.test(aragon.sf$RME, aragon_shape$isq_mean, paired = TRUE)
t_test_p <- t.test(aragon.sf$p.RME, aragon_shape$isq_p1, paired = TRUE)

t_test_RME
```

```
	Paired t-test

data:  aragon.sf$RME and aragon_shape$isq_mean
t = -6.0426, df = 728, p-value = 2.42e-09
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -0.015553115 -0.007925106
sample estimates:
mean difference
    -0.01173911
```

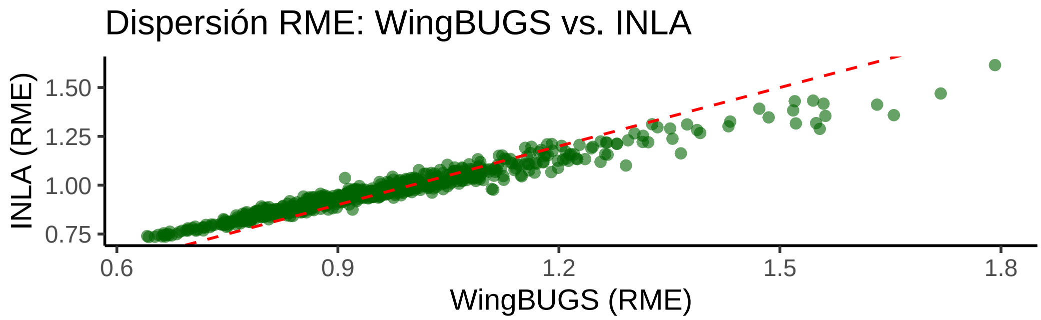
```
t_test_p
```

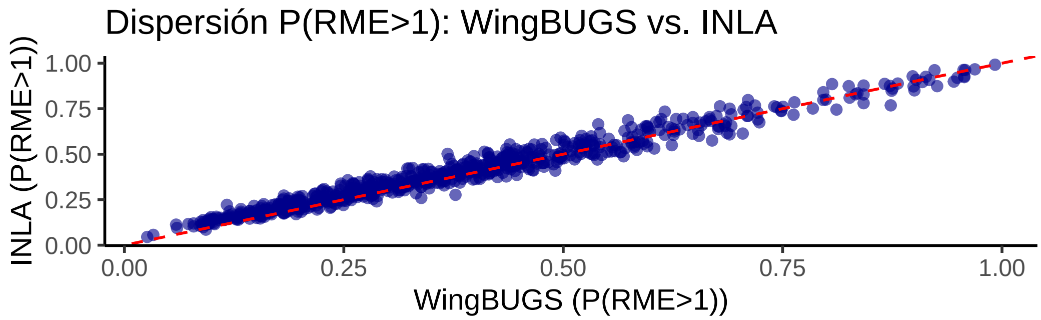
```
	Paired t-test

data:  aragon.sf$p.RME and aragon_shape$isq_p1
t = -14.01, df = 728, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -0.02152283 -0.01623227
sample estimates:
mean difference
    -0.01887755
```

```
## 4. Bland-Altman
bland_altman_data <- aragon.sf %>%
  mutate(
    mean_RME = (RME + aragon_shape$isq_mean) / 2,
    diff_RME = RME - aragon_shape$isq_mean,
    mean_p = (p.RME + aragon_shape$isq_p1) / 2,
    diff_p = p.RME - aragon_shape$isq_p1
  )

bland_RME_plot <- ggplot(bland_altman_data, aes(x = mean_RME, y = diff_RME)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  ggtitle("Bland-Altman para RME") +
  xlab("Media (WingBUGS e INLA)") + ylab("Diferencia (WingBUGS - INLA)") +
  theme_minimal()

bland_p_plot <- ggplot(bland_altman_data, aes(x = mean_p, y = diff_p)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  ggtitle("Bland-Altman para P(RME > 1)") +
  xlab("Media (WingBUGS e INLA)") + ylab("Diferencia (WingBUGS - INLA)") +
  theme_minimal()

grid.arrange(bland_RME_plot, bland_p_plot, nrow = 2)
```



```
## 5. Gráficos de dispersión
scatter_RME <- ggplot(data = NULL, aes(x = aragon.sf$RME, y = aragon_shape$isq_mean)) +
  geom_point(color = "darkgreen", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  ggtitle("Dispersión RME: WingBUGS vs. INLA") +
  xlab("WingBUGS (RME)") + ylab("INLA (RME)") +
  theme_classic()

scatter_p <- ggplot(data = NULL, aes(x = aragon.sf$p.RME, y = aragon_shape$isq_p1)) +
  geom_point(color = "darkblue", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  ggtitle("Dispersión P(RME>1): WingBUGS vs. INLA") +
  xlab("WingBUGS (P(RME>1))") + ylab("INLA (P(RME>1))") +
  theme_classic()

grid.arrange(scatter_RME, scatter_p, nrow = 2)
```


Dispersión RME: WingBUGS vs. INLA
INLA (RME)
1.50
1.25
1.00
0.75
0.6
0.9
1.2
1.5
1.8
WingBUGS (RME)


Dispersión P(RME>1): WingBUGS vs. INLA
INLA (P(RME>1))
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
WingBUGS (P(RME>1))