

**chikungunya-INLAspacetime-spde** / **Chik_SPDE_nbi_model.R** **Rpalasio** Update Chik_SPDE_nbi_model.R

fb55229 · now



805 lines (675 loc) · 33.4 KB

Code

Blame

Raw



```
1  ##### Spatio-temporal modeling of chikungunya in Brazil #####
2  # Author: Raquel Gardini Sanches Palasio
3  # Article: "Spatio-temporal Modeling of chikungunya in Brazil:
4  #           Interplay of Climate, Mobility, and Socioeconomic Factors"
5  # Description: Implementation of a DEMF(1,2,1) SPDE model using INLAspacetime
6  # R version 4.3.3 (2024-02-29) -- "Angel Food Cake"
7
8  # ---- 1. Load Packages -----
9  # Core libraries
10 library(tidyverse)
11 library(dplyr)
12 library(ggplot2)
13 library(sf)
14 library(cowplot)
15 library(gridExtra)
16
17 # Spatial and Bayesian modeling
18 library(INLA)          # version 24.3.29
19 library(INLAspacetime) # version 0.1.8
20 library(inlabru)       # version 2.10.1
21 library(fmesher)       # version 0.1.7
22 library(INLAOutputs)
23
24 # Mapping and shapefiles
25 library(geobr)          # for Brazilian municipalities and regions
26 library(ggspatial)      # to add north arrow and scale bar
27 library(ggsn)           # extra map tools
28
29
30 # ---- 2. Load and Prepare Data -----
31
32 # The dataset 'data' should contain chikungunya incidence and covariates
33 # Variables include: time (month), space (municipality), Tmin, mobility index, etc.
34 # Dataset and dictionary available at [**Figshare (link to be inserted)**]
35 # Please download the dataset from Figshare and place it in the data/ folder before
36 # running the script.
```

```

37
38 data <- read_delim("data/df_ChiK.csv", delim = ";", escape_double = FALSE, trim_ws = TRUE)
39
40 # ---- 2.1 Load Shapefile of Brazilian Municipalities -----
41
42 # Load 2022 municipal boundaries (from geobr)
43 MN_shp <- read_municipality(year = 2022, showProgress = FALSE)
44
45 # Adjust municipality code (remove last digit to match your dataset ID)
46 MN_shp$COD_MUN6 <- substr(MN_shp$code_muni, 1, nchar(MN_shp$code_muni) - 1)
47
48 # Remove Fernando de Noronha (not included in study area)
49 MN_shp <- MN_shp %>% filter(code_muni != 2605459)
50
51 # ---- 2.2 Prepare Spatial Data Frame -----
52
53 # Convert shapefile to dataframe (drop geometry)
54 BR.df <- MN_shp %>% st_drop_geometry()
55 N <- nrow(BR.df)
56
57 # Order by municipality code
58 BR.df <- BR.df[order(BR.df$COD_MUN6, decreasing = FALSE), ]
59
60 # Create spatial IDs for model indexing
61 BR.df$ID.space.1 <- seq(1, N)
62 shp.df <- as.data.frame(cbind(my.ID=1:N, ID.space.1=BR.df$ID.space.1, ID_MN_RESI=BR.df$COD
63 shp.df$ID_MN_RESI <- as.numeric(shp.df$ID_MN_RESI)
64
65 # ---- 2.3 Merge Spatial Identifiers with Main Data -----
66
67 # Add municipality and time identifiers to dataset
68 df <- left_join(data, shp.df[, c("my.ID", "ID_MN_RESI")], by = "ID_MN_RESI")
69
70 # Order dataset by time and municipality to match shapefile order
71 df <- arrange(df, ID.time.1, my.ID)
72
73 # ---- 2.4 Coordinate Transformation -----
74
75 # Convert to sf object using latitude and longitude
76 df <- st_as_sf(df, coords = c("LONG", "LAT"), crs = 4326, remove = FALSE)
77
78 # Define Albers Equal-Area projection for South America (ESRI:102033)
79 albers_km <- "+proj=aea +lat_0=-32 +lon_0=-60 +lat_1=-5 +lat_2=-42 +x_0=0 +y_0=0 +ellps=au
80
81 # Transform coordinates to projected CRS (in kilometers)
82 df <- st_transform(df, crs = albers_km)
83
84 # Extract UTM coordinates (for mesh creation)
85 df$xUTM <- st_coordinates(df)[, 1]
86 df$yUTM <- st_coordinates(df)[, 2]
87
88

```

```

89 # ---- 2.5 Prepare Coordinates for Mesh -----
90
91 # Create a dataframe of unique coordinates for mesh construction
92 coords.cv <- data.frame(unique(st_coordinates(df)))
93
94
95 # ---- 3. Mesh Construction -----
96 # Create SPDE mesh using the actual study border
97
98 # Load border shapefile (projected CRS: Albers South America)
99 border <- read_sf("data/borda_102033.shp")
100
101 # Transform CRS to match UTM coordinates of centroids
102 border <- st_transform(border, crs = albers_km)
103
104 # Remove unnecessary columns (e.g., FID)
105 border <- border[, !(names(border) %in% c("FID"))]
106
107 # Create a spatial mesh with small scale
108 smesh <- inla.mesh.2d(
109   loc = cbind(coords.cv[, 1], coords.cv[, 2]), # spatial points
110   boundary = st_as_sfc(border), # study area boundary
111   max.edge = c(10, 350),
112   offset = c(100, 350),
113   cutoff = 35
114 )
115
116 # Check mesh summary and manifold structure
117 print(fm_manifold(smesh, c("S", "R")))
118
119 # Number of mesh vertices
120 smesh$n
121
122 # Plot mesh with border and data points
123 mesh_plot <- ggplot() +
124   gg(smesh) +
125   geom_point(
126     data = data.frame(coords.cv),
127     aes(x = X, y = Y),
128     size = 0.2,
129     colour = "red"
130   ) +
131   theme_minimal()
132
133 # Save figure (high-resolution)
134 ggsave(
135   filename = "FigS9_mesh.jpeg",
136   plot = mesh_plot,
137   dpi = 1000,
138   width = 8,
139   height = 6,
140   units = "in"

```

```

141 )
142
143 # ---- 4. Prepare Climate and Covariates for Modeling -----
144
145 # ---- 4.1 Create Non-linear Climate Groups -----
146
147 # Group continuous climate variables using quantiles
148 df$Tmin.grp <- inla.group(df$Tmin, n = 25, method = "quantile") # minimum temperat
149 df$Prec.grp <- inla.group(df$Prec / 1000, n = 25, method = "quantile") # precipitation (m
150 df$Rtemp.grp <- inla.group(df$Rtemp, n = 25, method = "quantile") # relative tempera
151
152 # Note: Prec.grp and Rtemp.grp tested but not used in final model
153
154 # ---- 4.2 Select Covariates -----
155
156 xnames <- c(
157   "sewerage", "piped", "Tmin", "Rtemp", "Forest", "NDVI",
158   "Infra", "logKatz", "Tmin.grp", "Prec.grp", "Rtemp.grp"
159 )
160
161 # ---- 4.3 Create Modeling Data Frame -----
162
163 dataf <- data.frame(
164   df[c(
165     "id_key_time", "ID_MN_RESI", "xUTM", "yUTM",
166     "ID.time.1", "ID.space.time", "ID.space.1",
167     xnames
168   )],
169   pop10 = df$pop / 10000, # population in 10k units
170   Prec = df$Prec / 1000, # precipitation in meters
171   ALT.fc = factor(df$ALT.fc, levels = c(1, 2, 3, 4)),
172   BIO.fc = factor(df$BIO.fc, levels = c(1, 2, 3, 4, 5, 6)),
173   Enso.fc2 = factor(df$Enso.fc2, levels = c(1, 2, 3, 4, 5, 6)),
174   Reg.fc = factor(df$Reg.fc, levels = c(1, 2, 3, 4, 5)),
175   sewerage.fc = factor(df$sewerage.fc, levels = c(1, 2, 3)),
176   piped.fc = factor(df$piped.fc, levels = c(1, 2, 3)),
177   y = as.numeric(df$cases.chik)
178 )
179
180 # Remove geometry column if present
181 dataf <- dataf[, !(names(dataf) %in% c("geometry"))]
182
183 # Quick summary including missing values
184 summary(dataf, useNA = "always")
185
186 # ---- 5. Define Space-Time Model -----
187
188 # ---- 5.1 Temporal Mesh -----
189
190
191 nt <- max(dataf$ID.time.1) # total number of time points (10 year * 4 season)
192 h <- 1 # temporal resolution (1 season)

```

```

193
194 # Create 1D temporal mesh
195 tmesh <- inla.mesh.1d(
196   loc      = seq(1, nt + h/2, h), # sequence of time points
197   degree = 1                      # linear basis
198 )
199
200 # Check number of nodes in temporal mesh
201 tmesh$N
202
203
204 # ---- 5.2 Space-Time Model Definition -----
205
206 stmodel <- stModel.define(
207   smesh = smesh,          # spatial mesh (from section 3)
208   tmesh = tmesh,          # temporal mesh
209   model = '121',          # model type (see reference paper, Lindgren et al. 2024)
210   control.priors = list(
211     prs      = c(5, 0.1), # P(spatial range < 1) = 0.1
212     prt      = c(2, 0),   # temporal range fixed to 5
213     psigma   = c(1, 0.1)  # P(sigma > 1) = 0.1
214   )
215 )
216
217 # ---- 5.3 Define Model Formula -----
218
219 # Model formula: includes fixed effects, nonlinear effects, categorical factors,
220 # offset for population, and spatiotemporal random field
221 M <- ~ -1 + Intercept(1) + myoffset(log(pop10), model = "offset") +
222   sew(sewerage.fc, model = "factor_contrast") + # Socioeconomic: sewerage
223   pip(piped.fc, model = "factor_contrast") +    # Socioeconomic: piped water
224   Tmin.grp(Tmin.grp, model = 'rw2') +          # Climate: min temperature (nonlinear)
225   Prec +                                       # Climate: precipitation
226   logKatz +                                   # Mobility: Katz index (Log)
227   Forest +                                    # Environmental: forest cover
228   Alt(ALT.fc, model = "factor_contrast") +     # Elevation factor
229   Biome(BIO.fc, model = "factor_contrast") +   # Biome categorical factor
230   field(list(
231     space = cbind(xUTM, yUTM),                # Spatial coordinates
232     time  = ID.time.1),                        # Temporal index
233     model = stmodel)                          # Spatiotemporal model defined earlier
234
235
236 # ---- 6. Define Likelihood -----
237 lhood <- like(
238   formula = y ~ .,          # Response variable is count of cases (y)
239   family  = "nbinomial",    # Negative binomial model for overdispersed counts
240   data    = dataf
241 )
242
243 # ---- 7. Define Computation Options -----
244 ctrc <- list(

```

```

245     waic = TRUE, # Compute WAIC
246     dic = TRUE,  # Compute DIC
247     cpo = TRUE   # Compute CPO for model checking
248 )
249
250 # ---- 8. Fit the Model -----
251 fit121 <- bru(
252     M,          # Model formula
253     lhood,      # Likelihood
254     options = list(
255         control.compute = ctrc, # Specify computation options
256         verbose = FALSE        # Suppress detailed output during fitting
257     )
258 )
259
260 #Full model (not uploaded due to file size)
261 #save(fit121, file = "data/Results_models_chik.Rdata")
262
263 #To make it accessible on GitHub, the model was split and reduced into two files:
264 load("data/fit121_reduced.RData")
265 load("data/fit121_reduced_map.RData")
266
267 fit121 <- fit121_reduced
268 fit121$summary.fitted.values <- fit121_reduced2$summary.fitted.values
269
270 # ---- 9. Summarize Model Results -----
271 # Summary of posterior marginal distributions for the fixed effects
272 summary(fit121)
273
274 # Compute model fit statistics: WAIC, DIC, and mean CPO
275 round(c(fit121$waic$waic, fit121$dic$dic, -mean(log(fit121$cpo$cpo), na.rm=T)), 3)
276
277 # Posterior Distributions of Hyperparameters
278 # Avoid scientific notation for easier reading
279 options(scipen = 999)
280
281 # Posterior summaries for hyperparameters (e.g., spatial/temporal variance)
282 round(fit121$summary.hyperpar, 2)
283
284 # Posterior Summaries of Fixed Effects
285 # Columns: mean, sd, 0.025 quantile, 0.975 quantile
286 round(cbind(fit121$summary.fixed[, c(1, 2, 3, 5)]), 3)
287
288 # ---- 10. Prepare Table for Plot -----
289 # Fixed effects table
290 a <- data.frame(FixedEffects(fit121), fix.empty.names = FALSE)
291
292 var <- data.frame(
293     ID = 1:4,
294     mean = a$mean,
295     X0.025quant = a$qt0.025,
296     X0.975quant = a$qt0.975,

```

```

297     type = c("-", "Climate", "Mobility", "Environmental"),
298     Covariates = c("Intercept", "Precipitation total (m)", "Katz index", "Forest")
299 ) %>% rename("0.025quant" = X0.025quant, "0.975quant" = X0.975quant)
300
301
302 # Elevation
303 Elevation <- cbind(
304     fit121$summary.random$Alt$ID,
305     round(exp(fit121$summary.random$Alt[, c("mean", "0.025quant", "0.975quant")]), 2)
306 ) %>% rename( 'ID' = `fit121$summary.random$Alt$ID` )
307 linha <- data.frame(ID=1, mean=1, `0.025quant`=1, `0.975quant`=1)
308 linha <-rename (linha, `0.025quant` = X0.025quant , `0.975quant` = X0.975quant)
309
310 Elevation <- rbind(linha, Elevation)
311 Elevation$type <- c("Ref.", "Elevation", "Elevation", "Elevation")
312 Elevation$Covariates <- c("900-1601 (Ref.)", "600-899 (m)", "250-599 (m)", "3-249 (m)")
313
314 # Sewerage
315 sew <- cbind(fit121$summary.random$sew$ID,
316     round(exp(fit121$summary.random$sew [,c( "mean","0.025quant","0.975quant")]),2))
317 sew <-rename (sew, 'ID' = `fit121$summary.random$sew$ID` )
318 sew <- rbind(linha, sew)
319
320 sew$type <- c("Ref.", 'Socioeconomic', 'Socioeconomic')
321 sew$Covariates <- c ("sewerage <=0.3 (Ref.)", "sewerage>0.3 & <=0.6", "sewerage >0.6")
322
323 # Piped water
324 pip <- cbind(fit121$summary.random$pip$ID,
325     round(exp(fit121$summary.random$pip [,c( "mean","0.025quant","0.975quant")]),2))
326 pip <-rename (pip, 'ID' = `fit121$summary.random$pip$ID` )
327 pip <- rbind(linha, pip)
328 pip$type <- c("Ref.", 'Socioeconomic', 'Socioeconomic')
329 pip$Covariates <- c ("piped <=0.3 (Ref.)", "piped >0.3 & <=0.6", "piped >0.6")
330
331 # Biome
332 Biome <- cbind(fit121$summary.random$Biome$ID,
333     round(exp(fit121$summary.random$Biome [,c( "mean","0.025quant","0.975quant")]),2))
334 Biome <-rename (Biome, 'ID' = `fit121$summary.random$Biome$ID` )
335 Biome <- rbind(linha, Biome)
336
337 Biome$type <- c("Ref.", 'Biome', 'Biome', 'Biome', 'Biome', 'Biome')
338 Biome$Covariates <- c ("Mata Atlantica (Ref.)", "Pantanal", "Pampa", "Amazonia", "Cerrado",
339
340 # Combine all tables
341 tab <- rbind(pip, sew, Elevation, Biome, var)
342 tab <-rename (tab, ll = '0.025quant' )
343 tab <-rename (tab, ul = '0.975quant' )
344
345 tab$type <- factor(tab$type,
346     levels = c("-", "Socioeconomic", "Mobility", "Environmental", "Climate", "Elevati
347 tab <- tab %>% arrange(type)
348

```

```

349 tab$Covariates <- factor(tab$Covariates, levels = rev(c(
350   "Intercept",
351   "sewerage <=0.3 (Ref.)",
352   "sewerage>0.3 & <=0.6",
353   "sewerage >0.6",
354   "piped <=0.3 (Ref.)",
355   "piped >0.3 & <=0.6",
356   "piped >0.6",
357   "Katz index",
358   "Forest",
359   "Precipitation total (m)",
360   "900-1601 (Ref.)",
361   "600-899 (m)",
362   "250-599 (m)",
363   "3-249 (m)",
364   "Mata Atlantica (Ref.)",
365   "Pantanal",
366   "Pampa",
367   "Amazonia",
368   "Cerrado",
369   "Caatinga"
370 )))
371
372 tab <- tab %>% filter(Covariates != "Intercept")
373
374 # ---- 11. Plot IRR with 95% CI -----
375
376 Figplot <- ggplot(tab, aes(x = mean, y = Covariates, color = type, shape = type)) +
377   geom_point(size = 3) +
378   geom_errorbarh(aes(xmin = ll, xmax = ul), height = 0.2) +
379   geom_vline(xintercept = 1, linetype = "dashed", color = "gray") +
380   labs(
381     x = "IRR (95% CI)",
382     y = "Covariates") +
383   scale_color_manual( "",
384     values = c( "orange", "black","green3","blue","red", "brown", "grey")
385     ) +
386   scale_shape_manual( "",
387     values = c(16,16,16, 16, 16,16,17)) +
388   theme_minimal() +
389   theme(
390     panel.grid.major = element_blank(),
391     panel.grid.minor = element_blank(),
392     axis.line.x = element_line(color = "black"),
393     axis.line.y = element_line(color = "black"),
394     axis.text.y = element_text(size = 8, color = "black"),
395     legend.position = 'bottom',
396     legend.box = "horizontal",
397     legend.spacing.y = unit(0.0, "cm"),
398     legend.spacing.x = unit(0.5, "cm"),
399     legend.title = element_blank(),
400     legend.text = element_text(size = 10),

```

```

401     legend.key.width = unit(0.5, "cm"),
402     legend.margin = margin(0, 0, 0, 0)
403 ) +
404   guides(
405     color = guide_legend(nrow = 1),
406     shape = guide_legend(nrow = 1)
407   )
408
409 # Extract legend separately
410 legend = cowplot::get_plot_component(Figplot, 'guide-box-bottom', return_all = TRUE)
411 cowplot::ggdraw(legend)
412
413 # Plot without legend
414 Figplot <- ggplot(tab, aes(x = mean, y = Covariates, color = type, shape = type)) +
415   geom_point(size = 3) + # Define point size
416   geom_errorbarh(aes(xmin = ll, xmax = ul), height = 0.2) + # Horizontal error bars
417   geom_vline(xintercept = 1, linetype = "dashed", color = "gray") + # Reference line at R
418   labs(
419     x = "RR (95% CI)",
420     y = "Covariates"
421   ) +
422   scale_color_manual(
423     "", # Remove legend title
424     values = c("orange", "black", "green3", "blue", "red", "brown", "grey") # Manual colo
425   ) +
426   scale_shape_manual(
427     "",
428     values = c(16, 16, 16, 16, 16, 16, 17) # Manual shapes
429   ) +
430   theme_minimal() +
431   theme(
432     panel.grid.major = element_blank(), # Remove major grid lines
433     panel.grid.minor = element_blank(), # Remove minor grid lines
434     axis.line.x = element_line(color = "black"), # X-axis line
435     axis.line.y = element_line(color = "black"), # Y-axis line
436     axis.text.y = element_text(size = 8, color = "black"),
437     legend.position = "" # Remove legend
438   )
439
440 # ---- 12. Plot Tmin effect -----
441 data <- fit121$summary.random$Tmin.grp[,c( "ID", "mean", "0.025quant", "0.975quant")]
442
443 # Extract Tmin random effect summary
444 Tmin_max <- fit121$summary.random$Tmin.grp %>% filter(mean==max(mean))
445 Tmin_max_value <- round(Tmin_max$ID, 0)
446
447 FigTmin <- ggplot(data, aes(x = ID)) +
448   geom_ribbon(aes(ymin = exp(`0.025quant`), ymax = exp(`0.975quant`)), fill = "lightblue",
449   geom_line(aes(y = exp(`0.025quant`)), color = "blue", linetype = "dashed") +
450   geom_line(aes(y = exp(`0.975quant`)), color = "blue", linetype = "dashed") +
451   geom_line(aes(y = exp(mean)), color = "blue", linetype = "solid", size = 0.7) +
452   geom_hline(yintercept = 1, color = "grey", linetype = "dashed") +

```

```

453 geom_vline(xintercept = Tmin_max_value, color = "red", linetype = "dashed") +
454 scale_x_continuous(
455   breaks = seq(9, 25, by = 2),
456   labels = seq(9, 25, by = 2) ) +
457 labs(
458   x = "Minimum temperature (°Celsius)",
459   y = "RR (95% CI)"
460 ) +
461 theme_classic() +
462 theme(
463   axis.text = element_text(size = 9),
464   axis.title = element_text(size = 9),
465   axis.line.x = element_line(color = "black"), # Adiciona linha no eixo X
466   axis.line.y = element_line(color = "black"), # Adiciona linha no eixo Y
467   panel.grid.major = element_blank(), # Remove linhas de grade principais
468   panel.grid.minor = element_blank(), # Remove linhas de grade secundárias
469 )
470
471
472 # Combine plots with layout
473 layout.matrix <- matrix(c(1, 3, 2, 3), nrow = 2, ncol = 2)
474 layout(mat = layout.matrix,
475   heights = c(4, 1), # Heights of the two rows
476   widths = c(1.5, 2)) # Widths of the two columns
477
478 # Add label "A" to Figplot
479 Figplot2 <- Figplot +
480   annotate("text", x = 1.8, y = 18.5, label = "A",
481     hjust = 0, vjust = 0, size = 6, fontface = "bold", color = "black")
482 # Add label "B" to FigTmin
483 FigTmin2 <- FigTmin +
484   annotate("text", x = max(data$ID), y = max(exp(data$mean)), label = "B",
485     hjust = 1, vjust = -1.3, size = 6, fontface = "bold", color = "black")
486 # Arrange plots and legend using gridExtra
487 layout <- grid.arrange(
488   Figplot2, FigTmin2, legend,
489   widths = c(1.5, 2), heights = c(13, 1),
490   layout_matrix = rbind(c(1, 2),
491     c(3, 3))
492 )
493
494 # Save combined figure as PNG
495 ggsave("Fig5_grafico_RR_CI.png", layout, width = 19, height = 11, unit="cm", dpi = 1000)
496
497 # ---- 13. Prepare dataset for mapping -----
498 rnames <- c( paste0(
499   "summary.", c("fixed", "hyperpar", "random", "fitted.values")
500 ), "internal.marginals.hyperpar", "mlik", "mode", "cpu.used")
501
502 ndata <- nrow(dataf)
503 sres <- fit121[rnames]
504 sres$summary.fitted.values <- fit121$summary.fitted.values[1:ndata, c(1, 2)]

```

```

505
506 # Compute posterior mean incidence per 10,000 population
507 dataf$pred_mean = fit121$summary.fitted.values[1:ndata, "mean"] / dataf$pop10
508 # Compute lower limit of 95% credible interval
509 dataf$pred_ll = fit121$summary.fitted.values[1:ndata, "0.025quant"] / dataf$pop10
510 # Compute upper limits of the 95% credible interval
511 dataf$pred_ul = fit121$summary.fitted.values[1:ndata, "0.975quant"] / dataf$pop10
512
513 # Export results table for QGIS mapping (Figure 6 in the paper)
514 #write.table(dataf, "Result_model.csv", sep=";", dec = ".", row.names = FALSE, na="")
515
516 # ---- 14. Prepare region shapefile -----
517 # Load Brazilian regions (year 2020)
518 reg <- read_region(year = 2020, showProgress = FALSE)
519
520 # Convert multipolygons to individual polygons
521 reg_single <- reg %>%
522   st_cast("POLYGON") %>%
523   mutate(area = st_area(st_geometry(.)))
524
525 # Keep only the largest polygon for each region
526 reg_main <- reg_single %>%
527   group_by(code_region) %>% # change 'code_region' to 'code_muni' if working with munic
528   slice_max(order_by = area, n = 1) %>%
529   ungroup() %>%
530   select(-area)
531
532 # ---- 15. Map: Posterior Mean of Predicted Cases -----
533 dw <- dataf %>% select (id_key_time, ID_MN_RESI, pred_mean) %>%
534   reshape(timevar = "id_key_time",
535           idvar = "ID_MN_RESI", direction = "wide")
536
537 map <- merge(MN_shp[,c("COD_MUN6", "name_muni", "abbrev_state", "name_region")], dw, by.x
538 map_sf <- st_as_sf(map)
539
540 map_sf <- gather(map_sf, id_key_time, pred_mean, paste0("pred_mean.", 1:4, "_", rep(2015:2
541 map_sf$id <- gsub("pred_mean.", "", map_sf$id_key_time)
542 map_sf <- map_sf %>% separate(id, c("season", "year"), sep="_")
543
544 map_sf$season [map_sf$season == "1"] <- "Summer"
545 map_sf$season [map_sf$season == "2"] <- "Autumn"
546 map_sf$season [map_sf$season == "3"] <- "Winter"
547 map_sf$season [map_sf$season == "4"] <- "Spring"
548 map_sf$season <- factor(map_sf$season, levels = c("Summer", "Autumn", "Winter", "Spring"))
549
550 map_sf$id <- paste(map_sf$season, map_sf$year)
551 map_sf$id <- factor(map_sf$id,
552                   levels = c("Summer 2015", "Summer 2016", "Summer 2017", "Summer 2018", "S
553                             "Autumn 2015", "Autumn 2016", "Autumn 2017", "Autumn 2018", "A
554                             "Winter 2015", "Winter 2016", "Winter 2017", "Winter 2018", "W
555                             "Spring 2015", "Spring 2016", "Spring 2017", "Spring 2018", "S
556
557 ))

```

```

557
558 # Categorize predicted mean values by quantiles
559 map_sf$pred_mean <- as.numeric(map_sf$pred_mean)
560 quantile(map_sf$pred_mean)
561 summary(map_sf$pred_mean)
562 map_sf$quantil <- NA
563
564 map_sf$quantil[map_sf$pred_mean > 0 & map_sf$pred_mean < 0.2] <- "0.01-0.19"
565 map_sf$quantil[map_sf$pred_mean >= 0.2 & map_sf$pred_mean < 1] <- "0.2-0.9"
566 map_sf$quantil[map_sf$pred_mean >= 1 & map_sf$pred_mean < 3] <- "1.0-2.9"
567 map_sf$quantil[map_sf$pred_mean >= 3 & map_sf$pred_mean < 20] <- "3.0-19.9"
568 map_sf$quantil[map_sf$pred_mean >= 20] <- ">20"
569
570 map_sf$quantil <- factor(map_sf$quantil,
571                          levels = c("0.01-0.19", "0.2-0.9", "1.0-2.9", "3.0-19.9", ">20"))
572 table(map_sf$quantil)
573
574 # Create ggplot map
575 ggp <- ggplot() +
576   geom_sf(data = map_sf, aes(fill = quantil), color = "transparent") +
577   geom_sf(data = reg_main, fill = "transparent", aes(color = "Brazil's regions"), linewidth
578   theme_classic() +
579   scale_fill_manual(name = "", values= c("#0277BD", '#abdda4', '#ffffbf', '#fdae61', "red"))
580   limits = c("0.01-0.19", "0.2-0.9", "1.0-2.9", "3.0-19.9", ">20")) +
581   scale_color_manual( name = "", values = c("Brazil's regions" = "black") ) +
582   guides(
583     fill = guide_legend(order = 1,
584       label.theme = element_text(size = 8),# Tamanho do texto menor
585       keywidth = unit(0.3, "cm"),# Tamanho do ícone menor
586       keyheight = unit(0.3, "cm"),# Tamanho do ícone menor
587       nrow = 1),# Divide a legenda em duas linhas
588     color = guide_legend( order = 2,
589       override.aes = list(fill = NA, linetype = 1, size = 0.1),
590       label.theme = element_text(size = 8),
591       keywidth = unit(0.28, "cm"),
592       keyheight = unit(0.28, "cm"),
593       nrow = 1)) +
594   theme(legend.text = element_text(size = 10, face = "bold"),
595         legend.direction = "horizontal",
596         legend.position = "top", #"bottom",
597         legend.box = "horizontal", # ensures one block
598         legend.margin = margin(0, 0, 0, 0),
599         text = element_text(size = 10),
600         axis.line = element_blank(), # Remove as linhas dos eixos
601         axis.text = element_blank(), # Remove os rótulos dos eixos
602         axis.ticks = element_blank(),
603         strip.text = element_text(size = 4), #color = "dark green"),
604         panel.spacing = unit(0.01, "lines"),# Change spacing between facets on both axis
605         panel.spacing.x = unit(0.01, "lines"),# Change horizontal spacing between facets
606         panel.spacing.y = unit(0.01, "lines")) + # Change vertical spacing between facets
607   facet_wrap(~ id, ncol = 10 ) +
608   labs(title = "",

```

```

609         x = "",
610         y = "") +
611     blank() +
612     scalebar(map_sf, dist = 1000, dist_unit = "km", st.size = 1.1,
613             height=0.08, border.size = 0.1,
614             st.dist = 0.1, # Ajuste a distância vertical do texto da escala
615             anchor = c(x = -35, y = -38),
616             transform = TRUE, model = "WGS84",
617             facet.var = c("id"), #faz ficar soh no ultimo
618             facet.lev = c("Spring 2024"))
619
620 # Save map as JPEG
621 jpeg("FigS6_map_pred_mean_all.jpeg", width = 17, height = 10,
622     units = "cm", res = 2000, type = "cairo"
623 )
624 ggp
625 north2(ggp = ggp, scale = 0.065, x = 0.88, y = 0.04, symbol = 10)
626 dev.off()
627
628 # ---- 16. Lower Limit of Predicted Cases (95% CI) -----
629 dw_ll <- dataf %>% select (id_key_time, ID_MN_RESI, pred_ll) %>%
630     reshape(timevar = "id_key_time",
631             idvar = "ID_MN_RESI", direction = "wide")
632
633 map <- merge(MN_shp[,c("COD_MUN6", "name_muni", "abbrev_state", "name_region")], dw_ll, by
634 map_sf <- st_as_sf(map)
635
636 map_sf <- gather(map_sf, id_key_time, pred_ll, paste0("pred_ll.", 1:4, "_", rep(2015:2024,
637 map_sf$id <- gsub("pred_ll.", "", map_sf$id_key_time)
638 map_sf <- map_sf %>% separate(id , c("season", "year"), sep="_")
639
640 map_sf$season [map_sf$season == "1"] <- "Summer"
641 map_sf$season [map_sf$season == "2"] <- "Autumn"
642 map_sf$season [map_sf$season == "3"] <- "Winter"
643 map_sf$season [map_sf$season == "4"] <- "Spring"
644 map_sf$season <- factor(map_sf$season , levels = c("Summer", "Autumn", "Winter", "Spring" )
645
646 map_sf$id <- paste(map_sf$season, map_sf$year)
647 map_sf$id <- factor(map_sf$id,
648                     levels = c("Summer 2015", "Summer 2016", "Summer 2017", "Summer 2018", "S
649                               "Autumn 2015", "Autumn 2016", "Autumn 2017", "Autumn 2018", "A
650                               "Winter 2015", "Winter 2016", "Winter 2017", "Winter 2018", "W
651                               "Spring 2015", "Spring 2016", "Spring 2017", "Spring 2018", "S
652                               ))
653 map_sf$pred_ll <- as.numeric(map_sf$pred_ll)
654 map_sf$quantil <- NA
655 map_sf$quantil[map_sf$pred_ll > 0 & map_sf$pred_ll < 0.2] <- "0.01-0.19"
656 map_sf$quantil[map_sf$pred_ll >= 0.2 & map_sf$pred_ll < 1] <- "0.2-0.9"
657 map_sf$quantil[map_sf$pred_ll >= 1 & map_sf$pred_ll < 3] <- "1.0-2.9"
658 map_sf$quantil[map_sf$pred_ll >= 3 & map_sf$pred_ll < 20] <- "3.0-19.9"
659 map_sf$quantil[map_sf$pred_ll >= 20] <- ">20"
660 map_sf$quantil <- factor(map_sf$quantil,

```

```

661         levels = c("0.01-0.19", "0.2-0.9", "1.0-2.9", "3.0-19.9", ">20"))
662
663 ggp <- ggplot() +
664   geom_sf(data = map_sf, aes(fill = quantil), color = "transparent") +
665   geom_sf(data = reg_main, fill = "transparent", aes(color = "Brazil's regions"), linewidth
666   theme_classic() +
667   scale_fill_manual(name = "", values= c("#0277BD", '#abdda4', '#ffffbf', '#fdae61', "red")
668     limits = c("0.01-0.19", "0.2-0.9", "1.0-2.9", "3.0-19.9", ">20")) +
669   scale_color_manual( name = "", values = c("Brazil's regions" = "black") ) +
670   guides(
671     fill = guide_legend(order = 1,
672       label.theme = element_text(size = 8),
673       keywidth = unit(0.3, "cm"),
674       keyheight = unit(0.3, "cm"),
675       nrow = 1),
676     color = guide_legend( order = 2,
677       override.aes = list(fill = NA, linetype = 1, size = 0.2),
678       label.theme = element_text(size = 8),
679       keywidth = unit(0.28, "cm"),
680       keyheight = unit(0.28, "cm"),
681       nrow = 1)) +
682   theme(legend.text = element_text(size = 10, face = "bold"),
683     legend.direction = "horizontal",
684     legend.position = "top",
685     legend.box = "horizontal",
686     legend.margin = margin(0, 0, 0, 0),
687     text = element_text(size = 10),
688     axis.line = element_blank(),
689     axis.text = element_blank(),
690     axis.ticks = element_blank(),
691     strip.text = element_text(size = 4),
692     panel.spacing = unit(0.01, "lines"),
693     panel.spacing.x = unit(0.01, "lines"),
694     panel.spacing.y = unit(0.01, "lines")) +
695   facet_wrap(~ id, ncol = 10 ) +
696   labs(title = "",
697     x = "",
698     y = "") +
699   blank() +
700   scalebar(map_sf, dist = 1000, dist_unit = "km", st.size = 1.1,
701     height=0.08, border.size = 0.1,
702     st.dist = 0.1,
703     anchor = c(x = -35, y = -38),
704     transform = TRUE, model = "WGS84",
705     facet.var = c("id"),
706     facet.lev = c("Spring 2024"))
707
708 jpeg("FigS7_map_pred_ll.jpeg", width = 17, height = 10,
709     units = "cm", res = 2000, type = "cairo"
710 )
711 ggp
712 north2(ggp = ggp, scale = 0.065, x = 0.88, y = 0.04, symbol = 10)

```

```

713 dev.off()
714
715
716 # ---- 17. Map: Upper Limit of Predicted Cases (95% CI) -----
717 dw_ul <- dataf %>% select (id_key_time, ID_MN_RESI, pred_ul) %>%
718   reshape(timevar = "id_key_time",
719           idvar = "ID_MN_RESI", direction = "wide")
720 map <- merge(MN_shp[,c("COD_MUN6", "name_muni", "abbrev_state", "name_region")], dw_ul, by
721 map_sf <- st_as_sf(map)
722
723 map_sf <- gather(map_sf, id_key_time, pred_ul, paste0("pred_ul.", 1:4, "_", rep(2015:2024,
724 map_sf$id <- gsub("pred_ul.", "", map_sf$id_key_time)
725 map_sf <- map_sf %>% separate(id, c("season", "year"), sep="_")
726
727 map_sf$season [map_sf$season == "1"] <- "Summer"
728 map_sf$season [map_sf$season == "2"] <- "Autumn"
729 map_sf$season [map_sf$season == "3"] <- "Winter"
730 map_sf$season [map_sf$season == "4"] <- "Spring"
731 map_sf$season <- factor(map_sf$season, levels = c("Summer", "Autumn", "Winter", "Spring"))
732
733 map_sf$id <- paste(map_sf$season, map_sf$year)
734 map_sf$id <- factor(map_sf$id,
735                     levels = c("Summer 2015", "Summer 2016", "Summer 2017", "Summer 2018", "S
736                               "Autumn 2015", "Autumn 2016", "Autumn 2017", "Autumn 2018", "A
737                               "Winter 2015", "Winter 2016", "Winter 2017", "Winter 2018", "W
738                               "Spring 2015", "Spring 2016", "Spring 2017", "Spring 2018", "S
739                               ))
740
741 map_sf$pred_ul <- as.numeric(map_sf$pred_ul)
742 map_sf$quantil <- NA
743
744 map_sf$quantil[map_sf$pred_ul > 0 & map_sf$pred_ul < 0.2] <- "0.01-0.19"
745 map_sf$quantil[map_sf$pred_ul >= 0.2 & map_sf$pred_ul < 1] <- "0.2-0.9"
746 map_sf$quantil[map_sf$pred_ul >= 1 & map_sf$pred_ul < 3] <- "1.0-2.9"
747 map_sf$quantil[map_sf$pred_ul >= 3 & map_sf$pred_ul < 20] <- "3.0-19.9"
748 map_sf$quantil[map_sf$pred_ul >= 20] <- ">20"
749 map_sf$quantil <- factor(map_sf$quantil,
750                          levels = c("0.01-0.19", "0.2-0.9", "1.0-2.9", "3.0-19.9", ">20"))
751
752 ggp <- ggplot() +
753   geom_sf(data = map_sf, aes(fill = quantil), color = "transparent") +
754   geom_sf(data = reg_main, fill = "transparent", aes(color = "Brazil's regions"), linewidtht
755   theme_classic() +
756   scale_fill_manual(name = "", values = c("#0277BD", "#abdda4", "#ffffbf", "#fdae61", "red")
757                     limits = c("0.01-0.19", "0.2-0.9", "1.0-2.9", "3.0-19.9", ">20")) +
758   scale_color_manual(name = "", values = c("Brazil's regions" = "black")) +
759   guides(
760     fill = guide_legend(order = 1,
761                         label.theme = element_text(size = 8),
762                         keywidth = unit(0.3, "cm"),
763                         keyheight = unit(0.3, "cm"),
764                         nrow = 1),

```

```

765     color = guide_legend( order = 2,
766                           override.aes = list(fill = NA, linetype = 1, size = 0.2),
767                           label.theme = element_text(size = 8),
768                           keywidth = unit(0.28, "cm"),
769                           keyheight = unit(0.28, "cm"),
770                           nrow = 1)) +
771 theme(legend.text = element_text(size = 10, face = "bold"),
772       legend.direction = "horizontal",
773       legend.position = "top",
774       legend.box = "horizontal",
775       legend.margin = margin(0, 0, 0, 0),
776       text = element_text(size = 10),
777       axis.line = element_blank(),
778       axis.text = element_blank(),
779       axis.ticks = element_blank(),
780       strip.text = element_text(size = 4),
781       panel.spacing = unit(0.01, "lines"),
782       panel.spacing.x = unit(0.01, "lines"),
783       panel.spacing.y = unit(0.01, "lines")) +
784 facet_wrap(~ id, ncol = 10 ) +
785 labs(title = "",
786      x = "",
787      y = "") +
788 blank() +
789 scalebar(map_sf, dist = 1000, dist_unit = "km", st.size = 1.1,
790         height=0.08, border.size = 0.1,
791         st.dist = 0.1,
792         anchor = c(x = -35, y = -38),
793         transform = TRUE, model = "WGS84",
794         facet.var = c("id"),
795         facet.lev = c("Spring 2024"))
796
797
798 jpeg("FigS8_map_pred_ul.jpeg", width = 17, height = 10,
799     units = "cm", res = 2000, type = "cairo"
800 )
801 ggp
802 north2(ggp = ggp, scale = 0.065, x = 0.88, y = 0.04, symbol = 10)
803 dev.off()
804
805

```