



Spatial Modeling Covid Mortality in South America

Miguel Calvo Valente, 203129

Nelson Alejandro Gil Vargas, 203058

Valeria Roberts Trujillo, 173120

Profesor: Dr. Luis Enrique Nieto Barajas

Materia: Regresión Avanzada

May 2022

Índice de Contenidos

1	Introducción	6
1.1	Descripción del problema	6
1.2	Contexto	6
1.3	Objetivos	6
2	Datos	7
2.0.1	Análisis Exploratorio de Datos	7
3	Modelado e implementación	9
3.1	Modelo intrínseco: ICAR	9
3.1.1	Modelo intrínseco: ICAR con $Y = \frac{Y}{c}, c = cte$	11
3.1.2	Modelo intrínseco: ICAR con $Y' = \log(Y + 1)$ y $E' = \log(E + 1)$	13
3.1.3	Modelo intrínseco: ICAR con $\log(Y + 1) \sim N(\log(\lambda) + \log(E + 1), \sigma_y^2)$	14
3.2	Modelo CAR propio	14
3.2.1	CAR propio con $Y = \frac{Y}{c}, c = cte$	17
3.2.2	CAR propio con $Y' = \log(Y + 1), E' = \log(E + 1)$	17
3.2.3	CAR propio con $\log(Y + 1) \sim N(\log(\lambda) + \log(E + 1), \sigma_y^2)$	17
3.3	Modelo Jerárquico	18
4	Resultados	20
4.1	Comparación de DIC	20
4.1.1	Y, E	20
4.1.2	$Y' = \log(Y + 1), E' = \log(E + 1)$	20
4.2	Análisis de mejores modelos	22
4.2.1	Y, E	22
4.2.2	Modelo transformado: $Y' = \log(Y + 1), E' = \log(E + 1)$	28
5	Conclusiones	34
6	Referencias	35
7	Apéndice	36
7.1	Gráficas para el modelo CAR propio con ρ al 90% sin transformaciones en Y, E	36
7.1.1	β 's:	36
7.1.2	λ 's:	37
7.1.3	ϕ 's:	39
7.1.4	θ 's:	41
7.1.5	Predicciones, yf 's:	43
7.2	Gráficas para el modelo CAR propio con ρ al 90% sin transformaciones en $Y' = \log(Y + 1), E' = \log(E + 1)$	45
7.2.1	β 's:	45

7.2.2	λ 's:	46
7.2.3	ϕ 's:	48
7.2.4	θ 's:	50
7.2.5	Predicciones, yf 's:	52
7.3	Código R Base	54
7.4	Códigos Bugs	71

Índice de Figuras

1	Países de América del Sur estudiados.	10
2	Cadenas del modelo ICAR con $y/1000$	12
3	Cadenas del modelo ICAR con $Y' = \log(Y + 1)$ y $E' = \log(E + 1)$	13
4	Cadenas del modelo ICAR con $\log(Y + 1) \sim N(\log(\lambda) + \log(E + 1), \sigma_y^2)$	14
5	cadenas modelo CAR propio con $y/1000$	17
6	cadenas modelo jerárquico	19
7	Efecto combinado	23
8	Efecto individual de los países.	24
9	Efecto espacial de los países.	24
10	Comparación de tasas de riesgo: Observadas vs Suavizadas.	25
11	Predicciones : CAR propio vs Jerárquico	26
12	CAR propio : Y vs \hat{Y}	27
13	Efecto individual y espacial de los países.	29
14	Efecto individual de los países.	30
15	Efecto espacial de los países.	30
16	Comparación de tasas de riesgo: Observadas vs Suavizadas.	31
17	Predicciones CAR propio con ρ al 90% y $Y' = \log(Y + 1)$, $E' = \log(E + 1)$	32
18	Y vs \hat{Y} CAR propio con ρ al 90% y $Y' = \log(Y + 1)$, $E' = \log(E + 1)$	33

Índice de Tablas

1	Valores promedio de las variables por país.	8
2	Variables sin transformar para los modelos.	8
3	Países vecinos.	10
4	DIC Desempeño de Modelos Y, E	20
5	DIC Desempeño de Modelos LogLog. $\log(Y + 1), \log(E + 1)$	21
6	Resumen de Betas en Modelo Y, E	22
7	Resumen de Betas en Modelo Loglog.	28

1 Introducción

1.1 Descripción del problema

La regresión espacial es un enfoque popular en muchas disciplinas para estimar modelos de regresión en presencia de correlación espacial. El uso de la correlación espacial en nuestros modelos nos permite comprender más profundamente un área geográfica y como se relacionan sus regiones entre ellas.

Al analizar la propagación y el impacto de la enfermedad COVID-19, observamos una correlación espacial incontrolada en las muertes entre países vecinos. Esto puede ser el resultado del comercio o las migraciones entre países vecinos o el resultado de otras propiedades del clima o la geografía que son difíciles de reflejar en un modelo.

En este trabajo, nos interesa identificar si el control de esta auto correlación proporciona estimaciones más precisas a nuestro modelo, que reflejen la realidad del área geográfica. así como si la vecindad entre países refleja nociones honestas de similitud entre naciones que, de otro modo, podrían ser mejor captadas por el número total de vuelos, la migración entre países o el comercio.

1.2 Contexto

El año ante pasado, 2020, conceptos como “confinamientos”, “uso obligatorio de mascarillas” y “distanciamiento social” eran desconocidos para la mayoría de nosotros. Hoy forman parte de nuestro vocabulario habitual. El SARS-COV2 apareció en China el diciembre 2019 y provoca una enfermedad llamada COVID-19, que se ha extendido por el mundo y fue declarada pandemia global por la Organización Mundial de la Salud.

Mientras la pandemia de COVID-19 sigue afectando todos los aspectos de nuestras vidas, nos interesa estudiar sus efectos y como nos relaciona con el mundo.

1.3 Objetivos

Los objetivos del trabajo es utilizar un modelo espacial para entender un problema práctico, en específico la relación espacial de mortalidad de COVID entre los países de Latinoamérica. Además, nos gustaría incluir el efecto de la migración bajo el supuesto que está variable tiene poder explicativo.

2 Datos

Los datos espaciales surgen cuando las observaciones están geográficamente referenciadas. En nuestro caso contamos con una base de datos obtenida de (3). Los datos están basados en el concurso de *Kaggle COVID 19 Global Forecasting* (4) e indicadores demográficos del dataset *Countries of the World* (5).

La base de datos contiene 611 datos y 36 variables. Las variables más interesantes para nuestro análisis son:

- Country: país.
- Fatalities: Fallecimientos totales por COVID. Los recuentos incluyen los confirmados y los probables (cuando se informa de ello).
- Continent: Continente del país.
- Geometry: Polígonos del país. Un polígono es una figura cerrada definida por una secuencia de pares x,y conectados.
- Population: Población total del país.
- Net Migration: Migración neta, por cada 100,000 personas
- Death Rate: Fallecimientos por cualquier causa, por cada 100,000 personas.

2.0.1 Análisis Exploratorio de Datos

La base de datos contiene información de todos los países del mundo. Sin embargo, nos interesa analizar en específico Sudamérica, por lo que filtramos los países de interés. Se tiene información de 11 países: Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Guyana, Paraguay, Perú, Uruguay y Venezuela.

Tras el filtrado anterior contamos con 45 datos. Esto ocurre debido a que la base cuenta con información de varias semanas, es decir, que tenemos varios datos para un mismo país, por lo tanto, para unificarlo determinamos el promedio de las variables de interés y se junto en una única columna la información de un país.

Los valores promedio de nuestras variables por país son (redondeando *Fatalities*):

Country	Population	Fatalities	Death Rate	Net Migration
Argentina	39921833	25	7.55	0.61
Bolivia	8989046	0	7.53	-1.32
Brazil	188078227	139	6.17	-0.03
Chile	16134219	10	5.81	0
Colombia	43593035	13	5.58	-0.31
Ecuador	13547510	62	4.23	-8.58
Guyana	767245	2	8.28	-2.07
Paraguay	6506464	6	4.49	-0.08
Perú	28302603	25	6.23	-1.05
Uruguay	3431932	0	9.05	-0.32
Venezuela	5730435	2	4.92	-0.04

Tabla 1: Valores promedio de las variables por país.

Para nuestro objetivo, definimos la variable respuesta Y_i como la cantidad de fallecimientos totales por COVID en cada país, en este caso, la variable *Fatalities*. Con el propósito de ocupar un modelo Poisson, definimos el número esperado de fallecimientos por COVID, E_i , como un porcentaje sensible del total de muertes por país. Tras investigar posibles valores para este porcentaje, decidimos fijarlo en un 0.5% del total de fallecimientos. Es decir, $E_i = 0.5\% \times Population_i \times \frac{DeathRate_i}{100000}$. Consideramos que la migración neta puede añadir valor explicativo en el modelo, por lo que definimos X_i como *Net Migration*.

A manera de resumen, los datos que ocupamos se ven de la siguiente forma:

Country	$Y = \text{Fatalities}$	$E = \text{Expected Covid Deaths}$	$X = \text{Net Migration}$
Argentina	25	15.1	0.61
Bolivia	0	3.38	-1.32
Brazil	139	58	-0.03
Chile	10	4.69	0
Colombia	13	12.2	-0.31
Ecuador	62	2.87	-8.58
Guyana	2	0.318	-2.07
Paraguay	6	1.46	-0.08
Perú	25	8.82	-1.05
Uruguay	0	1.55	-0.32
Venezuela	2	6.33	-0.04

Tabla 2: Variables sin transformar para los modelos.

3 Modelado e implementación

Los modelos espaciales son modelos probabilísticos que reconocen la dependencia espacial entre cualesquiera dos observaciones Y_{s_1} y Y_{s_2} .

Debido a que nuestros datos son "datos en áreas": donde Y_s es una v.a. en la localización discreta $s \in N$ o en un subconjunto finito en donde cada localización está bien delimitada, e.g. municipios, estados o países.

Dado que nos interesa usar distribuciones para los modelos CAR intrínseco (ICAR) y CAR propio, las cuales están implementadas en BUGS, todo el análisis siguiente se realizara en BUGS.

A continuación se listan todas las especificaciones y detalles de los modelos probados. La comparación y análisis de los modelos se realizará posteriormente en la sección 4 de *Resultados*.

3.1 Modelo intrínseco: ICAR

El modelo intrínseco ICAR, es una especificación condicional del modelo condicionalmente auto-regresivo CAR.

$$f(y_i|y_j, j \neq i) = N\left(\frac{1}{m_i} \sum_{j=1}^n w_{i,j} y_j, \frac{\sigma^2}{m_i}\right)$$

Donde $Y' = (Y_1, Y_2, \dots, Y_n)$ se distribuye $f(y) \propto \exp\left\{-\frac{1}{2\sigma^2} y'(D - W)y\right\}$

Una aplicación común en bioestadística y epidemiología es el mapeo de enfermedades. En este caso tenemos datos de conteo del siguiente tipo:

- Y_i = número observado de muertes de Covid-19 en un país, $i = 1, \dots, 11$
- E_i = número esperado de muertes de Covid-19 en un país, $i = 1, \dots, 11$
- $w_{i,j} = I(i \sim j)$, es una variable indicadora que determina si la localización i es vecina de la localización j .
- $W = (w_{i,j})$, es la matriz indicadora de vecindades. En este caso, los vecinos se muestran en la tabla 3 y la ubicación de los países se observa en la figura 1.
- $m_i = \sum_{j=1}^n w_{i,j}$, es el número de vecinos de la localización i .
- $D = \text{diag}(m_1, m_2, \dots, m_n)$

La matriz W, queda de la siguiente forma (reemplazando los checks por 1 y el resto con cero):

	Arg	Bol	Bra	Chi	Col	Ecu	Guy	Par	Per	Uru	Ven
Argentina		✓	✓	✓				✓		✓	
Bolivia	✓		✓	✓				✓	✓		
Brazil	✓	✓			✓		✓	✓	✓	✓	✓
Chile	✓	✓							✓		
Colombia			✓			✓			✓		✓
Ecuador					✓				✓		
Guyana			✓								✓
Paraguay	✓	✓	✓								
Perú		✓	✓	✓	✓	✓					
Uruguay	✓		✓								
Venezuela			✓		✓		✓				

Tabla 3: Países vecinos.



Figura 1: Países de América del Sur estudiados.

Por lo tanto, consideramos, el siguiente modelo:

$$Y_i \sim Po(E_i \lambda_i)$$

$$\log(\lambda_i) = \beta_1 + \beta_2 X_i + \theta_i + \phi_i$$

donde X_i es la migración.

Con efectos aleatorios individuales:

- $\theta_i \sim N(0, \tau_t)^*$
- $\tau_t \sim Gamma(0.001, 0.001)^*$
- * Distribución en formato BUGS (con precisión)

Y efectos especiales:

- $\phi_i \sim CAR_{normal}(W, D, \tau_c)$
- $\tau_c \sim Ga(0.1, 0.1)$

Con las siguientes distribuciones iniciales no informativas:

- $\beta_1 \sim N(0, 0.001)^*$
- $\beta_2 \sim N(0, 0.001)^*$
- * Distribución en formato BUGS (con precisión)

Para llevar a cabo las cadenas se utilizaron los siguientes parámetros:

- n.iter = 10000
- n.chains = 3
- n.burnin = 1000
- n.thin = 1

Sin embargo, el modelo no fue capaz de compilar por la escala de Y . Al realizar el debug con BUGS, arroja un error indicando que la media de la Poisson es demasiado grande. Derivado de esto, decidimos escalar tanto Y como E dividiendo por alguna constante.

3.1.1 Modelo intrínseco: ICAR con $Y = \frac{Y}{c}, c = cte$

Se mantienen las mismas distribuciones que el modelo ICAR, sin embargo se probó:

$$y = \frac{y}{c}, c = 10, 100, 1000, 10000$$

Sin embargo, de nuevo nos enfrentamos con que el modelo no es capaz de compilar en todos los casos menos en $c = 1000$. Inclusive, como podemos observar en la gráfica siguiente, en este único caso ($y/1000$), aunque el modelo si logró compilar, se observa que no hay convergencia en las cadenas, por lo que este modelo fue descartado.

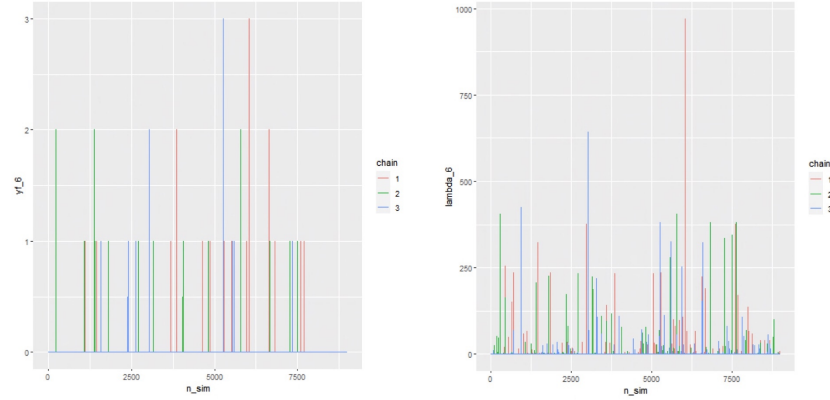


Figura 2: Cadenas del modelo ICAR con $y/1000$

3.1.2 Modelo intrínseco: ICAR con $Y' = \log(Y + 1)$ y $E' = \log(E + 1)$

Nuevamente, se mantienen las mismas distribuciones que el modelo ICAR, pero se decidió utilizar una transformación que logrará corregir los errores de compilación.

Se propone la siguiente transformación: $Y' = \log(Y + 1)$ y $E' = \log(E + 1)$, de esta manera nos encontramos en una escala positiva, y con el +1 evitar la posibilidad de tener valores con $-\infty$. En la siguiente gráfica, se puede observar que el modelo compila correctamente y que sus cadenas se mezclan adecuadamente.

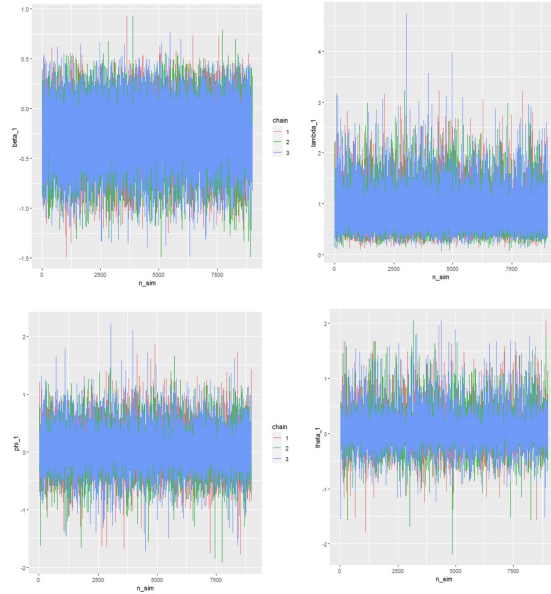


Figura 3: Cadenas del modelo ICAR con $Y' = \log(Y + 1)$ y $E' = \log(E + 1)$

3.1.3 Modelo intrínseco: ICAR con $\log(Y + 1) \sim N(\log(\lambda) + \log(E + 1), \sigma_y^2)$

Si $Y \sim \text{Poisson}(E\lambda)$, entonces $\lambda = \frac{Y}{E}$. De ello que $\log(Y) = \log(\lambda) + \log(E)$, y podemos definir un modelo normal para $\log(Y)$, para hacerlo comparable con el caso sin transformar Y, E . Por lo tanto definimos,

$$\log(Y + 1) \sim N(\log(\lambda) + \log(E + 1), \sigma_y^2)$$

De nuevo podemos observar en las gráficas siguientes que el modelo compila adecuadamente, y las cadenas son densas, mostrando que se están mezclando correctamente.

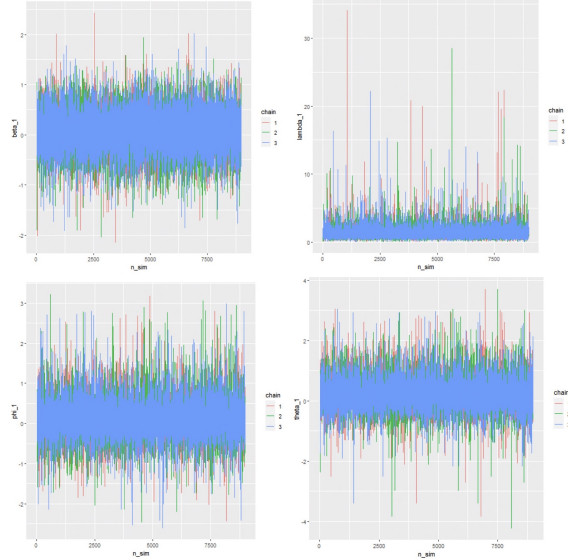


Figura 4: Cadenas del modelo ICAR con $\log(Y + 1) \sim N(\log(\lambda) + \log(E + 1), \sigma_y^2)$

3.2 Modelo CAR propio

El modelo CAR propio es una especificación condicional del Modelo condicionalmente auto regresivo CAR, similar a la condicional de ICAR, pero en este caso la a priori es una distribución propia, es decir que cumpla con todas las especificaciones de una distribución.

$$f(y_i | y_j, j \neq i) = N \left(\frac{p}{m_i} \sum_{j=1}^n w_{i,j} y_j, \frac{\sigma^2}{m_i} \right)$$

Donde p es un parámetro adicional que controla la dependencia entre vecinos,

$$p \in \left(\frac{1}{\lambda_{(1)}}, \frac{1}{\lambda_{(n)}} \right)$$

donde $\lambda_{(1)} \leq \dots \leq \lambda_{(n)}$ son los eigenvalores de la matriz $D^{-1/2}WD^{-1/2}$

Esto implica que la distribución conjunta de $Y' = (Y_1, Y_2, \dots, Y_n)$ es:

$$Y \sim N(0, \sigma^2(D - pW)^{-1})$$

La documentación de GeoBUGS indica una parametrización distinta para W y D . Entonces, para poder implementar el CAR propio, definimos W' y D' de la siguiente forma:

- $W' = (\frac{w_{i,j}}{m_i})$, con $w_{i,j}$ igual que en la otra parametrización es la indicadora de vecindad entre i y j . [3.2](#)

$$\begin{bmatrix} 0 & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & 0 & 0 & 0 & \frac{1}{5} & 0 & \frac{1}{5} & 0 \\ \frac{1}{5} & 0 & \frac{1}{5} & \frac{1}{5} & 0 & 0 & 0 & \frac{1}{5} & \frac{1}{5} & 0 & 0 \\ \frac{1}{5} & \frac{1}{5} & 0 & 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{5} & \frac{1}{5} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & \frac{1}{5} & \frac{1}{5} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \end{bmatrix}$$

- $D' = \text{diag}(\frac{1}{m_1}, \frac{1}{m_2}, \dots, \frac{1}{m_n})$

Es importante notar que para aplicar CAR propio **no es posible tener países sin vecinos**.

Para llevar a cabo las cadenas de CAR propio se utilizaron los siguientes parámetros:

- n.iter = 10000
- n.chains = 3
- n.burnin = 1000
- n.thin = 1

Con CAR propio requerimos definir ρ . Este es un valor contenido entre $(\frac{1}{\lambda_{(1)}}, \frac{1}{\lambda_{(n)}})$, donde $\lambda_{(i)}$ son los eigenvalores de la matriz $D^{-1/2}WD^{-1/2}$. No obstante, dada la parametrización de BUGS con W' y D' , el manual indica que se debe hacer el cálculo de los eigenvalores de $D'^{-1/2}W'D'^{1/2}$.

Calculamos los eigenvalores de ambas formas y verificamos que en efecto dieran lo mismo. Esto nos dio más certeza en que a pesar de que las parametrizaciones difirieran, BUGS iba a modelar correctamente el CAR propio.

Para nuestro caso de estudio en Sudamérica, los eigenvalores mínimo y máximo resultaron ser $(\lambda_{(1)}, \lambda_{(n)}) = (-0.58, 1)$, por lo que $\rho \in (\frac{1}{\lambda_{(1)}}, \frac{1}{\lambda_{(n)}}) = (-1.71, 1)$. Para elegir ρ , interpolamos valores en este rango con $w \in (0\%, 100\%)$ donde $w = 0\% \implies \rho = -1.71$ y $w = 100\% \implies \rho = 1$. Ocupamos las siguientes combinaciones:

- $w = 70\% \implies \rho = 0.186$
- $w = 80\% \implies \rho = 0.457$
- $w = 90\% \implies \rho = 0.729$

En todos estos casos, observamos un comportamiento correcto de las cadenas, estas se mezclan correctamente. Dado que con las pruebas de ρ se observó que el mejor desempeño era $w = 90\%$, se probaron algunas combinaciones:

- nox: Sin considerar nuestra variable explicativa X_i , que es la migración. Con este cambio esperábamos ver si verdaderamente la variable explicativa influenciaba a la mejora del modelo, ya que se mostró no significativa, como veremos en la sección de *Resultados*.
- xsc: Escalando X_i . Hicimos pruebas escalando X_i solo para verificar si mejoraba el desempeño del modelo.

3.2.1 CAR propio con $Y = \frac{Y}{c}, c = cte$

Dado que para el modelo intrínseco: ICAR con $Y = \frac{Y}{c}, c = cte$ solo compila con $c = 1,000$, es la constante que se tomo para realizar las pruebas. En las siguientes gráficas, podemos observar que en este caso, nuevamente no hay un buen comportamiento de las cadenas, no se mezclan correctamente.

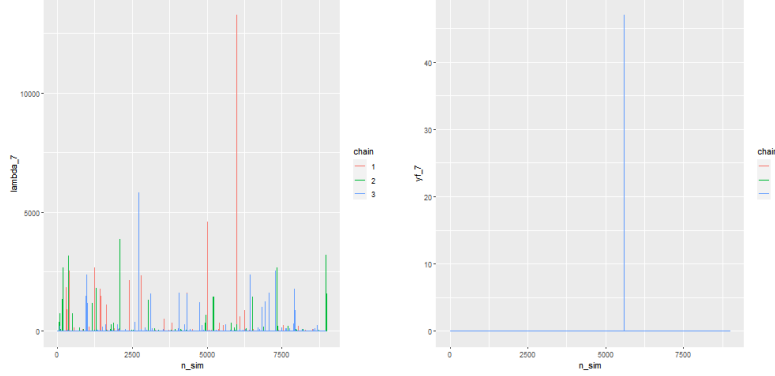


Figura 5: cadenas modelo CAR propio con $y/1000$

3.2.2 CAR propio con $Y' = \log(Y + 1), E' = \log(E + 1)$

Con la transformación de la Y podría ser que cambiara nuestra preferencia hacia ρ al 90%, la cual observamos que resultaba la mejor para el modelo CAR propio, sin ninguna transformación en la Y . Por lo tanto, se probaron las siguientes combinaciones:

- ρ al 70%
- ρ al 80 %
- ρ al 90 %

En este caso, observamos que nuevamente el mejor desempeño del modelo ocurre con ρ al 90%, es por eso que nuevamente se vuelven a probar las siguientes combinaciones:

- nox: Sin considerar nuestra variable explicativa " X_i ", que es la migración.
- xsc: Escalando " X_i "

3.2.3 CAR propio con $\log(Y + 1) \sim N(\log(\lambda) + \log(E + 1), \sigma_y^2)$

Con esta transformación volvemos a realizar pruebas para comprobar cual ρ es la mejor para modelar nuestro problema. Es por eso que se prueban las siguientes combinaciones:

- ρ al 70%
- ρ al 80%
- ρ al 90%

Y nuevamente observamos que el mejor modelo es cuando ρ al 90%.

3.3 Modelo Jerárquico

Decidimos comparar los modelos CAR con el modelo Jerárquico. Esto es para comprobar si el efecto espacial tiene poder descriptivo o no. Para el modelo Jerárquico continuamos con los siguientes supuestos:

- Y_i = número observado de muertes de Covid-19 en un país, $i = 1, \dots, 11$
- E_i = número esperado de muertes de Covid-19 en un país, $i = 1, \dots, 11$

Y el siguiente modelo:

$$Y_i \sim Po(E_i \lambda_i)$$

$$\log(\lambda_i) = \beta_1 + \beta_2 X_i + \theta_i$$

donde x_i es la migración.

Con efectos aleatorios individuales:

- $\theta_i \sim N(\mu_t, \tau_t)^*$
- $\mu_t \sim Normal(0, 0.1)^*$
- $\tau_t \sim Gamma(0.001, 0.001)^*$
- * Distribución en formato BUGS, con precisión

Es decir, el único efecto que se remueve son los efectos espaciales que se tenían. Por lo tanto, es como si no hubiera relación de vecindad entre ellos. Dado, este supuesto, esperaríamos que este modelo, no tenga un buen desempeño, ya que estamos removiendo información adicional.

Para llevar a cabo las cadenas jerárquicas se utilizaron los mismos parámetros que anteriormente:

- n.iter = 10000
- n.chains = 3
- n.burnin = 1000
- n.thin = 1

En este caso observamos un buen comportamiento de las cadenas, donde se mezclan correctamente.

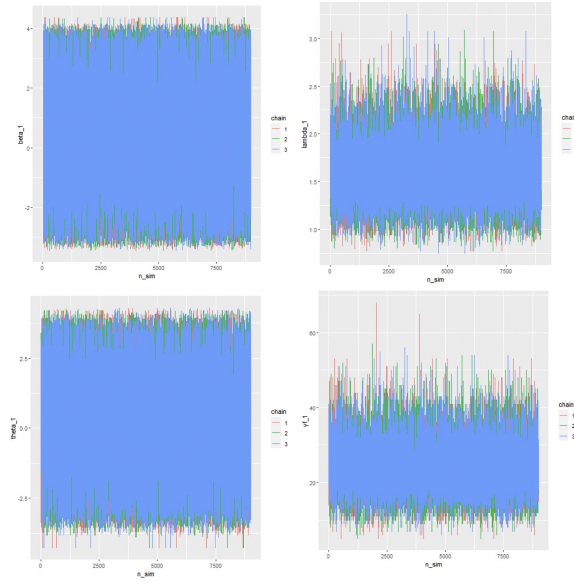


Figura 6: cadenas modelo jerárquico

4 Resultados

Previo a analizar los resultados de los mejores modelos, determinamos cual es el mejor para cada caso, tanto para Y, E sin transformar, como para $\log(Y + 1)$, $\log(E + 1)$. Calculamos el DIC de cada configuración para comparar todos los modelos y obtener los mejores.

Una vez seleccionados los mejores modelos, procedemos a analizar los efectos de la variable explicativa, los espaciales y los individuales. También revisamos las predicciones de Y para cada país, y comparamos el suavizamiento de la tasa relativa λ_i inducida por el modelo contra las originales.

4.1 Comparación de DIC

4.1.1 Y, E

Para el caso Y, E , observamos un claro mejor modelo, el CAR propio con ρ al 90%. Los siguientes mejores modelos fueron el CAR propio con X escalada (xsc) y sin X (nox) igual con ρ al 90%. Esto nos indica que en particular fue más importante determinar un valor apropiado para ρ antes que modificar algo en las variables explicativas. En general, los modelos con $Y_i \sim \text{Poisson}(\lambda_i E_i)$ (posiciones 1, 2, 3, 4, 5 y 9) se desempeñaron mejor que los de $\log(Y_i) \sim \text{Normal}(\log(\lambda_i) + \log(E_i), \sigma_y^2)$ (posiciones 6, 7 y 8). El peor modelo, como se esperaba, fue el jerárquico.

	Modelo	DIC
1	CAR Proper $\rho = 90$	-1438
2	CAR Proper $\rho = 90$ xsc	-393.1
3	CAR Proper $\rho = 90$ nox	-197.5
4	CAR Proper $\rho = 80$	-3.508
5	CAR Normal Normal	19.63
6	CAR Proper Normal $\rho = 70$	21.07
7	CAR Proper Normal $\rho = 90$	27.09
8	CAR Proper Normal $\rho = 80$	27.51
9	CAR Proper $\rho = 70$	39.64
10	Hierarchical	65.24

Tabla 4: DIC Desempeño de Modelos Y, E .

4.1.2 $Y' = \log(Y + 1), E' = \log(E + 1)$

Para el caso $Y' = \log(Y + 1), E' = \log(E + 1)$, observamos que los DIC están considerablemente más cercanos unos a los otros. De esto, podría ser que con

diferente configuración de parámetros, como el número de cadenas, de simulaciones, de adelgazamiento, etc., el orden cambie. No obstante, lo que obtuvimos nos indica que el mejor modelo fue nuevamente el CAR propio con ρ al 90%. A diferencia del caso Y, E , los modelos CAR propio con X escalada (xsc) y sin X (nox) igual con ρ al 90% tuvieron de las peores posiciones en la tabla de DIC (posiciones 5 y 7, la última). Algo fuera de lo esperado es que el jerárquico no fuera el último (posición 4).

	Modelo	DIC
1	CAR Proper ρ 90	27.69
2	CAR Normal	30.29
3	CAR Proper ρ 70	35.21
4	Hierarchical	35.57
5	CAR Proper ρ 90 nox	35.99
6	CAR Proper ρ 80	36.85
7	CAR Proper ρ 90 xsc	37.55

Tabla 5: DIC Desempeño de Modelos LogLog. $\log(Y + 1), \log(E + 1)$.

4.2 Análisis de mejores modelos

4.2.1 Y, E

El mejor modelo para Y, E fue el CAR propio con ρ al 90%. Este modelo contiene los parámetros β_1 para el intercepto, β_2 para el efecto de la migración neta X , θ_i con $i \in [1, n]$, $n = \text{número de países}$, para los efectos individuales, y ϕ_i con $i \in [1, n]$ para los efectos espaciales.

	Media	2.5 %	97.5 %	Probabilidad de no significancia
β_1	0.492	-1.379	2.537	0.251
β_2	-0.239	-0.566	0.127	0.087

Tabla 6: Resumen de Betas en Modelo Y, E.

El efecto de la migración neta es en promedio -0.239. Esto significa que para un cambio del 10% en la migración neta y dada una misma región i , tenemos:

$$\begin{aligned} \log(\lambda'_i) &= \beta_1 + \beta_2(X_i + 10\%) + \theta_i + \phi_i \\ -\log(\lambda_i) &= \beta_1 + \beta_2 X_i + \theta_i + \phi_i \end{aligned}$$

Se cancelan los términos $\beta_1, \theta_i, \phi_i$ ya que pertenecen a la misma región:

$$\begin{aligned} \log(\lambda'_i) - \log(\lambda_i) &= \beta_2(X_i + 10\%) - \beta_2(X_i) \\ \log\left(\frac{\lambda'_i}{\lambda_i}\right) &= \beta_2(10\%) \\ \frac{\lambda'_i}{\lambda_i} &= e^{\beta_2(10\%)} \\ &= e^{-0.239(10\%)} \\ &= 0.97638 \end{aligned}$$

Es decir, al aumentar la proporción de migración neta en una misma región en 10 puntos porcentuales, entonces el riesgo disminuye en un 2.4%. Este efecto es relativamente pequeño. Esto es reforzado al observar que la probabilidad de no significancia 8.7% supera a un nivel $\alpha = 5\%$. No obstante, dado que también probamos el modelo CAR propio sin X con ρ al 90% y obtuvo un menor DIC a este, entonces creemos que la migración sí está teniendo poder explicativo en la tasa λ .

Los efectos individuales θ_i y ϕ_i son **no identificables**. No obstante, su suma sí lo es. Observamos que en general todos los países contienen al 0 con un intervalo de confianza del 95%. Esto ocurre tanto para los efectos individuales θ_i , para los efectos espaciales ϕ_i y para la suma de ambos. Los únicos países con efectos un tanto más alejados del 0 son Uruguay, Bolivia y Venezuela. No obstante, como también corrimos el modelo Jerárquico, el cual tiene un desempeño peor a este, entonces consideramos que el efecto espacial sí debe estar teniendo poder descriptivo en λ_i

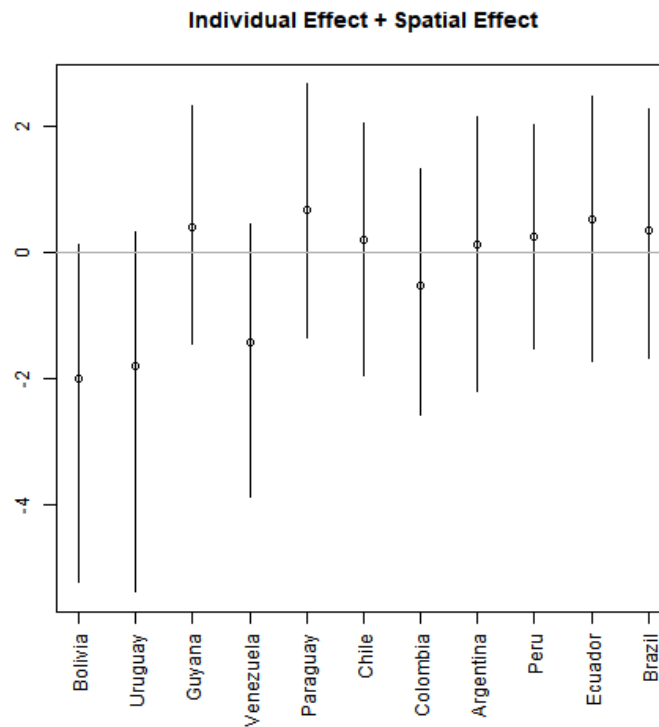


Figura 7: Efecto combinado

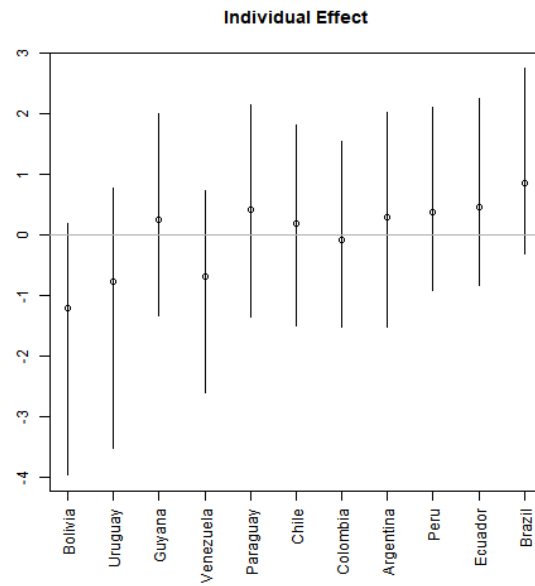


Figura 8: Efecto individual de los países.

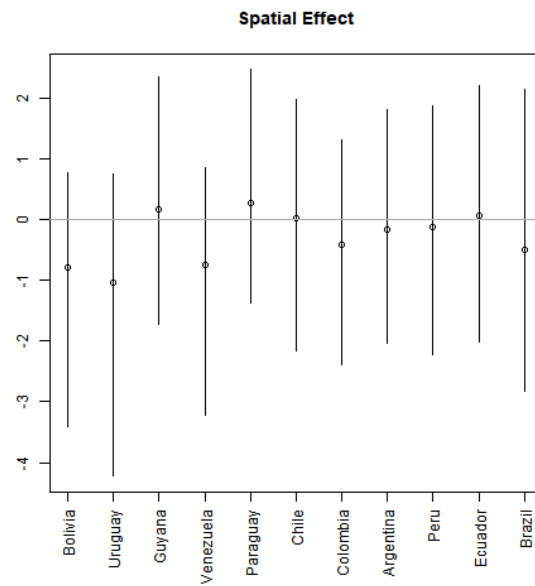


Figura 9: Efecto espacial de los países.

Las tasas relativas λ_i son suavizadas con el modelo, con base en la información que aporta la migración neta X y los vecinos. Esto se observa particularmente en países con una tasa muy baja o muy alta

Por ejemplo, tanto Bolivia como Uruguay tenían una tasa observada de 0, dado que las observaciones de casos de COVID en esos países igualmente era 0. A través del suavizamiento, estos países ahora tienen tasas de 0.46 y 0.52 respectivamente. Esto puede coincidir más con la realidad, ya que es poco probable que estos países tuvieran una tasa igual a 0 dado el rápido esparcimiento del virus.

Por otro lado, países como Guyana y Paraguay tenían tasas considerablemente altas, las cuales fueron suavizadas hacia abajo. Esto es algo que buscábamos realizar con el modelo, ya que por ciertos picos puede ser que estos países reflejaran una tasa mucho mayor de lo que ocurriera en el día a día.

Es interesante notar que en general las lambdas se quedan igual, cambian por muy poquito y si las ordenamos de menor a mayor, se preserva el orden (excepto venezuela y uruguay)

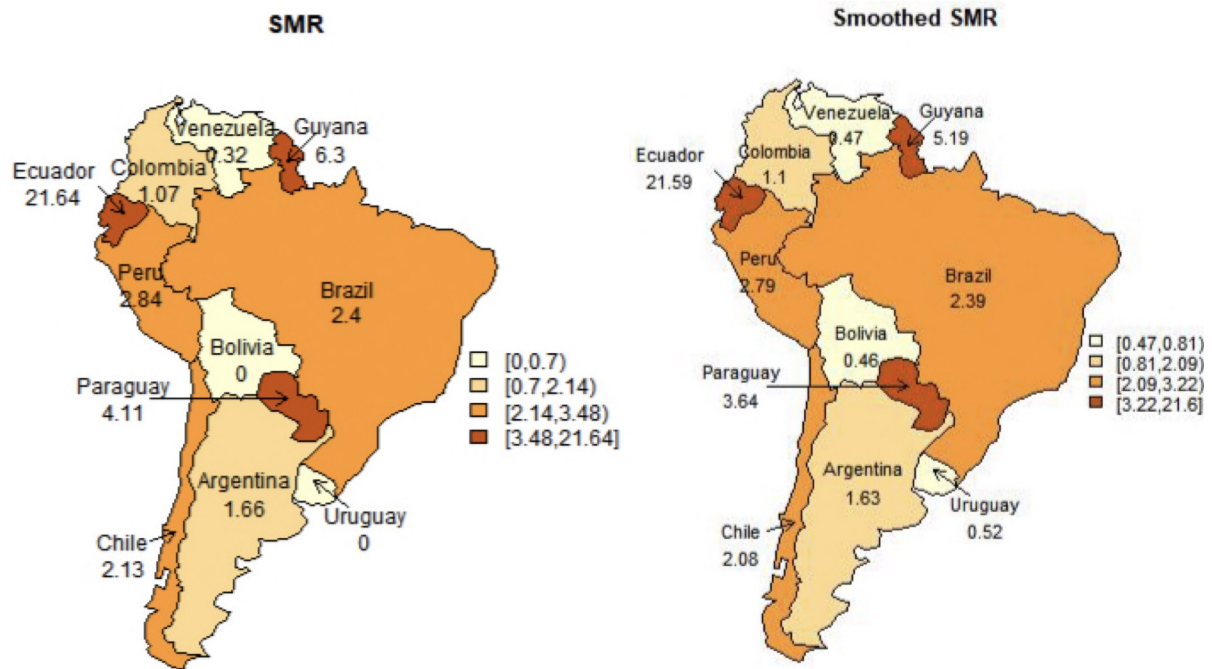


Figura 10: Comparación de tasas de riesgo: Observadas vs Suavizadas.

Observamos el comportamiento de las predicciones. Como referencia, comparamos contra el modelo con el peor desempeño, el jerárquico.

En general vemos que las predicciones en ambos casos son prácticamente idénticas entre sí, y que ambos ajustan muy bien a las Y observadas con muy poco suavizamiento. Los intervalos (línea punteada) son cercanos a la predicción (línea roja) en ambos casos, e igualmente son casi idénticos. Esto va de la mano con el hecho que los efectos espaciales en general se mostraban no significativos, resultando en predicciones muy similares entre los modelos CAR vs Jerárquico

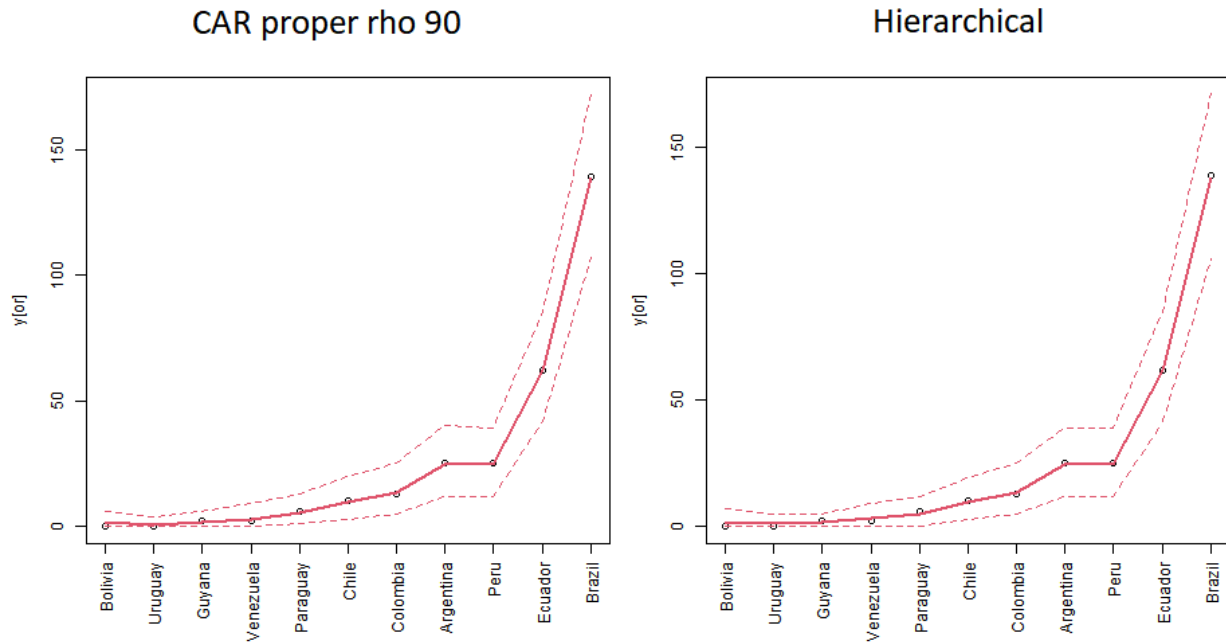


Figura 11: Predicciones : CAR propio vs Jerárquico

Las observaciones anteriores son reforzadas con la relación entre las predicciones \hat{Y} y las Y observadas, ya que hay un ajuste lineal casi perfecto, es decir un coeficiente de determinación $R^2 = 0.9997445 \approx 1$ en el CAR propio y de $R^2 = 0.9995524 \approx 1$ en el Jerárquico. La variación total de la variable "y" es completamente explicada por el modelo de regresión en ambos casos.

A primera vista puede parecer que nuestro modelo se está sobre ajustando, pero esto es falso dado que el término "sobre ajustar" únicamente aplica cuando contamos con muchos datos. Dado que en este caso específicamente únicamente contamos con 11 datos, no estamos sobre ajustando pero el suavizamiento resulta muy ligero en el caso de las predicciones, en contraste con el suavizamiento de

las λ 's que resulta más evidente.

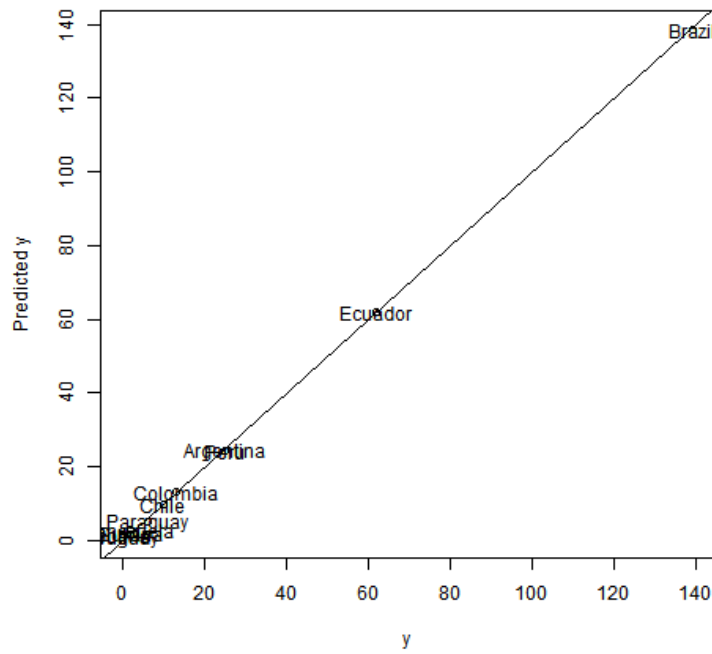


Figura 12: CAR propio : Y vs \hat{Y}

4.2.2 Modelo transformado: $Y' = \log(Y + 1)$, $E' = \log(E + 1)$

Al realizar una transformación en la Y , puede que perdamos interpretabilidad, pero muchas veces puede apoyarnos numéricamente para tener un mejor modelo. En este caso la transformación es la siguiente: $Y = \log(Y + 1)$ y $E = \log(E + 1)$, y el mejor modelo es el CAR propio con ρ al 90%.

	Media	2.5 %	97.5 %	Probabilidad de no significancia
β_1	-0.225	-1.208	0.676	0.299
β_2	-0.134	-0.339	0.083	0.101

Tabla 7: Resumen de Betas en Modelo Loglog.

El efecto de la migración neta, en este caso, es en promedio -0.134. Es decir, que para obtener un cambio del 10% en la migración neta y dada una misma región i , tenemos:

$$\begin{aligned} \log(\lambda'_i) &= \beta_1 + \beta_2(X_i + 10\%) + \theta_i + \phi_i \\ -\log(\lambda_i) &= \beta_1 + \beta_2 X_i + \theta_i + \phi_i \end{aligned}$$

Donde de nuevo se cancelan los términos β_1 , θ_i , ϕ_i ya que pertenecen a la misma región:

$$\begin{aligned} \log(\lambda'_i) - \log(\lambda_i) &= \beta_2(X_i + 10\%) - \beta_2(X_i) \\ \log\left(\frac{\lambda'_i}{\lambda_i}\right) &= \beta_2(10\%) \\ \frac{\lambda'_i}{\lambda_i} &= e^{\beta_2(10\%)} \\ &= e^{-0.13(10\%)} \\ &= 0.987084135 \end{aligned}$$

Es decir, al aumentar la proporción de migración neta en una misma región en 10 puntos porcentuales, entonces el riesgo disminuye en un 1.3%. Este efecto es relativamente pequeño. Esto es reforzado al observar que la probabilidad de no significancia 10.1% supera a un nivel $\alpha = 5\%$. No obstante, dado que para el caso $Y' = \log(Y + 1)$, $E' = \log(E + 1)$ también probamos el modelo CAR propio sin X con ρ al 90% y obtuvo un menor DIC a este, entonces creemos que la migración sí está teniendo poder explicativo en la tasa λ .

Observamos que en general todos los países contienen al 0 con un intervalo de confianza del 95% para ϕ_i , θ_i y la suma. Esto ocurre tanto para los efectos individuales θ_i , para los efectos espaciales ϕ_i y para la suma de ambos. A diferencia del caso Y , E , aquí las medias de los estimadores son mucho más cercanas al valor 0, por lo que el efecto espacial en general parece ser no significativo. No obstante, como también corrimos el modelo Jerárquico para este caso, el cual tiene un desempeño peor a este, entonces consideramos que el efecto espacial sí debe estar teniendo poder descriptivo en λ_i

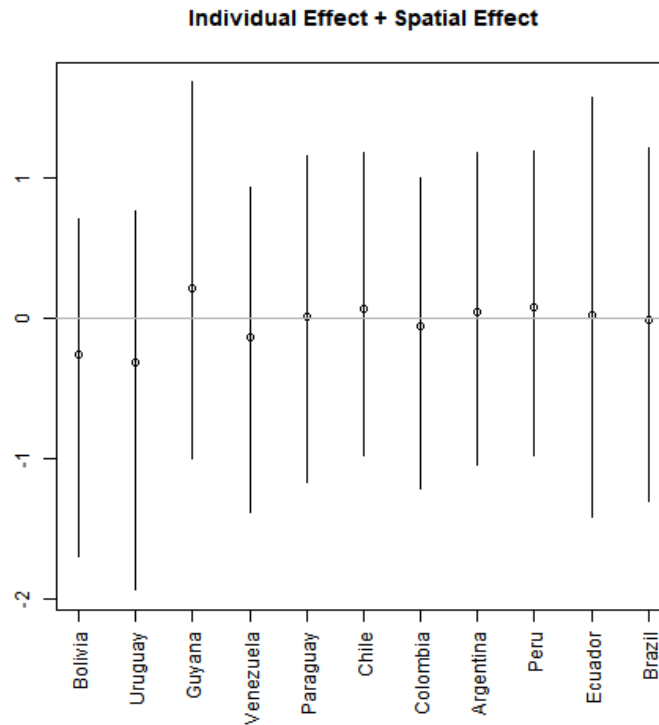


Figura 13: Efecto individual y espacial de los países.

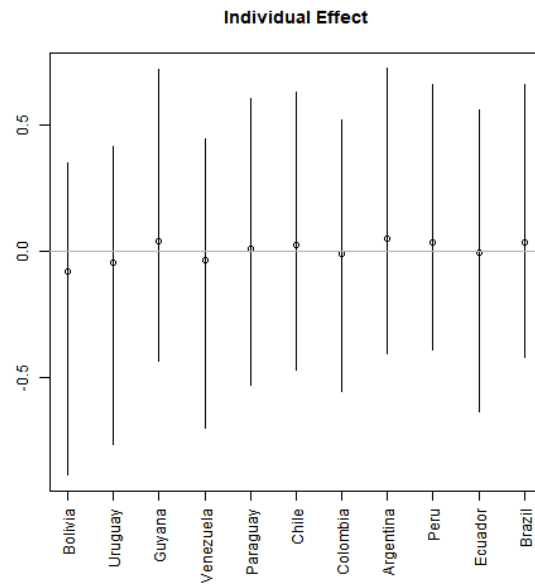


Figura 14: Efecto individual de los países.

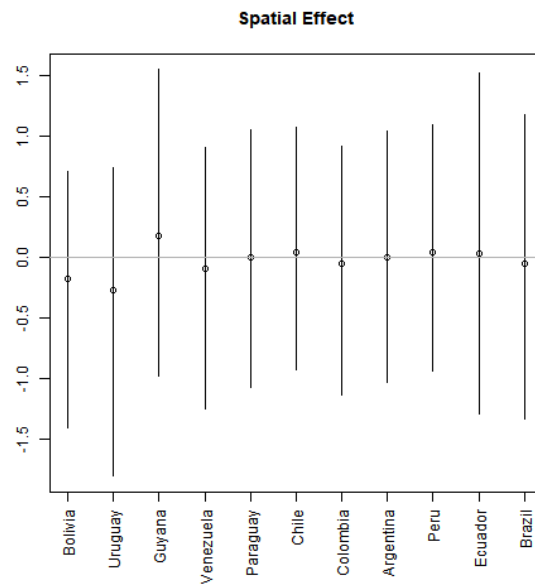


Figura 15: Efecto espacial de los países.

Las tasas relativas λ_i son suavizadas considerablemente. A diferencia del caso Y, E , el suavizamiento es mayor relativo a las λ_i observadas.

Igual que en el caso Y, E , tanto Bolivia como Uruguay tenían una tasa observada de 0, dado que las observaciones de casos de COVID en esos países igualmente era 0. A través del suavizamiento, estos países ahora tienen tasas de 0.84 y 0.75 respectivamente. Relativo al suavizamiento en Y, E , aumenta considerablemente el riesgo en estos países tras el suavizamiento

Es interesante notar como previamente en el caso Y, E , las lambdas casi no cambiaron en magnitud ni orden. No obstante, en este caso, hubieron movimientos muy drásticos, sean mayores o menores, como por ejemplo, el caso de Paraguay de 2.16 a 0.94, y Ecuador de 3.06 a 2.93, respectivamente.

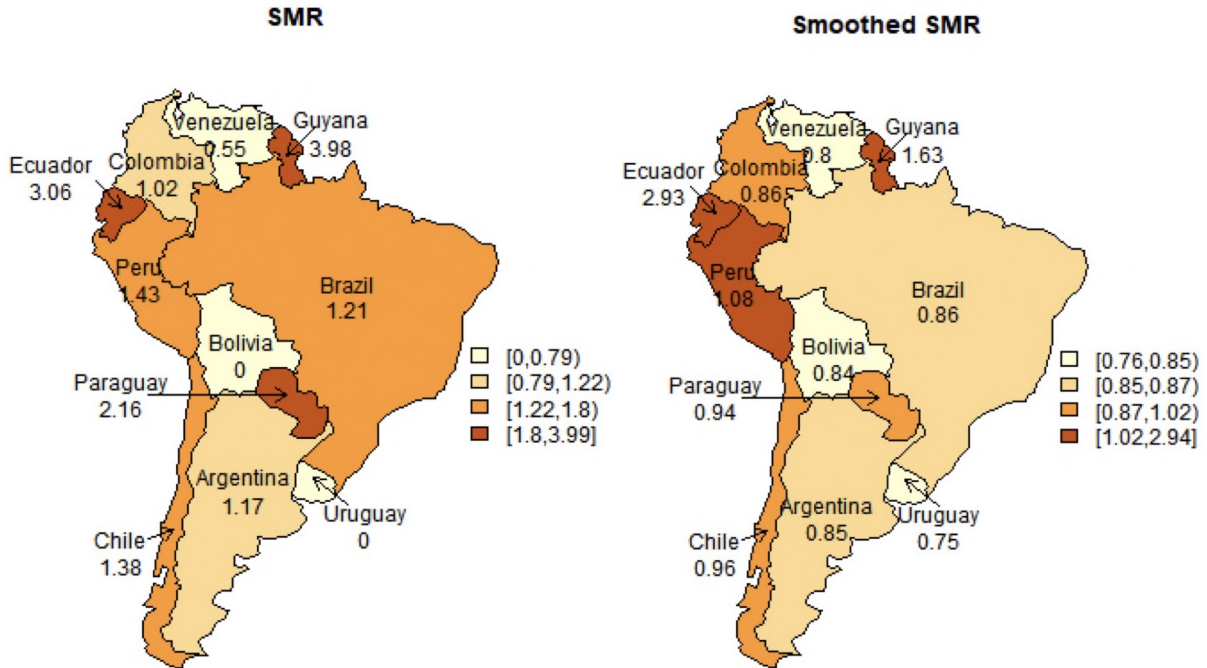


Figura 16: Comparación de tasas de riesgo: Observadas vs Suavizadas.

En la siguiente gráfica podemos observar el comportamiento de las predicciones. Si bien no es directamente comparable dada la transformación, notamos que en contraste con el modelo anterior, este ajuste tiene los intervalos (línea punteada) mucho más anchos que las predicciones pasadas, reflejando menor exactitud relativamente. Además, en la predicción (línea roja) podemos notar que no está ajustando tan detalladamente nuestros puntos. Es decir, en este caso sí se está realizando el suavizamiento.

Sin embargo, dado que el cambio en las jerarquía de las λ 's antes mencionado, es mucho más fuerte, se explica el porque las predicciones no se ven tan exactas gráficamente.

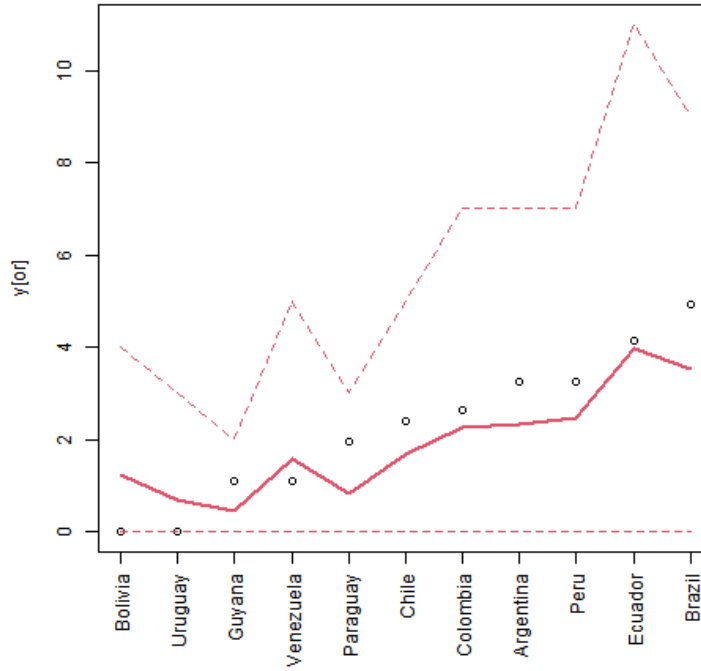


Figura 17: Predicciones CAR propio con ρ al 90% y $Y' = \log(Y + 1)$, $E' = \log(E + 1)$

En la gráfica anterior podemos observar nuestro estimador R^2 , en este caso $R^2 = 0.7648311 < 1$. Este indicador nos muestra que es un buen ajuste en general, dado el suavizamiento que estamos haciendo.

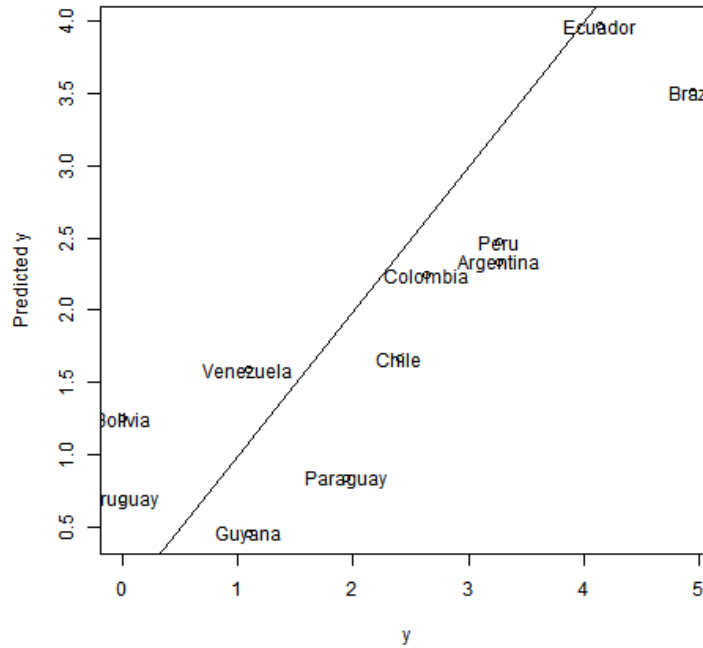


Figura 18: Y vs \hat{Y} CAR propio con ρ al 90% y $Y' = \log(Y+1)$, $E' = \log(E+1)$

5 Conclusiones

Utilizamos modelos espaciales para analizar la mortalidad por COVID-19 entre los países de América del Sur incluyendo el efecto de la migración. Particularmente, buscamos obtener estimaciones más suavizadas. Usamos modelos espaciales CAR e ICAR con variaciones en sus especificaciones. También, usamos un modelo jerárquico para comparar la bondad de los modelos espaciales, ya que este no ocupa la información espacial.

Se probó correr los modelos con las muertes observadas Y y las esperadas E , así como con dos transformaciones diferentes a los datos. Sin transformar, el ICAR no funcionó por un error en BUGS, mientras que el CAR propio y el jerárquico sí y sus cadenas convergieron. La primera transformación fue dividiendo las muertes observadas Y y las esperadas E por una constante, lo cual no funcionó ya que o BUGS no compilaba, o las cadenas no convergían. La segunda transformación fue $\log(Y + 1)$ y $\log(E + 1)$, la cual generó resultados mucho más suavizados que en el caso sin transformaciones. De esto que analizamos los resultados únicamente para Y , E y para $\log(Y + 1)$, $\log(E + 1)$

Dado que en un caso ocupamos la variable respuesta sin transformar y en el otro aplicamos logaritmo, los DIC no son comparables. Por ello analizamos cada caso independientemente y obtuvimos los dos mejores modelos de acuerdo al mínimo DIC.

Para ambos casos, el modelo de menor DIC fue el CAR propio con una especificación de ρ asociado al 90% del rango definido por los eigenvalores. Los modelos finales muestran características bastante favorables; comparten un buen comportamiento en sus cadenas, las cuales se mezclan y convergen correctamente. Ambos modelos generan buenas estimaciones con respecto a nuestros datos, tienen un $R^2 > 0.999$ para Y , E y $R^2 > 0.7$ $\log(Y + 1)$, $\log(E + 1)$, por lo que logran explicar nuestro modelo y además generar el suavizamiento deseado, esto particularmente para la transformación de logaritmo.

El mayor beneficio de ocupar estos modelos fue en la estimación de las λ_i , ya que para países con baja tasa, esta aumenta gracias al suavizamiento, y vice versa para los que tienen tasa alta. Esto es deseado, ya que pueden haber errores o picos no usuales en las observaciones que sesgan la información, y el suavizamiento logra generar tasas que quizá se acercan más a la realidad.

Adicionalmente, probamos un modelo jerárquico para contrastar la relevancia de la información espacial. Este modelo resulta tener una buena mezcla entre cadenas de simulación. Sin embargo, su DIC no es tan bueno como los modelos espaciales, por lo que podemos concluir que el efecto espacial sí está aportando información importante a los modelos.

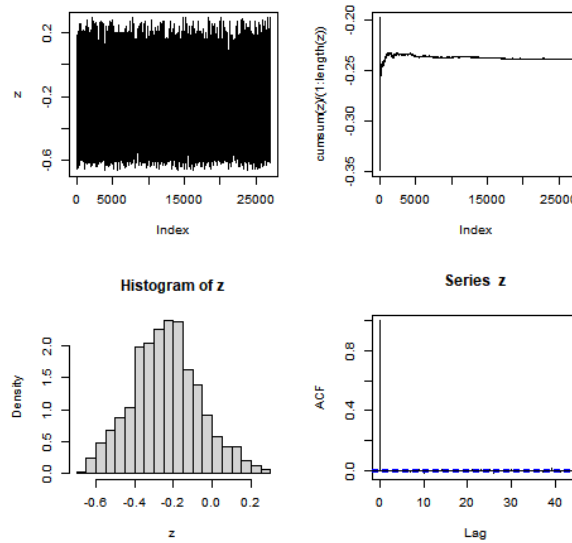
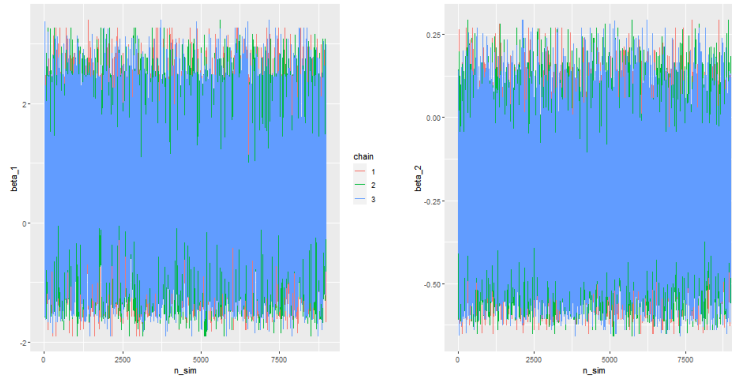
6 Referencias

- [1] Thomas, A., Best, N., Lunn, D., Arnold, R., & Spiegelhalter, D. (2004). *GeoBugs user manual..* Cambridge: Medical Research Council Biostatistics Unit.
- [2] CSSE at Johns Hopkins University, (2022). *CSSE Covid 19 Data*. Recuperado el 27 de mayo de 2022 de https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data
- [3] MARCUSINTHESKY, (2020). *COVID-19: Global Spatial Regression Model*. Recuperado el 27 de abril de 2022 de <https://www.kaggle.com/code/marcusgawronsky/covid-19-global-spatial-regression-model>.
- [4] Kaggle, (2020). *COVID19 Global Forecasting*.
- [5] Fernando Lasso, (2018). *Countries of the World*.
- [6] Nieto-Barajas, L. E., (2022). *Regresión Avanzada*. Modelos Espaciales. ITAM.
http://allman.rhon.itam.mx/~lnieto/index_archivos/NotasRA7.pdf

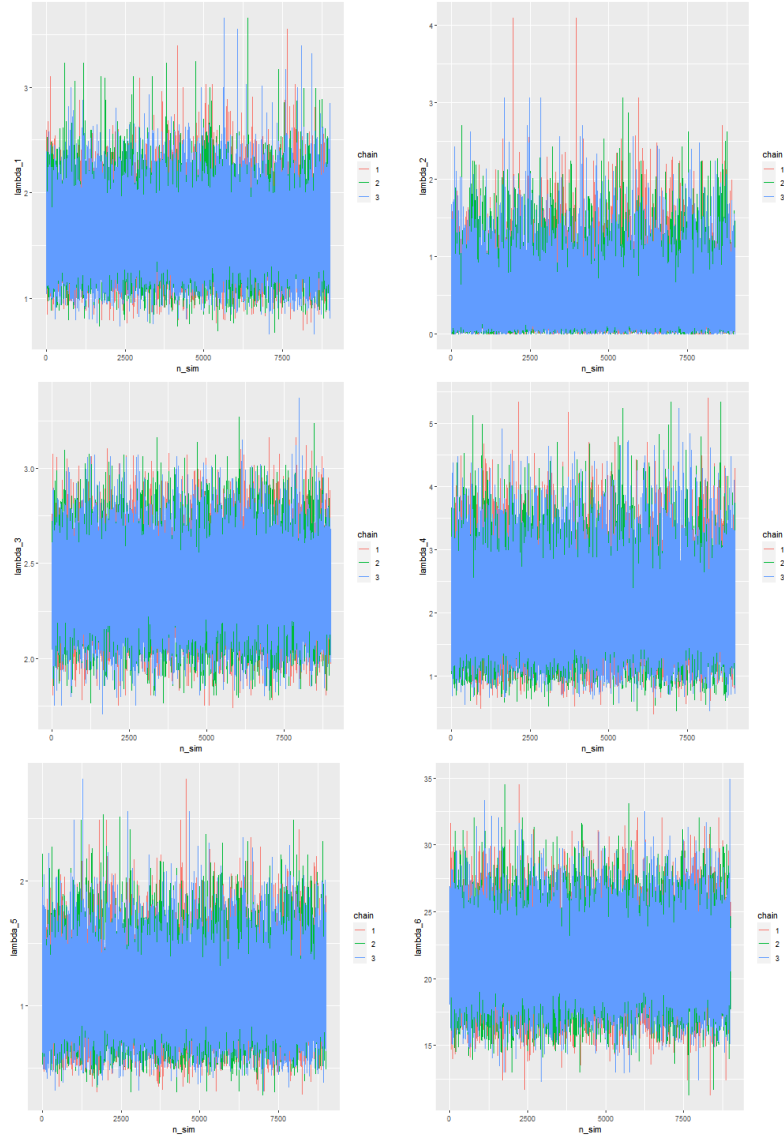
7 Apéndice

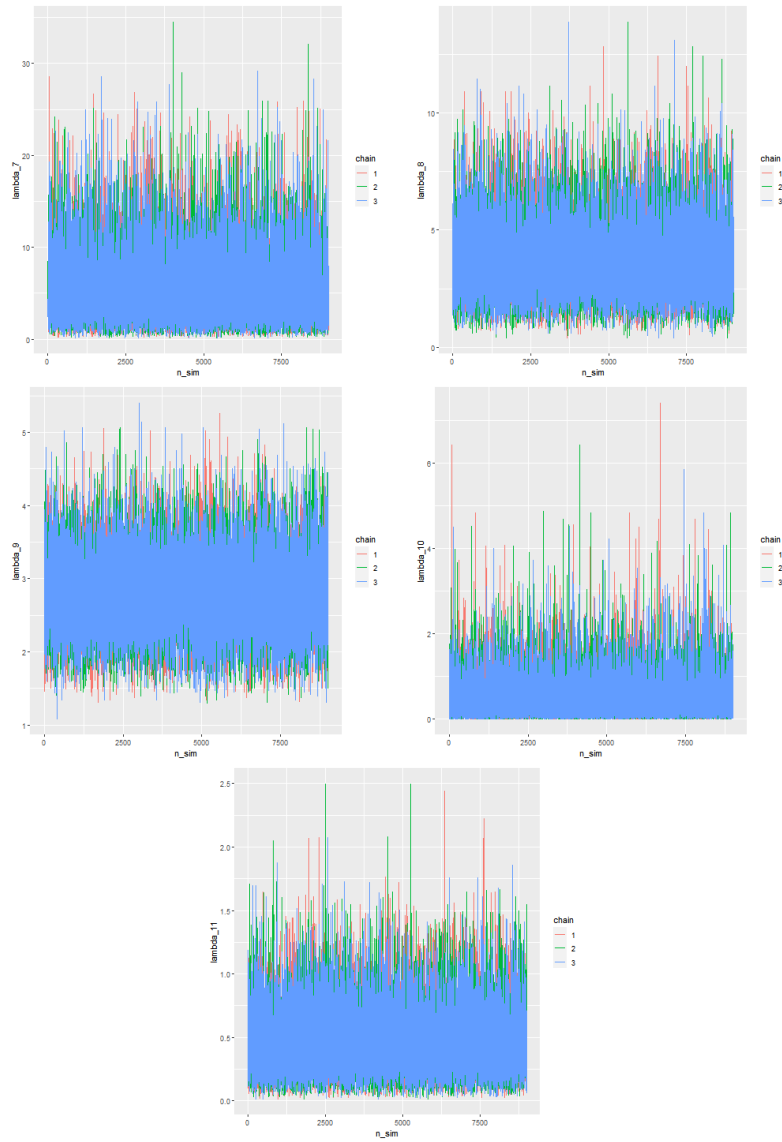
7.1 Gráficas para el modelo CAR propio con ρ al 90% sin transformaciones en Y , E

7.1.1 β 's:

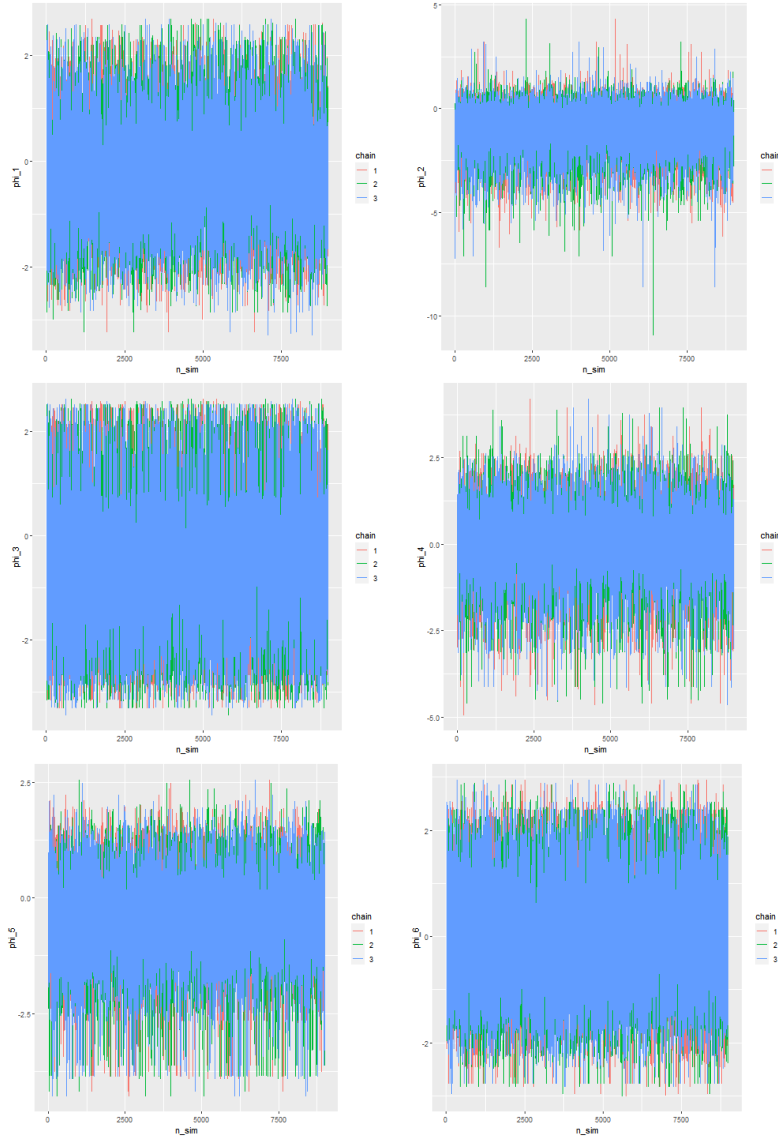


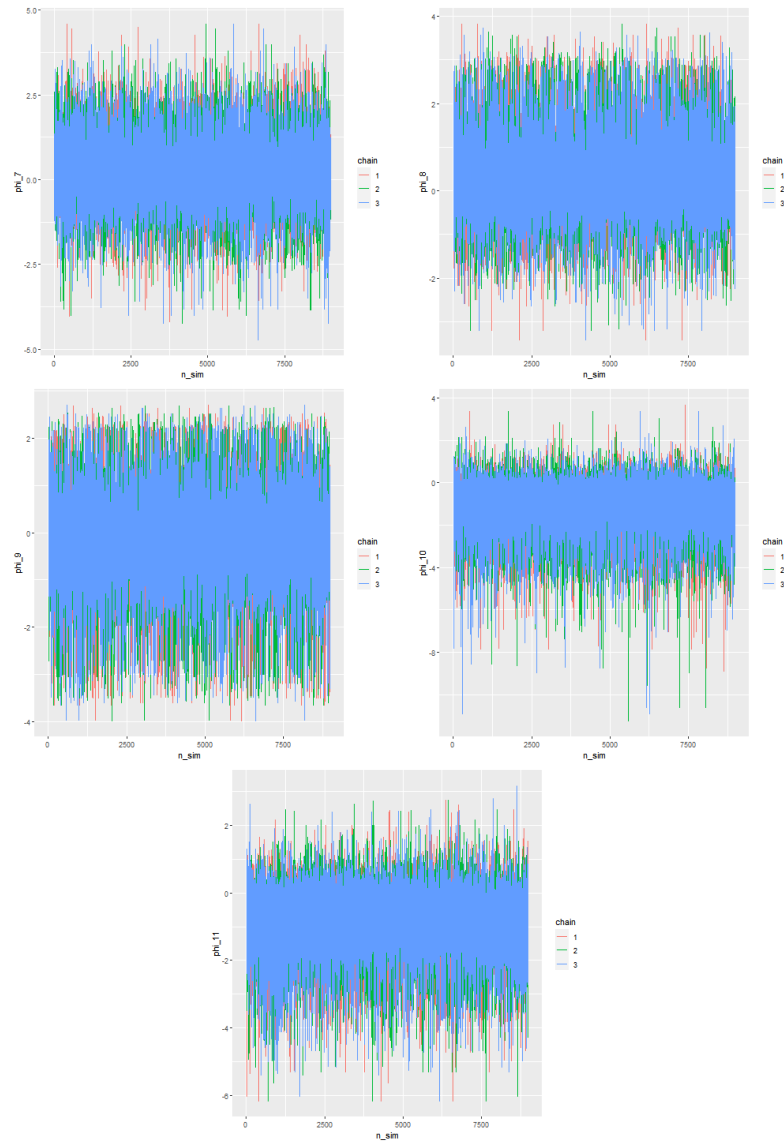
7.1.2 λ 's:



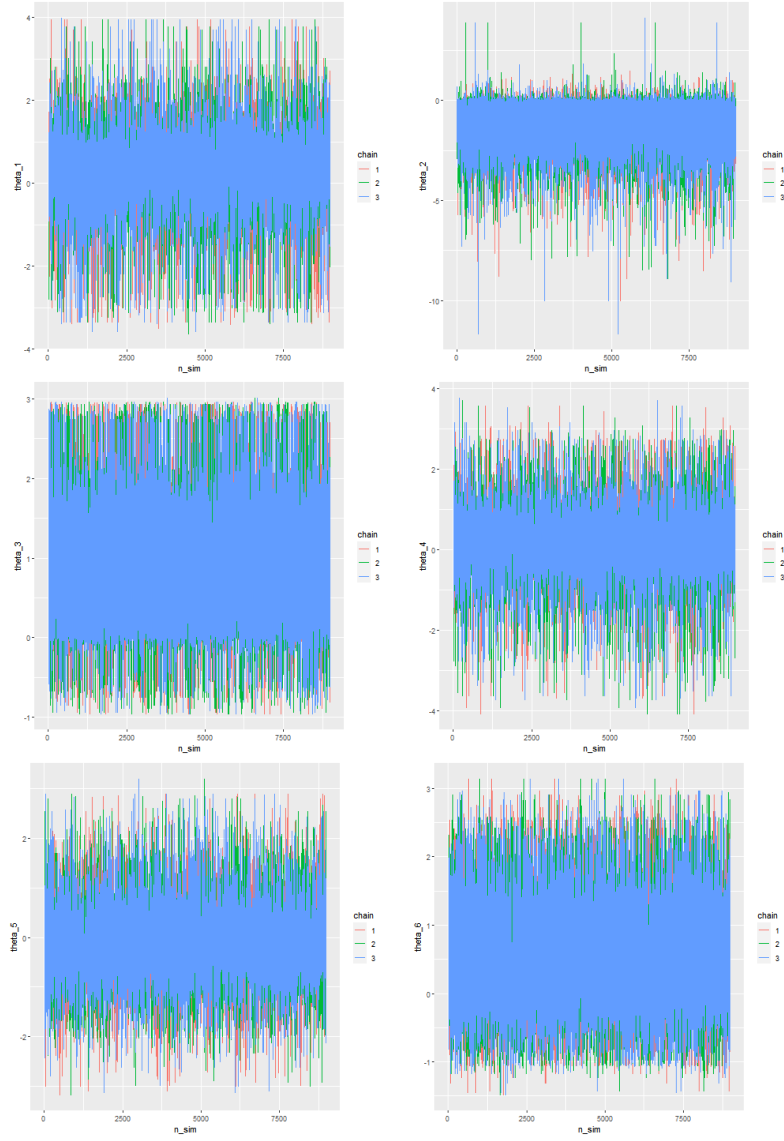


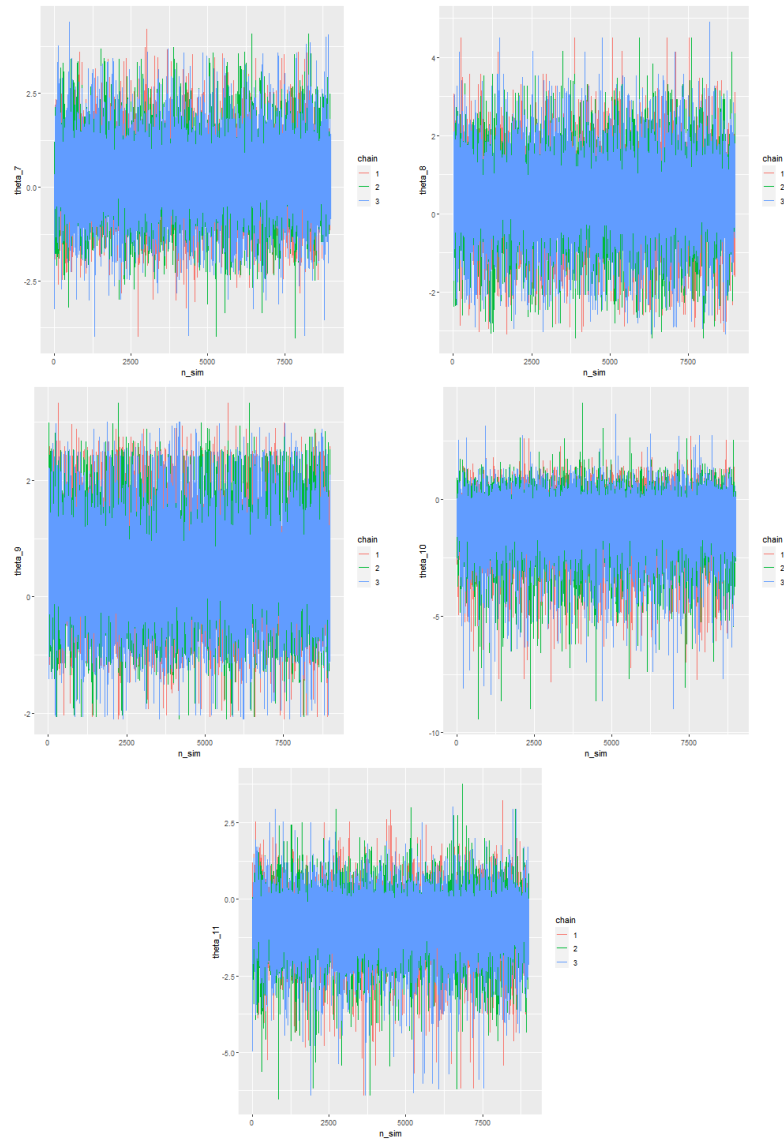
7.1.3 ϕ 's:



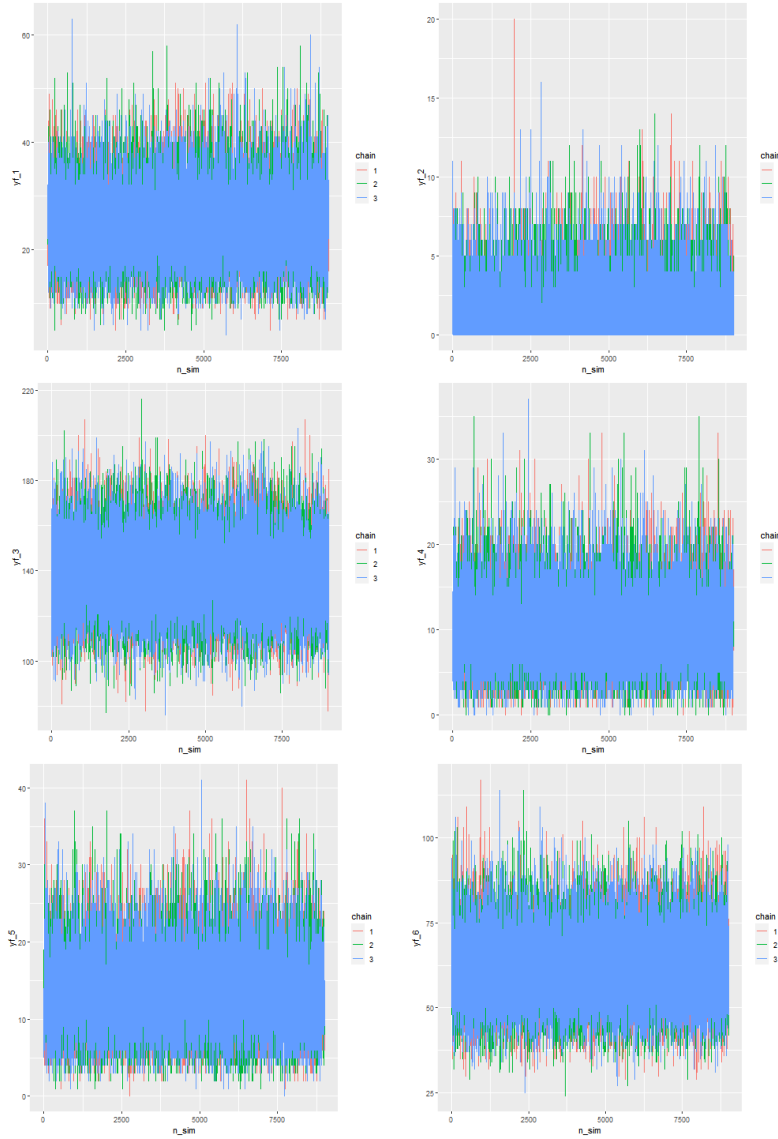


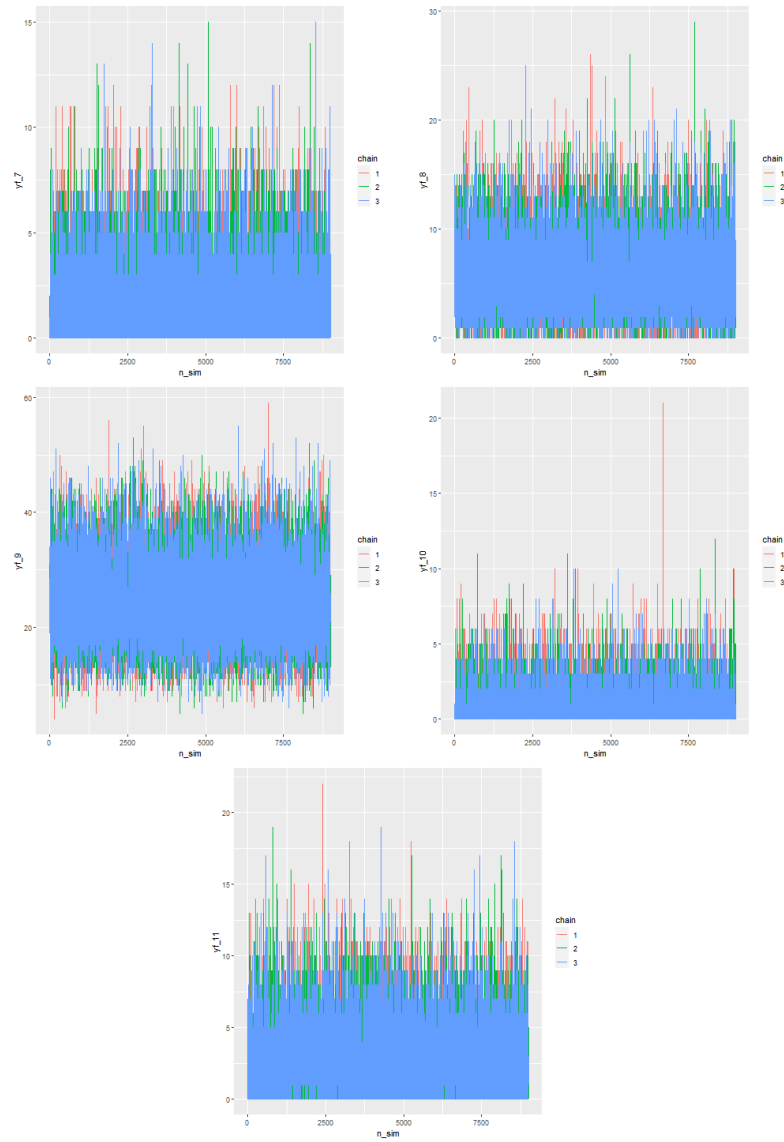
7.1.4 θ 's:





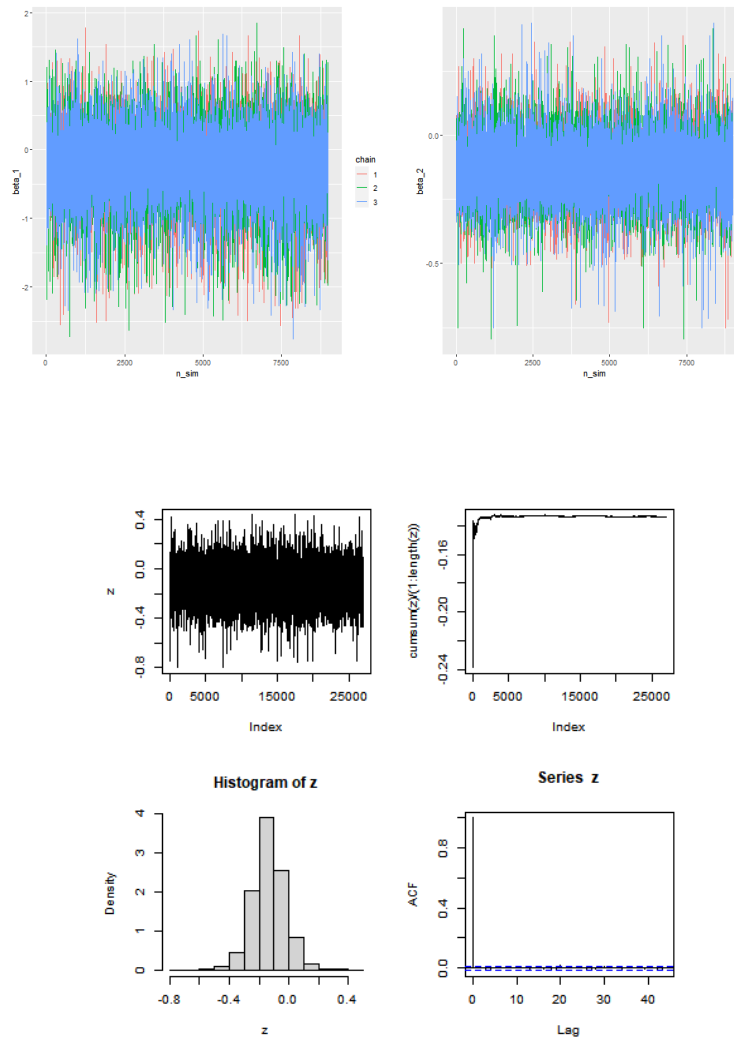
7.1.5 Predicciones, yf 's:



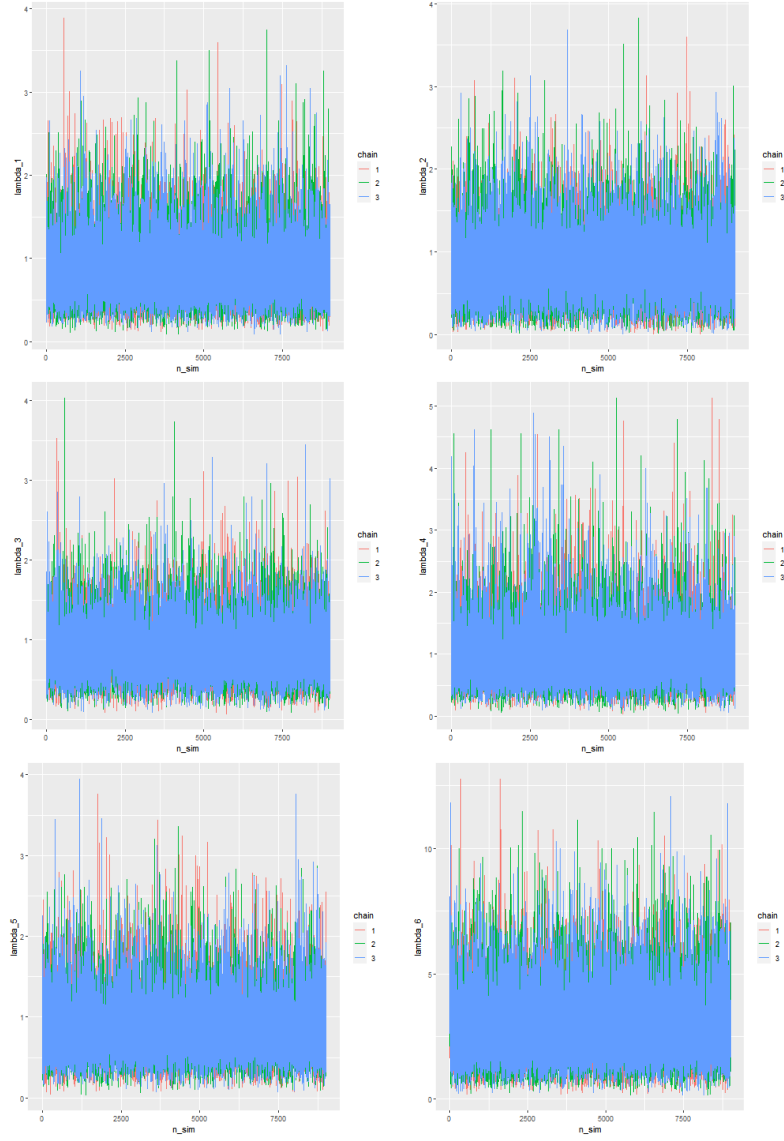


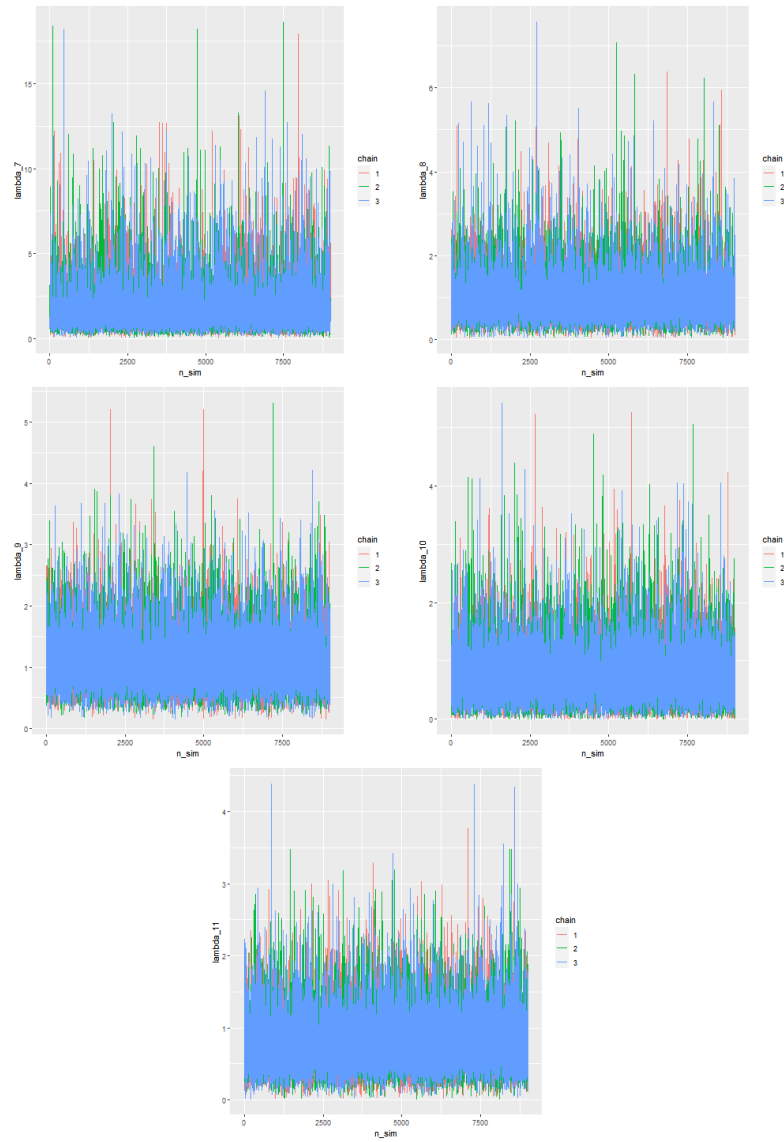
7.2 Gráficas para el modelo CAR propio con ρ al 90% sin transformaciones en $Y' = \log(Y + 1)$, $E' = \log(E + 1)$

7.2.1 β 's:

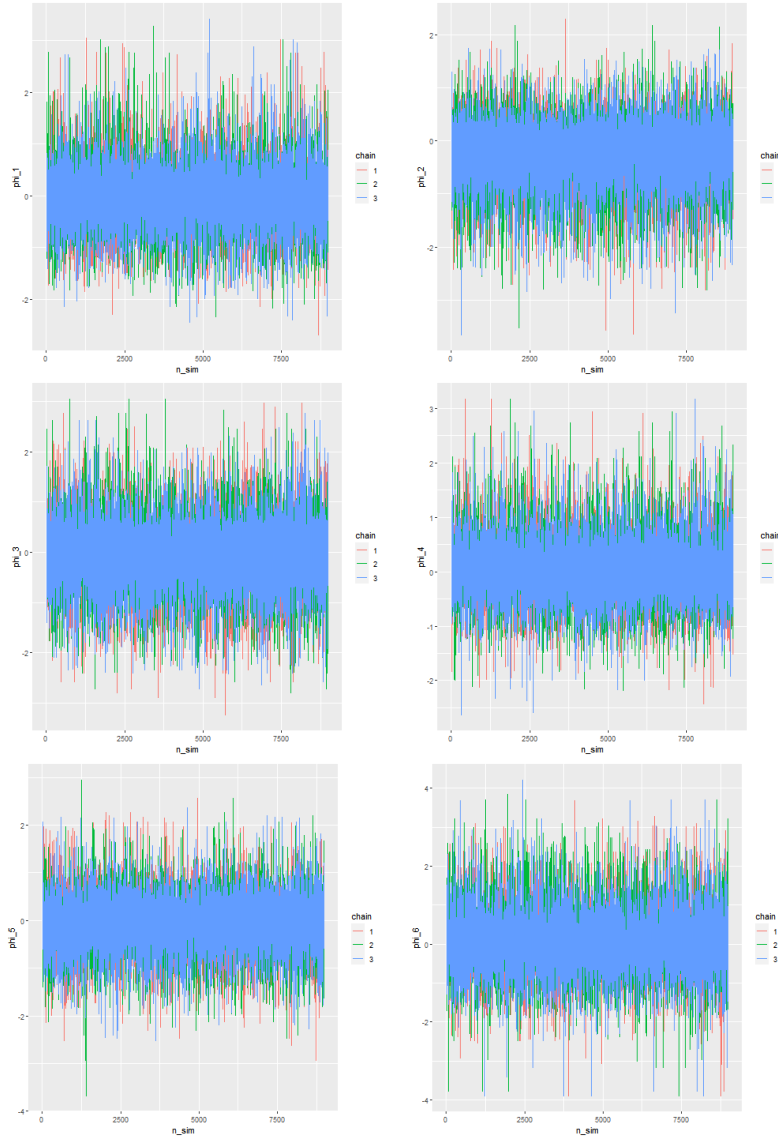


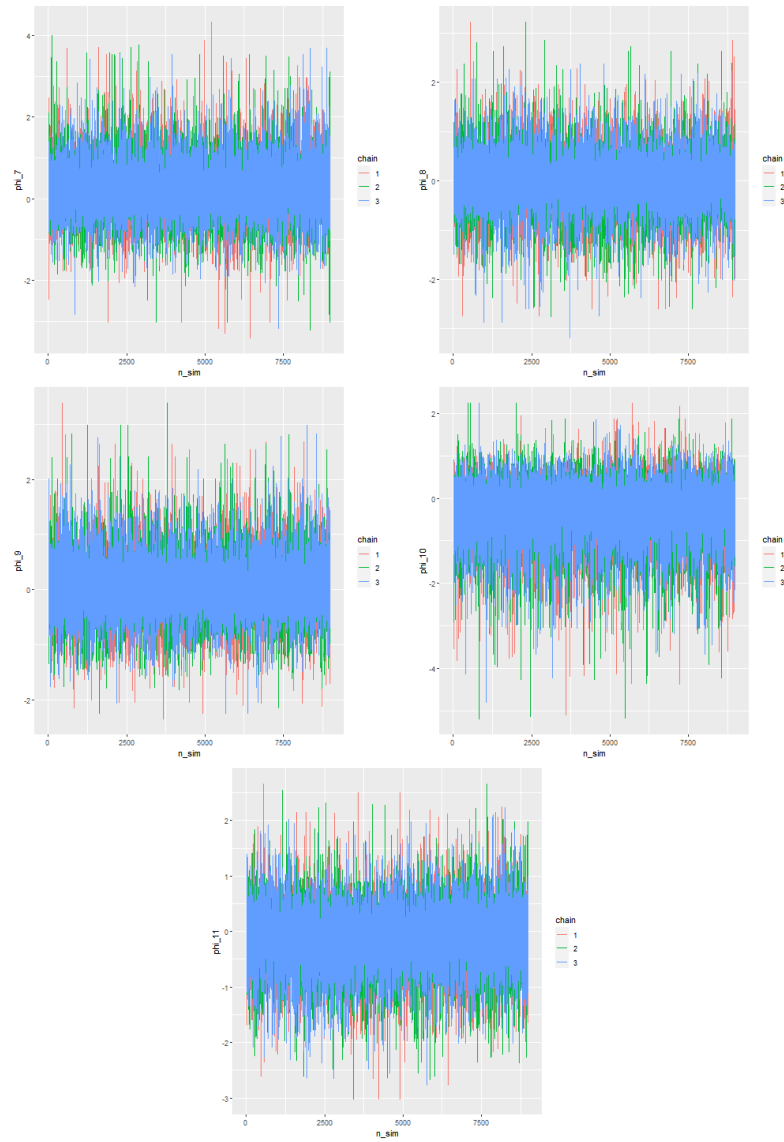
7.2.2 λ 's:



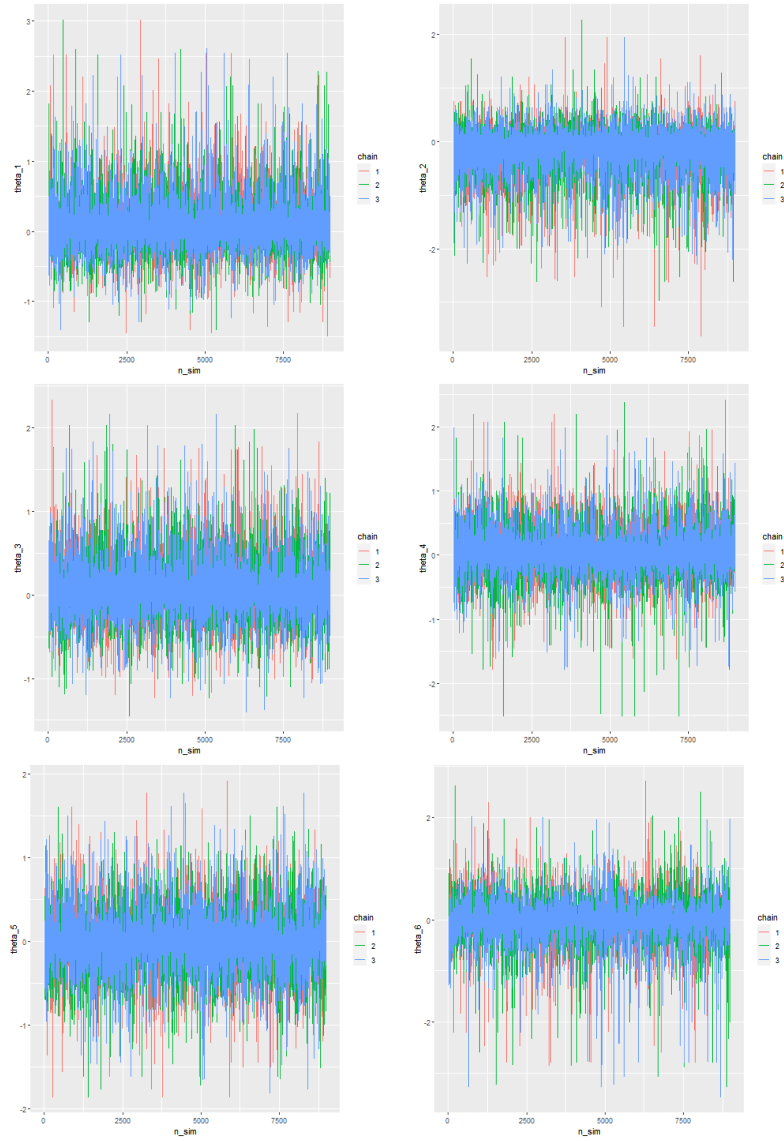


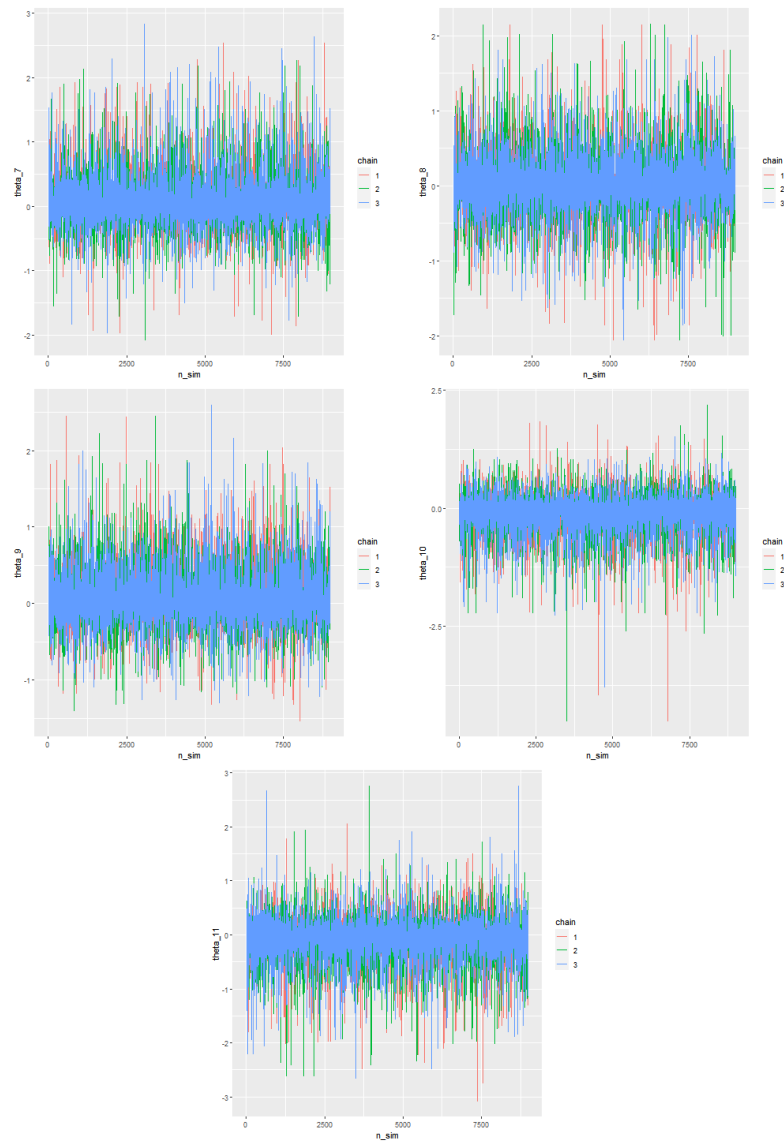
7.2.3 ϕ 's:



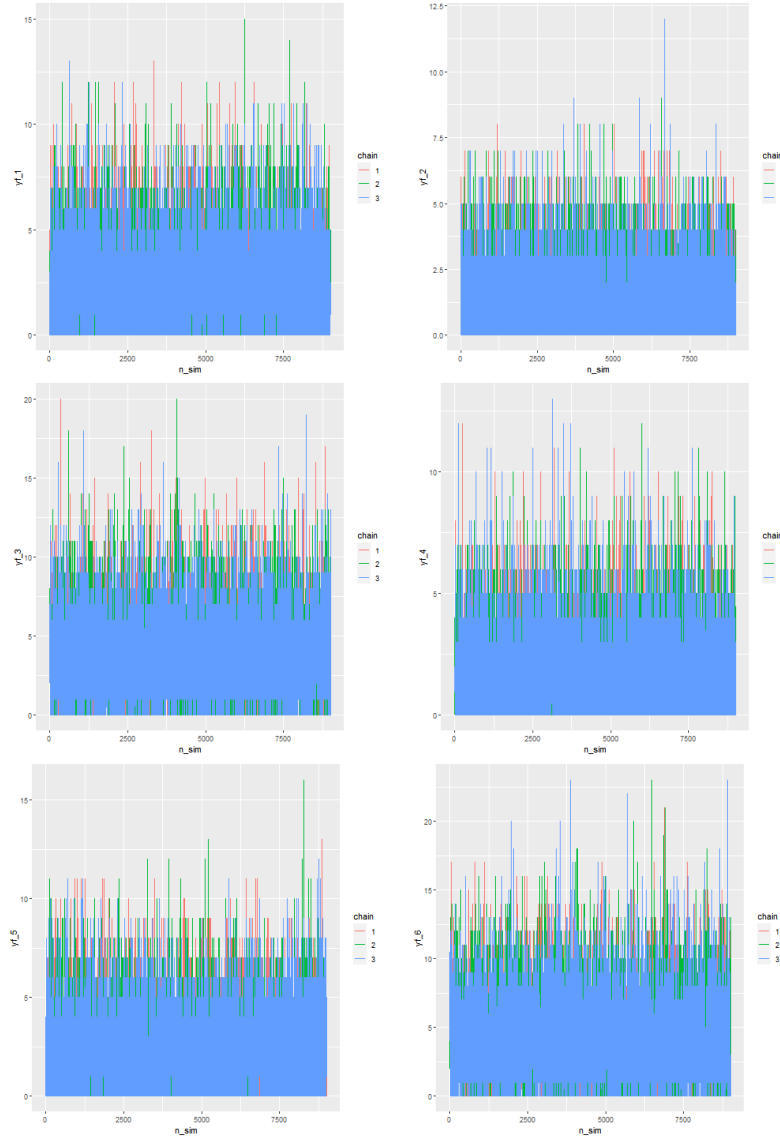


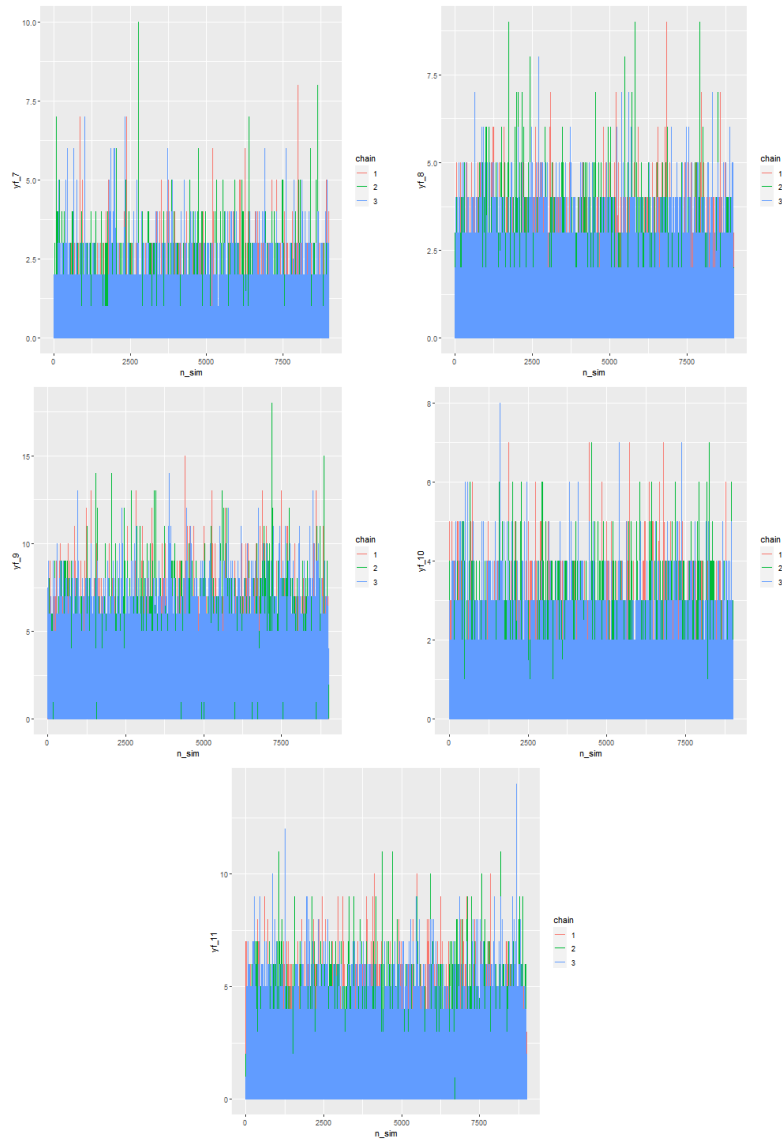
7.2.4 θ 's:





7.2.5 Predicciones, yf 's:





7.3 Código R Base

```

1 ##### Carga de librerias #####
2
3 # Proposito general
4 library(tidyverse)
5 library(ggplot2)
6
7 # Simuladores MCMC
8 library(R2OpenBUGS)
9 library(R2jags)
10
11 # Graficas espaciales
12 library(SpatialEpi)
13 library(maps)
14 library(mapttools)
15 library(RColorBrewer)
16 library(classInt)
17 library(spdep)
18 library(sp)
19
20 ##### Funciones auxiliares #####
21
22 prob <- function(x){
23   out<-min(length(x[x>0])/length(x),length(x[x<0])/length(x))
24   out
25 }
26
27 get_x <- function(value){(value |> strsplit(" "))[[1]][1]}
28 get_y <- function(value){(value |> strsplit(" "))[[1]][2]}
29
30 # La siguiente funcion nos ayuda a procesar las coordenadas
31 # que vienen en el csv, tal que la libreria sp los procese
32 # como poligonos
33 obtener_poligono <- function(pais){
34
35   # Filtramos el pais y sacamos el poligono como viene
36   poligonos <- datos |>
37     filter(Country == pais) |>
38     head(1) |>
39     pull(geometry) |>
40     strsplit("\\(")
41
42   ### Como hay algunos multipoligonos, queremos quedarnos con el
43   ### mas grande
44
45   # Primero vemos las longitudes de cada poligono
46   poligonos_length <- c()
47   for (poligono in poligonos[[1]]){
48     poligonos_length <- c(poligonos_length, poligono |> nchar())
49   }
50
51   # Al mayor le aplicamos un procesamiento para tener las
52   # coordenadas como las necesitamos
53   country_polygon <- poligonos[[1]][which.max(poligonos_length)]
54   country_polygon <- gsub('\\\\', '', country_polygon)
55   country_polygon <- country_polygon |> strsplit("\\\\,")

```

```

56 country_polygon <- country_polygon[[1]] |> as_tibble()
57 country_polygon <- country_polygon |>
58   mutate(value = ifelse(value |> substr(1, 1) == " ",
59                           value |> substr(2, 1000),
60                           value))
61
62 # Obtenemos las latitudes (x) y longitudes (y)
63 x <- country_polygon |> pull(value) |> sapply(get_x)
64 y <- country_polygon |> pull(value) |> sapply(get_y)
65
66 # Y las guardamos como numericos en una matriz
67 country_polygon <- country_polygon |>
68   mutate(x_coord = as.numeric(x), y_coord = as.numeric(y)) |>
69   select(-c("value")) |>
70   as.matrix()
71
72 # Creamos el objeto Polygon
73 p = Polygon(country_polygon |> as.matrix())
74 ps = Polygons(list(p), pais)
75
76 ps
77
78 }
79
80 plot_arrows <- function(region){
81
82   if (region == region){
83     arrows(-75, -39, -72, -38, length = 0.1) #Chile
84     arrows(-56, 8, -59, 5, length = 0.1) #Guyana
85     arrows(-81, 2, -78, -1, length = 0.1) #Ecuador
86     arrows(-75, -22.5, -59, -22.5, length = 0.1) #Paraguay
87     arrows(-52, -35, -56, -32, length = 0.1) #Uruguay
88   }
89
90 }
91
92 save_definition <- function(model_id, data, inits, parameters,
93                               covid.sim){
94
95   dir.create(paste0("../results/", model_id))
96
97   sink(paste0("../results/", model_id, "/data.txt"))
98   print(data)
99   sink()
100
101   sink(paste0("../results/", model_id, "/inits.txt"))
102   print(inits)
103   sink()
104
105   sink(paste0("../results/", model_id, "/parameters.txt"))
106   print(parameters)
107   sink()
108
109   sink(paste0("../results/", model_id, "/dic.txt"))
110   print(covid.sim$DIC)
111   sink()

```

```

112 save_mcmc_params(model_id, covid.sim)
113
114 save_traceplots(model_id, covid.sim)
115 }
116
117 save_traceplots <- function(model_id, covid.sim){
118
119   # chain <- c()
120   chain <- rep(1:covid.sim$n.chains, covid.sim$n.sims / covid.sim$n
121     .chains)
122   n_sim <- c()
123   for (ch in 1:covid.sim$n.chains){
124     # chain <- c(chain, rep(ch, covid.sim$n.sims / covid.sim$n.
125       chains))
126     n_sim <- c(n_sim, seq(1:(covid.sim$n.sims / covid.sim$n.chains)
127       ))
128   }
129
130   params_names <- grepl("deviance", names(covid.sim$sims.list))
131   params_names <- names(covid.sim$sims.list)[!params_names]
132
133   for (name in params_names){
134
135     if(length(covid.sim$sims.list[name][[1]]) < (covid.sim$n.sims +
136       1)){
137
138       print(paste0("Saving traceplot for ", name))
139       col_name <- paste0(name)
140
141       sims <- covid.sim$sims.list[name][[1]]
142       sim <- sims
143
144       p <- as_tibble(sim) |>
145         mutate(chain = factor(chain), n_sim = n_sim) |>
146         ggplot() +
147         geom_line(aes(n_sim, value, color = chain)) +
148         labs(y = col_name)
149
150       png(paste0("../results/", model_id, "/", col_name, "_tp.png")
151     )
152       print(p)
153       dev.off()
154
155     }
156
157     else{
158
159       for (k in 1:dim(covid.sim$sims.list[name][[1]])[2]){
160
161         print(paste0("Saving traceplot for ", name, "_",k))
162         col_name <- paste0(name, "_",k)
163
164         sims <- covid.sim$sims.list[name][[1]]
165         sim <- sims[,k]
166
167         p <- as_tibble(sim) |>

```



```

164     mutate(chain = factor(chain), n_sim = n_sim) |>
165     ggplot() +
166     geom_line(aes(n_sim, value, color = chain)) +
167     labs(y = col_name)
168
169     png(paste0("../results/", model_id, "/", col_name, "_tp.png
170         "))
171     print(p)
172     dev.off()
173   }
174
175   }
176
177 }
178
179 }
180
181 save_mcmc_params <- function(model_id, covid.sim){
182
183
184   mcmc_params <- list("n.chains" = covid.sim$n.chains, "n.iter" =
185     covid.sim$n.iter,
186     "n.burnin" = covid.sim$n.burnin, "n.thin" =
187     covid.sim$n.thin,
188     "n.keep" = covid.sim$n.keep, "n.sims" = covid.
189     sim$n.sims)
190
191   sink(paste0("../results/", model_id, "/mcmc_params.txt"))
192   print(mcmc_params)
193   sink()
194 }
195
196 ##### Procesamiento inicial de los datos #####
197
198 ### Leemos los datos, filtramos por region, calculamos las
199 ### metricas de interes, y realizamos cualquier proceso
200 ### adicional segun la region.
201
202 # Carga de datos
203 datos <- read.csv("../data/covid.csv")
204
205 # Filtramos la region de interes
206 region <- "South America"
207 datos <- datos |> filter(continent == region)
208
209 # Variable auxiliar para desfazar las defunciones
210 aux <- datos |>
211   select(c(Fatalities)) |>
212   lag() |>
213   pull(Fatalities)
214
215 # Obtenemos defunciones esperadas
216 # !!! Aqui el 5% es un supuesto muy importante !!!
217 covid <- datos |>

```

```

217 mutate(total_deaths = Population * Deathrate / 1000000) |>
218 mutate(expected_covid_deaths = (5 / 100 * total_deaths))
219
220 # Calculamos defunciones diarias observadas
221 # !!! Aqui lo del -1 puede que genere malos datos si
222 # el ultimo del pais anterior fue menor al primero del pais
223 # que le sigue
224 covid <- covid |>
225   mutate(Fatalities_per_day = covid |> pull(Fatalities) - aux) |>
226   filter(Fatalities_per_day > -1)
227
228 # Calculamos las medias de las metricas
229 covid <- covid |>
230   select(c("Country", "Fatalities_per_day",
231            "expected_covid_deaths", "Net.migration")) |>
232   group_by(Country) |>
233   summarise(mean_fatalities = mean(Fatalities_per_day),
234             mean_deathrate = mean(expected_covid_deaths),
235             mean_net_migration = mean(Net.migration)) |>
236   mutate(mean_fatalities = round(mean_fatalities, 0) |> as.integer
237         ())
238
239 # Procesamiento adicional segun la region
240 if (region == "South America"){
241   covid <- covid |> filter(Country != "Suriname")
242 }
243
244 #####
245 # covid <- covid |> filter(Country != "Brazil")
246 # covid <- covid |> filter(Country != "Bolivia")
247 #####
248
249 # Conjunto de datos final
250 covid
251
252 ##### Coordenadas #####
253
254 # Definimos la lista en la que guardaremos los poligonos
255 lista_poligonos <- list()
256 lista_coords_x <- list()
257 lista_coords_y <- list()
258
259 # Guardamos los poligonos en la lista que definimos
260 for (pais in covid |> pull(Country)){
261   lista_poligonos[pais] <- obtener_poligono(pais)
262   lista_coords_x[pais] <- (lista_poligonos[pais][[1]]@Polygons[[1]]
263     @coords |> colMeans())[1]
264   lista_coords_y[pais] <- (lista_poligonos[pais][[1]]@Polygons[[1]]
265     @coords |> colMeans())[2]
266 }
267
268 # Y creamos el objeto SpatialPolygons
269 sps = SpatialPolygons(lista_poligonos)
270 plot(sps)
271
272 # Guardamos las coordenadas centrales de cada pais

```

```

271 mean_coords <- tibble(
272   Country = covid$Country,
273   lista_coords_x |> as_tibble() |> t() |> as_tibble() |> rename(x
    = V1),
274   lista_coords_y |> as_tibble() |> t() |> as_tibble() |> rename(y
    = V1)
275 )
276
277 # Pequeños ajustes para que no se encimen
278 mean_coords <- mean_coords |>
279   mutate(x = replace(x, Country == "Argentina", -64)) |>
280   mutate(y = replace(y, Country == "Argentina", -32)) |>
281   mutate(x = replace(x, Country == "Brazil", -52)) |>
282   mutate(x = replace(x, Country == "Chile", -78)) |>
283   mutate(y = replace(y, Country == "Chile", -39)) |>
284   mutate(x = replace(x, Country == "Colombia", -74)) |>
285   mutate(x = replace(x, Country == "Ecuador", -86)) |>
286   mutate(y = replace(y, Country == "Ecuador", 4)) |>
287   mutate(x = replace(x, Country == "Guyana", -54)) |>
288   mutate(y = replace(y, Country == "Guyana", 9)) |>
289   mutate(x = replace(x, Country == "Paraguay", -78)) |>
290   mutate(y = replace(y, Country == "Paraguay", -21)) |>
291   mutate(x = replace(x, Country == "Peru", -76)) |>
292   mutate(x = replace(x, Country == "Uruguay", -50)) |>
293   mutate(y = replace(y, Country == "Uruguay", -36)) |>
294   mutate(y = replace(y, Country == "Venezuela", 9)) |>
295   mutate(x = replace(x, Country == "Venezuela", -66))
296
297 ##### Graficas de las coordenadas #####
298
299 # Procesamiento adicional segun la region
300 if (region == "South America"){
301   x_etiquetas <- -40
302   y_etiquetas <- -15
303   nclr <- 4
304 }
305
306 plot_arrows <- function(region){
307   if (region == region){
308     arrows(-75, -39, -72, -38, length = 0.1) #Chile
309     arrows(-56, 8, -59, 5, length = 0.1) #Guyana
310     arrows(-81, 2, -78, -1, length = 0.1) #Ecuador
311     arrows(-75, -22.5, -59, -22.5, length = 0.1) #Paraguay
312     arrows(-52, -35, -56, -32, length = 0.1) #Uruguay
313   }
314 }
315
316 }
317
318 #### SMR (standarized morbidity rate)
319 # En este caso, analizamos lambda = observados / esperados
320 {
321   plotvar <- covid$mean_fatalities/covid$mean_deathrate
322   plotvar <- ifelse(plotvar == Inf, 0, plotvar)
323   plotclr <- brewer.pal(nclr,"YlOrBr")
324   class <- classIntervals(plotvar, nclr, dataPrecision=2,style="
325

```

```

    quantile")
326 colcode <- findColours(class,plotclr)
327
328 covid.map <- sps
329
330 png(paste0("../results/smr.png"))
331 plot(covid.map, col = colcode)
332 legend(x_etiquetas, y_etiquetas, legend = names(attr(colcode, "
    table")),
333       fill = attr(colcode, "palette"), cex=1, bty="n")
334 text(mean_coords$x, mean_coords$y, mean_coords$Country)
335 text(mean_coords$x, mean_coords$y-3, plotvar |> round(2))
336 plot_arrows(region)
337 title(main="SMR")
338 dev.off()
339 }
340
341 ##### SMR (standarized morbidity rate)
342 # En este caso, analizamos lambda = log(observados) / log(esperados
    )
343 {
344   plotvar <- log(covid$mean_fatalities+1)/log(covid$mean_deathrate
    +1)
345   plotvar <- ifelse(plotvar == Inf, 0, plotvar)
346   plotclr <- brewer.pal(nclr,"YlOrBr")
347   class <- classIntervals(plotvar, nclr, dataPrecision=2,style="
    quantile")
348   colcode <- findColours(class,plotclr)
349
350   covid.map <- sps
351
352   png(paste0("../results/smr_log_log.png"))
353   plot(covid.map, col = colcode)
354   legend(x_etiquetas, y_etiquetas, legend = names(attr(colcode, "
    table")),
355         fill = attr(colcode, "palette"), cex=1, bty="n")
356   text(mean_coords$x, mean_coords$y, mean_coords$Country)
357   text(mean_coords$x, mean_coords$y-3, plotvar |> round(2))
358   plot_arrows(region)
359   title(main="SMR")
360   dev.off()
361 }
362
363 ##### Net Migration
364 {
365   plotvar <- covid$mean_net_migration
366   plotclr <- brewer.pal(nclr,"YlOrBr")
367   class <- classIntervals(plotvar, nclr, dataPrecision=2,style="
    quantile")
368   colcode <- findColours(class,plotclr)
369
370   covid.map <- sps
371
372   png(paste0("../results/net_migration.png"))
373   plot(covid.map, col = colcode)
374   legend(x_etiquetas, y_etiquetas, legend = names(attr(colcode, "
    table")),

```

```

375     fill = attr(colcode, "palette"), cex=1, bty="n")
376     text(mean_coords$x, mean_coords$y, mean_coords$Country)
377     text(mean_coords$x, mean_coords$y-3, plotvar |> round(2))
378     plot_arrows(region)
379     title(main="Net Migration")
380     dev.off()
381 }
382
383
384
385 ##### Matrices de adyacencia #####
386
387 n <- nrow(covid)
388
389 W.nb <- poly2nb(sps)
390 print(W.nb)
391 W.nb_list <- head(W.nb, 100)
392 max_n_neighbours <- 0
393 for (pais in W.nb_list){
394     if (max_n_neighbours < length(pais)){
395         max_n_neighbours <- length(pais)
396     }
397 }
398 m <- rep(0,n)
399 W.l <- matrix(NA, nrow=n , ncol=max_n_neighbours )
400 adj <- NULL
401
402 for (i in 1:n){
403     if (W.nb[[i]][1]!=0){
404         m[i] <- length(W.nb[[i]])
405         W.l[i,1:m[i]] <- W.nb[[i]]
406         adj <- c(adj,W.nb[[i]])
407     }
408 }
409
410 W <- matrix(0, nrow = n, ncol = n)
411
412 for (i in 1:n){
413     for (j in 1:m[i]){
414         W[i,W.l[i,j]] <- 1
415         W[W.l[i,j],i] <- 1
416     }
417 }
418
419 weights <- rep(1,length(adj))
420
421 ##### Parametros CAR Propio #####
422
423 ##### Verificacion de eigenvalores #####
424 #
425 # El calculo del los eigenvalores para definir el rango de
426 # rho difiere entre las notas de clase y el manual de OpenBugs.
427 # Se observo que esto se debe a que las matrices D (en Bugs es M) y
428 # W (en Bugs es C) difieren en la definicion.
429 #
430 # Esta subseccion tiene el proposito de verificar que ambas formas
431 de

```

```

431 # calcular los eigenvalores sea la misma.
432 #
433 # Nota: Para calcular ambas se requiere que NO haya regiones
      aisladas.
434 #
435 # https://www.mrc-bsu.cam.ac.uk/wp-content/uploads/geobugs12manual.
      pdf
436 # https://www.multibugs.org/documentation/latest/spatial/
      SpatialDistributions.html#Conditional
437
438 ##### Notas de Clase #####
439
440 D_neg_sqr <- diag(m^(-1/2))
441
442 B <- D_neg_sqr %*% W %*% D_neg_sqr
443 eigenvalues_clase <- eigen(B)$values
444 max(eigenvalues_clase)
445 min(eigenvalues_clase)
446
447 ##### OpenBugsManual #####
448
449 D_sqr <- diag((1/m)^(1/2))
450 D_neg_sqr <- diag((1/m)^(-1/2))
451
452 W_OB <- matrix(W, nrow = n)
453 for (k in 1:n){
454   W_OB[k,] <- W_OB[k,]/m[k]
455 }
456
457 B <- D_neg_sqr %*% W_OB %*% D_sqr
458 # B <- D_neg_sqr %*% W_OB %*% D_neg_sqr
459
460 eigenvalues_bugs <- eigen(B)$values
461 max(eigenvalues_bugs)
462 min(eigenvalues_bugs)
463
464 ##### Comparacion #####
465
466 # La suma debe ser cercana a 0
467 sum((eigenvalues_clase - eigenvalues_bugs)^2)
468
469 ##### Definicion de parametros #####
470 #
471 # Una vez verificado que los eigenvalores sean los mismos,
472 # ocupamos la version de Bugs para que corra.
473 #
474
475 mu_areas <- rep(0, n)
476
477 M <- 1/m
478 C <- matrix(W_OB)
479 C <- C[C != 0]
480
481 ### rho puede estar en el rango (1/min_eigen, 1/max_eigen)
482 max_eigen <- max(eigenvalues_bugs)
483 min_eigen <- min(eigenvalues_bugs)
484 rho_weight <- 90 # Elegir del 0 al 100

```

```

485 rho <- (rho_weight/100) * (1 / max_eigen) + ((100-rho_weight)/100)
      * (1 / min_eigen)
486
487 ##### Asignacion de variables para modelo #####
488 #
489 # !!! Importante: hay que elegir solo alguno de los siguientes
490 # !!! chunks para definir y, ee
491 #
492
493 {
494   y <- covid$mean_fatalities
495   ee <- covid$mean_deathrate
496 }
497
498 {
499   y <- covid$mean_fatalities/1000
500   ee <- covid$mean_deathrate/1000
501 }
502
503 {
504   y <- log(covid$mean_fatalities + 1)
505   ee <- log(covid$mean_deathrate + 1)
506 }
507
508 x <- covid$mean_net_migration
509
510 x <- (covid$mean_net_migration |> scale(T,T))[,1]
511
512 #-Defining data-
513
514 data_normal <- list("n" = n, "y" = y, "ee" = ee, "x" = x,
515                    "adj" = adj, "weights" = weights, "num" = m)
516
517 data_proper <- list("n" = n, "y" = y, "ee" = ee, "x" = x,
518                   "adj" = adj, "num" = m, "mu_areas" = mu_areas,
519                   "C"=C, "M"=M, "gamma"=rho)
520
521 data_proper_nox <- list("n" = n, "y" = y, "ee" = ee,
522                        "adj" = adj, "num" = m, "mu_areas" = mu_areas,
523                        "C"=C, "M"=M, "gamma"=rho)
524
525 data_proper_rho_prior <- c("n" = n, "y" = y, "ee" = ee, "x" = x,
526                           "adj" = adj, "num" = m, "mu_areas" = mu_
527                           areas,
528                           "C"=C, "M"=M, #quitamos gamma
529                           "min_ie"=(1 / min_eigen), "max_ie"=(1 /
530                           max_eigen))
531 # data_proper_rho_prior <- c(data_proper[-length(data_proper)]) #
532 # quitamos gamma
533
534 data_hierarchical <- list("n" = n, "y" = y, "ee" = ee, "x" = x)
535
536 #-Defining inits-
537
538 inits <- function(){list(beta = rep(0,2),
539                          tau.t = 1,
540                          tau.c = 1,

```

```

538         theta = rep(0,n),
539         phi = rep(0,n),
540         yf = rep(0,n))}
541
542 inits_proper_nox <- function(){list(beta = 0,
543         tau.t = 1,
544         tau.c = 1,
545         theta = rep(0,n),
546         phi = rep(0,n),
547         yf = rep(0,n))}
548
549 inits_proper_rho_prior <- function(){list(beta = rep(0,2),
550         tau.t = 1,
551         tau.c = 1,
552         gamma.c = 0.85,
553         theta = rep(0,n),
554         phi = rep(0,n),
555         yf = rep(0,n))}
556
557 inits_normal_normal <- function(){list(beta = rep(0,2),
558         tau.y = 1,
559         tau.t = 1,
560         tau.c = 1,
561         theta = rep(0,n),
562         phi = rep(0,n),
563         yf = rep(0,n))}
564
565 inits_hierarchical <- function(){list(beta = rep(0,2),
566         mu.t = 0,
567         tau.t = 1,
568         theta = rep(0,n),
569         yf = rep(0,n))}
570
571 #-Selecting parameters to monitor-
572
573 parameters <- c("beta", "lambda", "theta", "phi", "yf")
574
575 parameters_hierarchical <- c("beta", "lambda", "theta", "yf")
576
577 #-Running code-
578
579 covid.sim_normal <- bugs(data_normal, inits, parameters, model.file
580     = "covid_car_normal.txt",
581     n.iter = 10000, n.chains = 3, n.burnin =
582     1000, n.thin = 1,
583     debug = T)
584
585 covid.sim_proper <- bugs(data_proper, inits, parameters, model.file
586     = "covid_car_proper.txt",
587     n.iter = 10000, n.chains = 3, n.burnin =
588     1000, n.thin = 1,
589     debug = T)
590
591 covid.sim_proper_nox <- bugs(data_proper_nox, inits_proper_nox,
592     parameters, model.file = "covid_car_proper_nox.txt",
593     n.iter = 10000, n.chains = 3, n.burnin =
594     1000, n.thin = 1,
595     debug = T)

```



```

589                                     debug = T)
590 # Este no corre
591 covid.sim_proper_rho_prior <- bugs(data_proper_rho_prior, inits_
    proper_rho_prior, parameters,
592                                     model.file = "covid_car_proper_
    rho_prior.txt",
593                                     n.iter = 10000, n.chains = 3, n.
    burnin = 1000, n.thin = 1,
594                                     debug = T)
595
596 covid.sim_normal_normal <- bugs(data_normal, inits_normal_normal,
    parameters,
597                                     model.file = "covid_car_normal_
    normal.txt",
598                                     n.iter = 10000, n.chains = 3, n.
    burnin = 1000, n.thin = 1,
599                                     debug = T)
600
601 covid.sim_proper_normal <- bugs(data_proper, inits_normal_normal,
    parameters,
602                                     model.file = "covid_car_proper_
    normal.txt",
603                                     n.iter = 10000, n.chains = 3, n.
    burnin = 1000, n.thin = 1,
604                                     debug = T)
605
606 covid.sim_hierarchical <- bugs(data_hierarchical, inits_
    hierarchical, parameters_hierarchical,
607                                     model.file = "covid_hierarchical.
    txt",
608                                     n.iter = 10000, n.chains = 3, n.
    burnin = 1000, n.thin = 1,
609                                     debug = T)
610
611 ##### Identificacion del modelo #####
612 #
613 # Elegimos un identificador para guardar la corrida actual
614
615 model_id <- "car_proper_rho_90_nox"
616
617 ##### !!! Importante !!! #####
618 #
619 # Hay que correr solo 1 de los siguientes chunks, dependiendo
620 # de cual es el modelo que se acaba de correr.
621 #
622 # Tambien de preferenci hay que elegir un nombre de identificador
623 # diferente si es que cambiaron varias cosas.
624
625 ### Modelo CAR Intrinseco
626 {
627   covid.sim <- covid.sim_normal
628   save_definition(model_id, data_normal, inits, parameters, covid.
    sim)
629 }
630
631 ### Modelo CAR Propio
632 {

```

```

633 covid.sim <- covid.sim_proper
634 save_definition(model_id, data_proper, inits, parameters, covid.
    sim)
635 }
636
637 ### Modelo CAR Propio (Sin x)
638 {
639   covid.sim <- covid.sim_proper_nox
640   save_definition(model_id, data_proper_nox, inits_proper_nox,
    parameters, covid.sim)
641 }
642
643 ### Modelo CAR Intrinseco con log(y) Normal
644 {
645   covid.sim <- covid.sim_normal_normal
646   save_definition(model_id, data_normal, inits_normal_normal,
    parameters, covid.sim)
647 }
648
649 ### Modelo CAR Propio con log(y) Normal
650 {
651   covid.sim <- covid.sim_proper_normal
652   save_definition(model_id, data_proper, inits_normal_normal,
    parameters, covid.sim)
653 }
654
655 ### Modelo Hierarchical
656 {
657   covid.sim <- covid.sim_hierarchical
658   save_definition(model_id, data_hierarchical, inits_hierarchical,
    parameters_hierarchical, covid.sim)
659 }
660
661 ##### Verificacion del modelo #####
662
663 #Traza de la cadena
664 # traceplot(covid.sim)
665
666 #Cadena y resumen
667 {
668   out <- covid.sim$sims.list
669   out.sum <- covid.sim$summary
670 }
671
672 #DIC
673 {
674   out.dic <- covid.sim$DIC
675   print(out.dic)
676 }
677
678 #Beta
679 {
680   z <- out$beta[,2]
681   png(paste0("../results/", model_id, "/beta_2_summary.png"))
682   par(mfrow=c(2,2))
683   plot(z, type="l")
684   plot(cumsum(z)/(1:length(z)), type="l")
685 }

```

```

686 hist(z,freq=FALSE)
687 acf(z)
688 dev.off()
689 }
690
691
692 #Tabla resumen
693 {
694   out.b <- out.sum[grep("beta",rownames(out.sum)),c(1,3,7)]
695   out.b <- cbind(out.b,apply(out$beta,2,prob))
696   dimnames(out.b)[[2]][4] <- "prob"
697   print(out.b)
698   sink(paste0("../results/", model_id, "/beta_summary.txt"))
699   print(out.b)
700   sink()
701 }
702
703 #Predictions
704 {
705   out.yf <- out.sum[grep("yf",rownames(out.sum)),]
706   or <- order(y)
707   ymin <- min(y, out.yf[,c(1,3,7)])
708   ymax <- max(y, out.yf[,c(1,3,7)])
709 }
710
711 {
712   png(paste0("../results/", model_id, "/predictions.png"))
713   par(mfrow=c(1,1))
714   plot(y[or],ylim=c(ymin,ymax), xlab="", xaxt='n')
715   axis(1, at=1:n, labels=covid$Country[or], las=2)
716   lines(out.yf[or,1],lwd=2,col=2)
717   lines(out.yf[or,3],lty=2,col=2)
718   lines(out.yf[or,7],lty=2,col=2)
719   dev.off()
720 }
721
722 #R^2
723 {
724   png(paste0("../results/", model_id, "/r2.png"))
725   par(mfrow=c(1,1))
726   plot(y,out.yf[,1], ylab="Predicted y")
727   text(y,out.yf[,1], covid$Country)
728   abline(a=0, b=1)
729   dev.off()
730   R2<-(cor(y,out.yf[,1]))^2
731   print(R2)
732   sink(paste0("../results/", model_id, "/r2.txt"))
733   print(R2)
734   sink()
735 }
736
737 #phi
738 {
739   out.phi<-out.sum[grep("phi",rownames(out.sum)),]
740   out.est<-out.phi
741   k<-n
742   ymin<-min(out.est[,c(1,3,7)])

```

```

743   ymax<-max(out.est[,c(1,3,7)])
744 }
745
746 {
747   png(paste0("../results/", model_id, "/phi.png"))
748   par(mfrow=c(1,1))
749   plot(1:k,out.est[,1][or], ylab="",ylim=c(ymin,ymax),
750        xlab="", xaxt='n')
751   axis(1, at=1:n, labels=covid$Country[or], las=2)
752   segments(1:k,out.est[,3][or],1:k,out.est[,7][or])
753   abline(h=0,col="grey70")
754   title("Spatial Effect")
755   dev.off()
756 }
757
758 #theta
759 {
760   out.the<-out.sum[grepl("the",rownames(out.sum)),]
761   out.est<-out.the
762   k<-n
763   ymin<-min(out.est[,c(1,3,7)])
764   ymax<-max(out.est[,c(1,3,7)])
765 }
766
767 {
768   png(paste0("../results/", model_id, "/theta.png"))
769   par(mfrow=c(1,1))
770   plot(1:k,out.est[,1][or],ylab="",ylim=c(ymin,ymax),
771        xlab="", xaxt='n')
772   axis(1, at=1:n, labels=covid$Country[or], las=2)
773   segments(1:k,out.est[,3][or],1:k,out.est[,7][or])
774   abline(h=0,col="grey70")
775   title("Individual Effect")
776   dev.off()
777 }
778
779
780 #Map of lambda
781 {
782   out.lam<-out.sum[grepl("lam",rownames(out.sum)),]
783   plotvar<-out.lam[,1]
784   plotclr <- brewer.pal(nclr,"Yl0rBr")
785   class <- classIntervals(plotvar, nclr, dataPrecision=2,style="
       quantile")
786   colcode <- findColours(class,plotclr)
787
788   covid.map <- sps
789
790   png(paste0("../results/", model_id, "/predicted_smr.png"), res =
       70)
791   plot(covid.map, col = colcode)
792   legend(x_etiquetas, y_etiquetas, legend = names(attr(colcode, "
       table")),
793         fill = attr(colcode, "palette"), cex=1, bty="n")
794   text(mean_coords$x, mean_coords$y, mean_coords$Country)
795   text(mean_coords$x, mean_coords$y-3, plotvar |> round(2))
796   plot_arrows(region)

```

```

797 title(main="Smoothed SMR")
798 dev.off()
799 }
800
801 ##### SMR (standarized morbidity rate)
802 #
803 # Esta solo es para comparar rapidamente con la que se genera en el
804 # chunk anterior
805
806 {
807   plotvar <- y / ee
808   plotvar <- ifelse(plotvar == Inf, 0, plotvar)
809   plotclr <- brewer.pal(nclr,"YlOrBr")
810   class <- classIntervals(plotvar, nclr, dataPrecision=2,style="
      quantile")
811   colcode <- findColours(class,plotclr)
812
813   covid.map <- sps
814
815   plot(covid.map, col = colcode)
816
817   legend(x_etiquetas, y_etiquetas, legend = names(attr(colcode, "
      table")),
818         fill = attr(colcode, "palette"), cex=1, bty="n")
819   text(mean_coords$x, mean_coords$y, mean_coords$Country)
820   text(mean_coords$x, mean_coords$y-3, plotvar |> round(2))
821   plot_arrows(region)
822   title(main="SMR")
823 }
824
825 ##### Resumen de resultados #####
826
827 ##### log(y+1), log(e+1) #####
828
829 log_log_folders <- list.files("../results/", "log_log",
830                               recursive=TRUE, include.dirs=TRUE)
831
832 log_log_folders <- log_log_folders[!(log_log_folders |> grepl(
833   pattern = ".png"))]
834
835 log_log_dics <- list()
836
837 for (folder in log_log_folders){
838   dic <- read.table(paste0("../results/",folder,"/dic.txt"))
839   log_log_dics[folder] <- dic[2][[1]]
840 }
841
842
843 log_log_dics_df <- log_log_dics |>
844   as_tibble() |>
845   t() |>
846   as_tibble() |>
847   rename(dic = V1) |>
848   mutate(model = log_log_folders |>
849     gsub(pattern = "_log_log", replacement = "")) |>
850   select(c(model, dic)) |>

```

```
851   arrange(dic)
852
853 log_log_dics_df
854
855 ##### y, e #####
856
857 folders <- list.files("../results/", "", include.dirs=TRUE)
858
859 folders <- folders[!(folders |> grepl(pattern = ".png"))]
860 folders <- folders[!(folders |> grepl(pattern = "_log_log"))]
861 folders <- folders[!(folders |> grepl(pattern = "_div"))]
862
863 dics <- list()
864
865 for (folder in folders){
866
867   dic <- read.table(paste0("../results/",folder,"/dic.txt"))
868   dics[folder] <- dic[2][[1]]
869
870 }
871
872 dics_df <-dics |>
873   as_tibble() |>
874   t() |>
875   as_tibble() |>
876   rename(dic = V1) |>
877   mutate(model = folders)|>
878   select(c(model, dic)) |>
879   arrange(dic)
880
881 dics_df
```

Código 1: Código R Base

7.4 Códigos Bugs

```

1 model
2 {
3   #Likelihood
4   for (i in 1:n) {
5     y[i] ~ dnorm(mu[i], tau.y)
6     mu[i] <- log(lambda[i]) + ee[i]
7     log(lambda[i]) <- beta[1] + beta[2] * x[i] + theta[i] + phi[i]
8   }
9
10  #Priors
11  tau.y ~ dgamma(0.1, 0.1)
12  for (i in 1:2) { beta[i] ~ dnorm(0, 0.001) }
13  for (i in 1:n) { theta[i] ~ dnorm(0, tau.t) }
14  tau.t ~ dgamma(0.001, 0.001)
15  phi[1:n] ~ car.normal(adj[], weights[], num[], tau.c)
16  tau.c ~ dgamma(0.1, 0.1)
17
18  #Prediction
19  for (i in 1:n) { yf[i] ~ dpois(mu[i]) }
20
21 }

```

Código 2: Modelo CAR Normal Normal

```

1 model
2 {
3   #Likelihood
4   for (i in 1:n) {
5     y[i] ~ dpois(mu[i])
6     mu[i] <- lambda[i] * ee[i]
7     log(lambda[i]) <- beta[1] + beta[2] * x[i] + theta[i] + phi[i]
8   }
9
10  #Priors
11  for (i in 1:2) { beta[i] ~ dnorm(0, 0.001) }
12  for (i in 1:n) { theta[i] ~ dnorm(0, tau.t) }
13  tau.t ~ dgamma(0.001, 0.001)
14  phi[1:n] ~ car.normal(adj[], weights[], num[], tau.c)
15  tau.c ~ dgamma(0.1, 0.1)
16
17  #Prediction
18  for (i in 1:n) { yf[i] ~ dpois(mu[i]) }
19
20 }

```

Código 3: Modelo CAR Normal

```

1 model
2 {
3   #Likelihood
4   for (i in 1:n) {
5     y[i] ~ dnorm(mu[i], tau.y)
6     mu[i] <- log(lambda[i]) + ee[i]
7     log(lambda[i]) <- beta[1] + beta[2] * x[i] + theta[i] + phi[i]
8   }
9
10  #Priors

```

```

11 tau.y ~ dgamma(0.1,0.1)
12 for (i in 1:2) { beta[i] ~ dnorm(0,0.001) }
13 for (i in 1:n) { theta[i] ~ dnorm(0,tau.t) }
14 tau.t ~ dgamma(0.001,0.001)
15 phi[1:n] ~ car.proper(mu_areas[],C[],adj[],num[],M[],tau.c, gamma)
16 tau.c ~ dgamma(0.1,0.1)
17
18 #Prediction
19 for (i in 1:n) { yf[i] ~ dpois(mu[i]) }
20
21 }

```

Código 4: Modelo CAR Propio Normal

```

1 model
2 {
3   #Likelihood
4   for (i in 1:n) {
5     y[i] ~ dpois(mu[i])
6     mu[i] <- lambda[i]*ee[i]
7     log(lambda[i]) <- beta+theta[i]+phi[i]
8   }
9
10  #Priors
11  beta ~ dnorm(0,0.001)
12  for (i in 1:n) { theta[i] ~ dnorm(0,tau.t) }
13  tau.t ~ dgamma(0.001,0.001)
14  phi[1:n] ~ car.proper(mu_areas[],C[],adj[],num[],M[],tau.c, gamma)
15  tau.c ~ dgamma(0.1,0.1)
16
17  #Prediction
18  for (i in 1:n) { yf[i] ~ dpois(mu[i]) }
19
20 }

```

Código 5: Modelo CAR Propio NOX

```

1 model
2 {
3   #Likelihood
4   for (i in 1:n) {
5     y[i] ~ dpois(mu[i])
6     mu[i] <- lambda[i]*ee[i]
7     log(lambda[i]) <- beta[1]+beta[2]*x[i]+theta[i]+phi[i]
8   }
9
10  #Priors
11  for (i in 1:2) { beta[i] ~ dnorm(0,0.001) }
12  for (i in 1:n) { theta[i] ~ dnorm(0,tau.t) }
13  tau.t ~ dgamma(0.001,0.001)
14  phi[1:n] ~ car.proper(mu_areas[],C[],adj[],num[],M[],tau.c, gamma.c)
15  tau.c ~ dgamma(0.1,0.1)
16  gamma.c ~ dunif(0.8, 0.9)
17  gamma.c ~ dunif(min_ie, max_ie)
18
19  #Prediction

```



```

20 for (i in 1:n) { yf[i] ~ dpois(mu[i]) }
21
22 }

```

Código 6: Modelo CAR Propio Rho

```

1  model
2  {
3    #Likelihood
4    for (i in 1:n) {
5      y[i] ~ dpois(mu[i])
6      mu[i] <- lambda[i]*ee[i]
7      log(lambda[i]) <- beta[1] + beta[2]*x[i] + theta[i] + phi[i]
8    }
9
10   #Priors
11   for (i in 1:2) { beta[i] ~ dnorm(0,0.001) }
12   for (i in 1:n) { theta[i] ~ dnorm(0,tau.t) }
13   tau.t ~ dgamma(0.001,0.001)
14   phi[1:n] ~ car.proper(mu_areas[],C[],adj[],num[],M[],tau.c, gamma)
15   tau.c ~ dgamma(0.1,0.1)
16
17   #Prediction
18   for (i in 1:n) { yf[i] ~ dpois(mu[i]) }
19
20 }

```

Código 7: Modelo CAR Propio

```

1  model
2  {
3    #Likelihood
4    for (i in 1:n) {
5      y[i] ~ dpois(mu[i])
6      mu[i] <- lambda[i]*ee[i]
7      log(lambda[i]) <- beta[1] + beta[2]*x[i] + theta[i]
8    }
9
10   #Priors
11   for (i in 1:2) { beta[i] ~ dnorm(0,0.001) }
12   for (i in 1:n) { theta[i] ~ dnorm(mu.t,tau.t) }
13   mu.t ~ dnorm(0, 0.1)
14   tau.t ~ dgamma(0.001,0.001)
15   #Prediction
16   for (i in 1:n) { yf[i] ~ dpois(mu[i]) }
17 }

```

Código 8: Modelo Jerárquico