

MODELAMIENTO BASADO EN AGENTES DE LAS MEDIDAS DE PROTECCIÓN PERSONAL ANTE EL ESPARCIMIENTO DEL VIRUS SARS-COV 2

Por: Jhon Vargas Reyes

Maestría de Ciencia de la Computación - Curso: MCC 639D Modelos Basadas en Agentes y Aplicaciones

Facultad de Ciencias, Universidad Nacional de Ingeniería, Lima, Perú

Lima, 31 de Agosto de 2021

I. INTRODUCCIÓN

Luego de haber presentado varias olas de propagación de la pandemia de COVID-19(SARS-COV 2), todos los países del mundo están luchando por recuperar su economía al levantar lentamente las restricciones de movilidad y las medidas de distanciamiento social aplicadas durante la crisis. Numerosos estudios nos proporcionan evidencia convincente sobre cómo el distanciamiento de contacto, el uso de máscaras faciales y los hábitos de lavado de manos pueden reducir el riesgo de transmisión del SARS-CoV-2. El presente trabajo muestra la investigación realizada a modelos basados en agentes para simular la propagación del COVID-19 incorporando medidas de protección personal.

Palabras clave: COVID-19, SARS-COV 2, SEIR, .

II. MODELOS CLÁSICOS

Se realizará un estudio básico a los modelos SIR y SEIR cómo antesala al modelo propuesto.

Modelo SIR

Este modelo utiliza agentes caracterizados por una variable de 3 estados que representa su estado de salud, que puede tomar los valores SUSCEPTIBLE, INFECTADO o RECUPERADO (e inmune a nuevos contagios).

Llamemos β a la tasa a la que van cada uno de los N individuos en total de susceptible a infectado (dependiendo del contacto con otros individuos infectados dentro de algunos radio de infección, dando lugar a un término de interacción por pares), y γ la tasa a la que los individuos infectados recuperarse y volverse inmune a la enfermedad.

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{S \cdot I}{N} \\ \frac{dI}{dt} &= \beta \frac{S \cdot I}{N} - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

Figura 1.

Este modelo se puede representar con el sistema de ecuaciones de la figura 1.

En la figura 2, podemos ver cómo es el comportamiento con el paso del tiempo y número de casos:

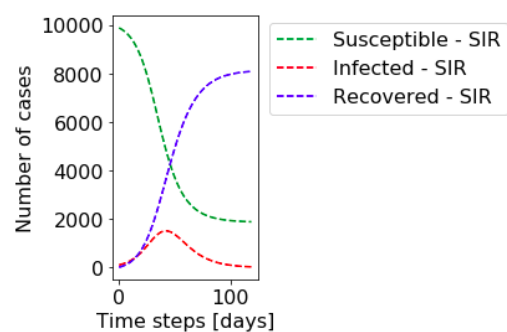


Figura 2.

Modelo SEIR

Un modelo mejorado al SEIR es en donde se consideran a los EXPUESTO(E) a ser contagiados donde se toma en cuenta el periodo de incubación.

Este modelo está basado en la Dinámica de Sistemas y como tal está compuesto por una serie de stocks que

representan diferentes poblaciones y flujos que generan las variaciones entre ellos.

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{S}{N} I \\ \frac{dE}{dt} &= \beta \frac{S}{N} I - \omega E \\ \frac{dI}{dt} &= \omega E - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

Figura 3.

En la figura 3, vemos que los parámetros son los mismos, pero en este caso β es la tasa a la que van cada uno de los N individuos susceptibles pasan a expuestos en donde se toma en cuenta el periodo de incubación.

En la figura 4, podemos ver cómo es el comportamiento con el paso del tiempo y número de casos:

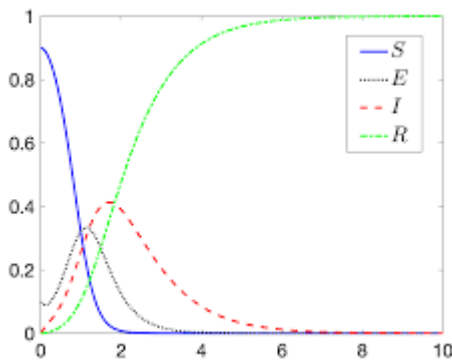
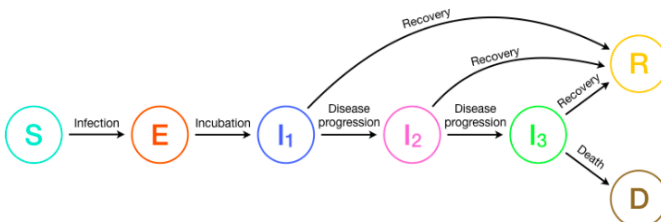


Figura 4.

III. MODELOS PROPUESTO

El modelo propuesto está basado en SEIR, pero incluye estados intermedios de INFECTADOS en los que se considera niveles MODERADO, GRAVE, CRÍTICO. También considera un nuevo estado MUERTO(D)



Este modelo tiene las siguientes características:

Susceptible (S): Las personas susceptibles son aquellas que nunca se han infectado y, por lo tanto, no tienen inmunidad contra COVID-19. Las personas susceptibles quedan expuestas una vez que se infectan con la enfermedad.

Expuesto (E): Los individuos expuestos son aquellos que han sido infectados con COVID-19 pero que aún no son contagiosos para otros. Un individuo permanece expuesto durante el periodo de incubación, después del cual se vuelve infeccioso y experimenta síntomas clínicos leves (I1).

Infectado - leve (I1): las personas con una infección leve experimentan síntomas como fiebre y tos e incluso pueden tener una neumonía leve, pero no requieren hospitalización. Estos individuos pueden recuperarse o progresar a la etapa grave de la enfermedad.

Infectado - grave (I2): las personas con una infección grave tienen neumonía grave y requieren hospitalización. Estos individuos pueden recuperarse o progresar a la etapa crítica de la enfermedad.

Infectado - crítico (I3): los individuos con una infección crítica experimentan insuficiencia respiratoria, choque séptico y / o disfunción o falla de múltiples órganos y requieren tratamiento en una UCI. Estos individuos pueden recuperarse o morir a causa de la enfermedad.

Recuperado (R): los individuos recuperados son aquellos que se han recuperado y se supone que son inmunes a futuras infecciones con COVID-19.

Muertos (D): Los individuos muertos son aquellos que han muerto por COVID-19.

Su formulación matemática se rige con el siguiente sistema de ecuaciones diferenciales que se presenta en la figura 5.

$$\begin{aligned}\dot{S} &= -(\beta_1 I_1 + \beta_2 I_2 + \beta_3 I_3) S \\ \dot{E} &= (\beta_1 I_1 + \beta_2 I_2 + \beta_3 I_3) S - a E \\ \dot{I}_1 &= a E - (\gamma_1 + p_1) I_1 \\ \dot{I}_2 &= p_1 I_1 - (\gamma_2 + p_2) I_2 \\ \dot{I}_3 &= p_2 I_2 - (\gamma_3 + \mu) I_3 \\ \dot{R} &= \gamma_1 I_1 + \gamma_2 I_2 + \gamma_3 I_3 \\ \dot{D} &= \mu I_3\end{aligned}$$

Figura 5.

Descripción del Modelo

El modelo tiene los siguientes componentes:

Agentes, son las personas de una población que pueden pasar a los siguientes estados:

1. S: individuos susceptibles

2. E: Personas expuestas: infectadas pero aún no infecciosas o sintomáticas
3. I_i: individuos infectados en la clase de gravedad i. La gravedad aumenta con i y asumimos que los individuos deben pasar por todas las clases anteriores.
4. I₁: infección leve
5. I₂: infección grave
6. I₃: Infección crítica
7. R: individuos que se han recuperado de la enfermedad y ahora son inmunes
8. D: Individuos muertos
9. N = S + E + I₁ + I₂ + I₃ + I + D Tamaño total de la población (constante)

En la figura 6, podemos ver el diagrama de estados del modelo.

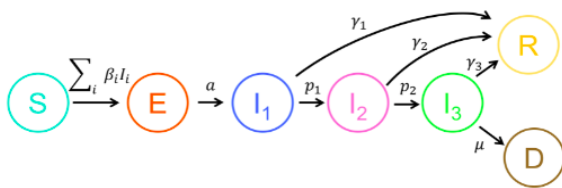


Figura 6.

Entorno, es el ambiente en donde van a interactuar los agentes que se encuentran en diferentes estados. Para nuestro modelo, hemos considerado en ubicar en puntos discretos de un plano cartesiano X, Y.

En la figura 7, podemos ver un snapshot de una ambiente de 100 x 100.

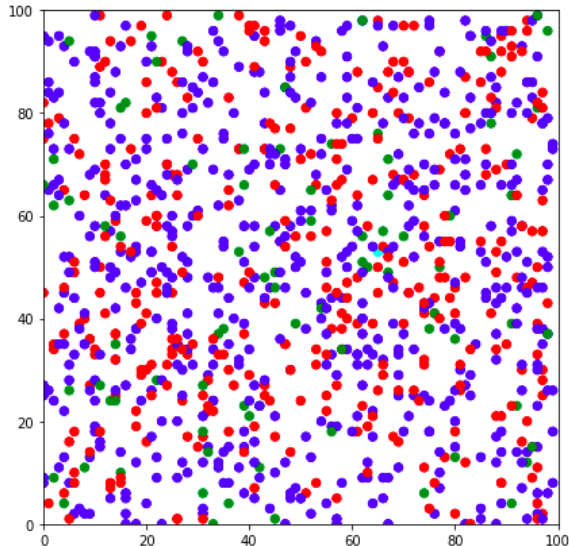


Figura 7.

Reglas, y funciones de transición son las siguientes:

1. Las personas susceptibles (S) que se infectan comienzan en una clase E expuesta, donde son asintomáticas y no transmiten la infección.
2. La tasa de progresión de la etapa expuesta(E) a la etapa infectada(I), donde el individuo es sintomático e infeccioso, ocurre a una tasa a.
3. Las descripciones clínicas de las diferentes etapas de la infección: Los individuos infectados comienzan con una infección MODERADO (I₁), de la cual se recuperan, a una tasa γ_1 , o progresan a una infección GRAVE (I₂), a una tasa p_1 .
4. La infección GRAVE se recupera a una tasa γ_2 o progresa a una etapa CRÍTICA (I₃) a una tasa p_2 .
5. Los individuos con infección CRÍTICA se recuperan a una tasa γ_3 y mueren a una tasa μ .
6. Los individuos recuperados son rastreados por clase R y se supone que están protegidos de la reinfección de por vida.
7. Los individuos pueden transmitir la infección en cualquier etapa, aunque con diferentes tasas.

Time, se usó un tiempo discreto en la que en cada instante se evalúa a todos los individuos según la distancia social.

Parameters, que fueron usados fueron basados en la siguiente tabla 1:

Parámetro	Descripción
$\beta_1, \beta_2, \beta_3$	tasa a la que los individuos infectados de la clase I _i contactan a los susceptibles y los infectan
a	tasa de progresión de la clase expuesta a la infectada
$\gamma_1, \gamma_2, \gamma_3$	tasa a la que los individuos infectados en la clase I _i se recuperan de la enfermedad y se vuelven inmunes
p_1, p_2	tasa a la que los individuos infectados en la clase I _i progresan a la clase I _{i+1}
μ	Tasa de mortalidad de para personas en la etapa más grave de la enfermedad

Tabla 1.

IV. EXPERIMENTACIÓN Y RESULTADOS

Modelos, en la parte del agente, entorno e interacciones se creacion las siguientes clases que describen el modelo:

```
class Patient:
    def __init__(self, x, y, input_status):
        self.status = input_status # S=0, E=1, I1=2, I2=3, I3=4, R=5, D=6
        self.x = x
        self.y = y
        self.next_status = input_status
        self.next_x = x
        self.next_y = y
```

```
def recover(self):
    self.next_status = 5

def death(self):
    self.next_status = 6

def infect(self):
    for target in AGENTS:
        if target.status == 0 and \
            fabs(target.x - self.x) <= I_RADIUS and \
            fabs(target.y - self.y) <= I_RADIUS:
            r = rand()

            I1 = len([a for a in AGENTS if a.status == 2])
            I2 = len([a for a in AGENTS if a.status == 3])
            I3 = len([a for a in AGENTS if a.status == 4])

            if r < ((b[1]*I1+b[2]*I2+b[3]*I3):
                target.next_status = 1 # Estado inicial cuando el S pasa a E

    if target.status == 1:
        r = rand()
        if r < a:
            target.next_status = 2 # Estado inicial cuando el E pasa a I
```

```
def showPlot(end):

    view.clear()

    view.scatter([a.x for a in AGENTS], [a.y for a in AGENTS],
                 s=40, cmap = matplotlib.colors.ListedColormap(["blue", "yellow", "black", "cyan", "magenta"]
                 vmin=0, vmax=6))

    view.set aspect(1)
    view.axis([0, SIZE, 0, SIZE])
```

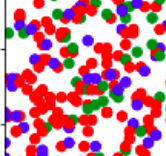
```
def move(self):
    r = rand()
    if r < MOVE:
        r = rand()
        if r < 0.25:
            self.next_x = (self.x - 1) % SIZE # Movimiento Horizontal Izquierda
        elif r < 0.5:
            self.next_x = (self.x + 1) % SIZE # Movimiento Horizontal Derecha
        elif r < 0.75:
            self.next_y = (self.y - 1) % SIZE # Movimiento Vertical Abajo
        else:
            self.next_y = (self.y + 1) % SIZE # Movimiento Vertical Arriba
```

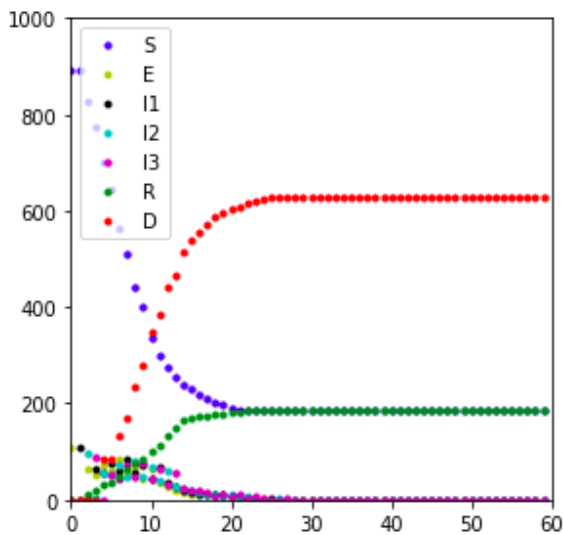
```
def do_actions(self):
    self.move()
    self.infect()

    if self.status == 1:
        r = rand()
        if r < a:
            self.next_status = 2
```

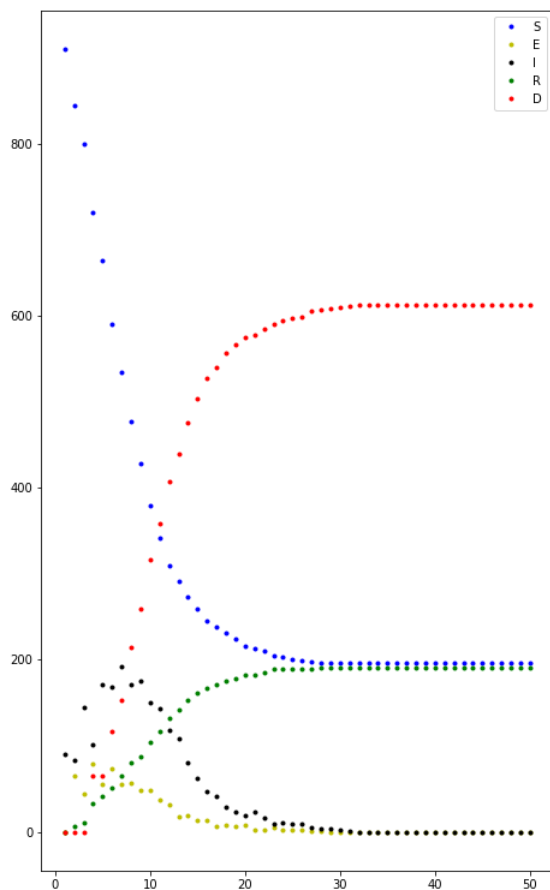
```
# Analisis Estado I1 (2)
if self.status == 2:
    r = rand()
    if r < g[1]:
        self.recover()
    elif r > p[1]:
        self.next status = 3
```

```
# Analisis Estado I3 (4)
if self.status == 4:
    r = rand()
    if r < g[3]:
        self.recover()
    elif r > u:
        self.death()
```





Acumulado de infectados:



Análisis de Incertidumbre y Sensibilidad, para iniciar este análisis tenemos que definir la distribución PDF de los parámetros. En la tabla 2, podemos ver las diferentes distribuciones.

Parameter	Distribution	Arguments
beta1	normal	$\mu = 0.5, \sigma = 0.15$
beta2	normal	$\mu = 0.1, \sigma = 0.25$
beta3	normal	$\mu = 0.1, \sigma = 0.13$
a	normal	$\mu = 0.2, \sigma = 0.27$
gama1	normal	$\mu = 0.133, \sigma = 0.41$
gama2	normal	$\mu = 0.125, \sigma = 0.25$
gama3	normal	$\mu = 0.075, \sigma = 0.39$
p1	normal	$\mu = 0.033, \sigma = 0.33$
p2	normal	$\mu = 0.042, \sigma = 0.27$
u	normal	$\mu = 0.05, \sigma = 0.11$

Tabla 2.

Luego usando R, generamos el LHS de los parámetros(SEIR_LHS_TEST):

```

parameters <- c("beta1", "beta2", "beta3", "a", "gama1", "gama2", "gama3", "p1", "p2", "u")
parameters_dist <- c("qnorm", "qnorm", "qnorm", "qnorm", "qnorm", "qnorm", "qnorm", "qnorm", "qnorm", "qnorm", "qnorm")
parameters_dist_arg <- list(list(mean=0.5, sd=0.15),
                             list(mean=0.1, sd=0.25),
                             list(mean=0.1, sd=0.13),
                             list(mean=0.2, sd=0.27),
                             list(mean=0.133, sd=0.41),
                             list(mean=0.125, sd=0.25),
                             list(mean=0.075, sd=0.39),
                             list(mean=0.033, sd=0.33),
                             list(mean=0.042, sd=0.27),
                             list(mean=0.05, sd=0.11))

SEIR_LHS_TEST <- LHS(model=NULL, parameters, 50, parameters_dist, parameters_dist_arg)

# Guardando los Datos:
write.csv(get.data(SEIR_LHS_TEST), file="SEIR_LHS_TEST.csv")

```

En la figura 8, podemos ver el LHS generado para los parámetros.

	beta1	beta2	beta3	a	gama1
1	0.643124787971929	0.511213406737868	0.171940013542237	-0.131162592409885	0.3133645
2	0.518849202028261	-0.053203247754157	0.313830971503691	-0.307814274200838	0.3842533
3	0.601173462529412	-0.009978291418308	0.164460545155169	0.26143714369311	0.0184791
4	0.316020781994508	0.68158696851021	0.143140935036786	0.179677137233046	-0.0249813
5	0.848952181106126	0.223962586836863	0.122928641421512	-0.428113925991027	0.3362988
6	0.503760336238807	0.196330116601892	0.234736340634193	-0.017733736694925	0.738017
7	0.253271955957279	0.030170241388137	0.259448655604759	-0.079837015163323	-0.1699272
8	0.416992292066649	0.435188758422554	-0.113830971503691	0.562003859096359	-0.5413895
9	0.278631345773124	0.284711712296303	0.344503169059663	0.318776554731773	0.2475208
10	0.458102144832882	0.182963336609204	0.129580846963349	0.275416139300813	0.0397066
11	0.721368654226875	0.169829758611864	0.402425223625309	0.349413874280032	-0.6381253
12	0.442201930038865	0.30160531175456	0.035539454844831	0.206768605229852	0.081478
13	0.496239663761193	-0.48158696851021	0.150091660632984	0.124583860699187	0.5948203
14	0.379036812947264	0.468947757044792	-0.004834762112371	0.644110479276897	0.2909813
15	0.574377552102118	0.406632030009153	0.070419153036651	0.457624618349472	0.4929374
16	0.151047818893874	-0.235188758422554	-0.01412651835666	0.220322862766954	-0.0030596
17	0.526456124717129	0.068584663286232	0.196050090394078	-0.037031999663832	-0.472074
18	0.425622447897882	0.131415336713769	0.224041482909005	0.23392856365087	-0.1976327

Figura 8.

Luego con esos parámetros, lo aplicamos al modelo y generamos SEIR_MODEL_Result. Esto lo generamos en python, y generamos un archivo csv y lo corremos 50 iteraciones:

```

1 import pandas as pd
2 df = pd.read_csv ('../SEIR_LHS_TEST.csv')
3
4 df.head()

```

	Unnamed: 0	beta1	beta2	beta3	a	gama1	gama2	gama3	p1	p2	u
0	1	0.643125	0.511213	0.171940	-0.131163	0.313364	0.234978	-0.566493	0.074468	0.404004	-0.010872
1	2	0.518849	-0.053203	0.313831	-0.307814	0.384253	0.493948	0.417380	0.024727	-0.140112	0.080725
2	3	0.601173	-0.009978	0.164461	0.261437	0.018479	0.001037	-0.140820	-0.587662	-0.586114	0.001610
3	4	0.316021	0.681587	0.143141	0.179677	-0.024981	0.118733	-0.033934	0.299119	-0.076777	0.173903
4	5	0.848952	0.223963	0.122929	-0.428114	0.336299	-0.456587	0.163743	0.178171	0.075929	-0.004544

Ejecucion de SEIR_LHS_TEST

```

1 # Modelo SIR extendido
2 EstadosGeneral = []
3
4 #Parametros Interaccion
5 MOVE = 0.75
6 SIZE = 100
7 I0 = 0.1
8 I_RADIUS = 2
9
10 # Generacion de Poblacion Inicial
11 NAGENTS = 1000
12
13 Estado=[]
14
15 for i in range(0,50):
16     parametros.append([df.beta1[i], df.beta2[i], df.beta3[i], a, df.gama1[i], df.gama2[i], df.gama3[i], df.
17     b[i] = df.beta1[i]
18     b[2] = df.beta2[i]
19
20 Iteracion 0:
21 0.643124787971929 0.511213406737868 0.171940013542237 -0.131162592409885 0
22 566492914511074 0.0744682444621745 0.404003859096358 -0.010872319151124
23 Iteracion 1:
24 0.518849202028261 -0.0532032477541567 0.313830971503691 -0.307814274200838
25 0.417379555069979 0.0247272602746254 -0.140112232552942 0.08072509378922
26 Iteracion 2:
27 0.601173462529412 -0.0099782914183084 0.164460545155169 0.26143714369311 0
28 0.140820040626712 -0.587661890689913 -0.586113925991027 0.0016095517759442
29 Iteracion 3:
30 0.316020781994508 0.68158696851021 0.143140935036786 0.179677137233046 -0.
31 0339344234345071 0.299119011516019 -0.0767765547317731 0.173903024194268
32 Iteracion 4:
33 0.848952181106126 0.223962586836863 0.122928641421512 -0.428113925991027 0
34 63742540890048 0.178171344672167 0.07592856365087 -0.0045435382082198
35 Iteracion 5:
36 0.503760336238807 0.196330116601892 0.234736340634193 -0.0177337366949249
37 267379555069979 -0.210819460231121 -0.123459507574489 0.0582796848309813
38 Iteracion 6:
39 0.583007707933351 0.0559064588047847 0.274298154379728 0.417733736694925 0
40 31399706496485 0.215616957453372 0.486110479276897 0.138706337172006
41 Iteracion 46:
42 0.7011325053533 0.238346179888918 0.0836640249088404 0.437031999663832 0
43 0.0137425408900483 0.375023018532951 -0.465814274200838 0.197483053705924
44 Iteracion 47:
45 0.66895866935582 0.26862243754902 -0.144503169059663 -0.198463577608376 0.
46 13150271182233 0.475449161117771 -0.157488649280008 0.0076147486951675
47 Iteracion 48:
48 0.21788958777312 0.106267227064678 0.109785082072978 0.365459507574489 0.
49 566134612561 0.235228287035487 0.191413874280032 0.0134961318919502
50 Iteracion 49:
51 0.534131746496172 -0.370198402037813 0.291852833663292 0.598463577608376 0
52 0.0544228051103584 0.520011039299126 0.035231394770148 -0.0312731534103735

```

Lego realizamos el análisis de correlación PRCC:

```

# Mis Resultado
#SEIR_MODEL_Result <- read.table("SEIR_MODEL_Result.csv",sep = ",",header = TRUE)
SEIR_MODEL_Result <- read.table("SEIR_MODEL_Result.csv")

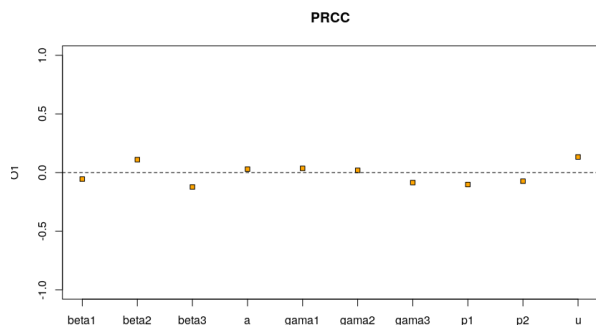
```

```

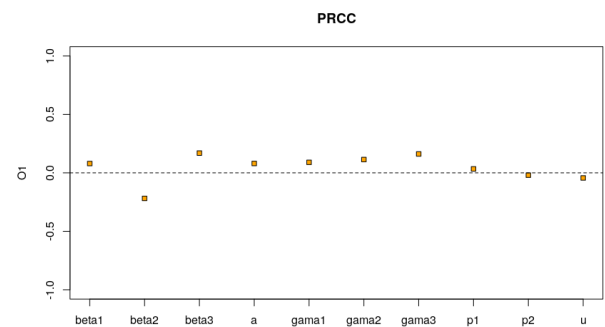
acopladaLHS_1 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,2])
acopladaLHS_1 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,1])
acopladaLHS_2 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,2])
acopladaLHS_3 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,3])
acopladaLHS_4 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,4])
acopladaLHS_5 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,5])
acopladaLHS_6 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,6])
acopladaLHS_7 <- tell(SEIR_LHS_TEST, SEIR_MODEL_Result[,7])

```

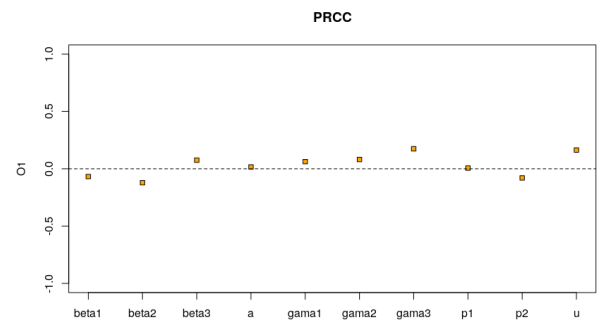
SUSCEPTIBLE - PRCC:



RECOVERED - PRCC:



DEATH - PRCC:



V. CONCLUSIONES

1. Los modelos SEIR, SIR tienen mucha similitud con la propagación de virus, pero tienen algunas limitaciones al momento de asumir la variabilidad de otros factores haciendo más complejo el modelo.
2. Analizando el PRCC podemos ver que casi todas los parámetros tienen influencia en los parámetros de estudio S, R y D.

VI. REFERENCIAS

[1] Md. Salman Shamil , Farhanaz Farheen , Nabil Ibtehaz, Irtesam Mahmud Khan , M. Sohel Rahman. An Agent-Based Modeling of COVID-19: Validation, Analysis, and Recommendations

[2] Yaqing Fang , Yiting Nie , Marshare Penny. Transmission dynamics of the COVID-19 outbreak and effectiveness of government interventions: A data-driven analysis

[3] Y. Fang, Y. Nie, and M. Penny. Transmission dynamics of the covid-19 outbreak and effectiveness of

government interventions: A data-driven analysis.
Journal of Medical Virology, n/a(n/a).

[4] Abbott, Sam. 2020. "Temporal Variation in Transmission During the COVID-19 Outbreak." CMMID Repository.
<https://cmmid.github.io/topics/covid19/current-patterns-transmission/global-time-varying-transmission.html>.

[5] Wu, Joseph T., Kathy Leung, Mary Bushman, Nishant Kishore, Rene Niehus, Pablo M. de Salazar, Benjamin J. Cowling, Marc Lipsitch, and Gabriel M. Leung. 2020. "Estimating Clinical Severity of COVID-19 from the Transmission Dynamics in Wuhan, China." Nature Medicine, March, 1–5.
<https://doi.org/10.1038/s41591-020-0822-7>.