

SIMULACIÓN DE EPIDEMIAS SIR

UNIVERSIDAD AUTÓNOMA DE MADRID



GUILLERMO RODRÍGUEZ SENÍS
ADRIÁN RUBIO PINTADO

INDICE

1.Introducción.....	3
2.Modelo.....	4
3.Librerías.....	5
4.Topologías.....	7
4.1 Erdős-Rényi.....	7
4.2 Barabasi-Albert.....	8
4.3 Newman-Watts-Strogatz.....	9
5.Simulaciones.....	10
5.1 Simulaciones con diferentes topologías.....	11
5.2 Simulaciones variando la topología del grafo.....	18
5.2.1 Introducción.....	18
5.2.2 Parámetros.....	20
5.2.3 Simulación fijando el tiempo t a partir de medidas	20
5.2.4 Simulación fijando el porcentaje de eliminación.....	22
5.2.4.1 Medidas sociales restrictivas moderadas.....	22
5.2.4.2 Medidas sociales restrictivas ligeras.....	26

6. Validación.....	28
6.1 Validación de la implementación del modelo.....	28
6.2 Validación del modelo.....	29
6.2.1 Sin modificaciones de las relaciones individuales.....	30
6.2.1 Test de Kolmogorov-Smirnov.....	30
6.2.3 Valor esperado e intervalo de confianza.....	31
6.2.4 Con modificaciones de las relaciones individuales.....	31
6.2.4.1 Validación parte 1.....	33
6.2.4.2 Validación parte 2.....	33
6.2.4.3 Validación sobre la simulación total.....	35
7. Conclusiones.....	38
8. Bibliografía.....	39
9. Anexos.....	40

1.Introducción

Queremos analizar las dinámicas de un SIR estándar con dinámicas Markovianas. Dado que con dichas dinámicas es posible para las tasas de transmisión depender intrínsecamente de propiedades individuales y de relaciones entre personas, se pretende utilizar un grafo para modelar esto.

EL uso de un grafo para modelar contactos entre personas, abre la posibilidad para observar cómo comportamientos individuales afectan a la transmisión.

El objetivo del proyecto es doble:

1. Analizar el efecto de diferentes topologías de red
2. Analizar el comportamiento de la epidemia cuando se incluye *la reacción de los individuos ante una epidemia*.

2.Modelo

Se modelará la red de individuos como un grafo para hacer una simulación markoviana en tiempo discreto: los nodos del grafo representan a los individuos y las aristas las relaciones sociales entre ellos. Cada nodo puede estar en 3 posibles estados: Susceptible(S), Infectado(I) o Recuperado(R).

En cada instante de tiempo el estado de la red estará caracterizado por el estado de sus N nodos (n -tupla o vector de tamaño n). Por tanto la red podrá estar en 3^N estados diferentes.

El diagrama de flujo de estados de un nodo es el siguiente:

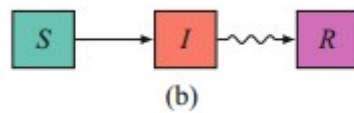


Diagrama de flujo de un nodo

En el modelo tenemos 2 tipos de eventos discretos: infección y recuperación:

La **infección** es transmitida a través de un enlace entre un nodo infectado y uno susceptible. Cuando un nodo en estado susceptible (S) pasa a estado (I), este podrá infectar a sus vecinos, de acuerdo a los enlaces entre ellos. El tiempo que transcurra desde que un nodo pasa a estado I hasta que infecta a uno de sus vecinos es una variable aleatoria que se distribuye según una exponencial de **ratio tau: τ (o beta)**. (Proceso de Markov)

Un nodo en estado infectado se **recuperará** pasando a estado (R) de manera independiente al resto de los nodos tras un intervalo de tiempo aleatorio dado por un proceso exponencial de **ratio γ** . (Proceso de Markov)

Estos dos parámetros los fijaremos a un valor determinado para todas las simulaciones.

Para procesar los eventos se utilizará una cola de prioridad de eventos futuros. Se procesarán eventos en ella mientras no esté vacía.

Nos centraremos, únicamente en la curva de infectados, dado el alcance del proyecto. Analizaremos la influencia de las topologías del grafo en este modelo, ya que éstas representan comportamientos individuales de relaciones sociales.

3.Librerías

Como librería principal para la simulación en el proyecto **EoN** (Epidemics on Networks). En concreto la función `fast_sir`, que nos permitirá mediante una cola de eventos discretos futuros simular a nivel individual de cada nodo los contagios en la red con una dinámica Markoviana. En ella los tiempos de contagios serán una variable aleatoria continua con distribución exponencial.

Podemos ver el pseudocódigo de dicha función en la siguiente figura:

```
Input: Network  $G$ , per-edge transmission rate  $\tau$ , recovery rate  $\gamma$ , set of index node(s)
initial_infecteds, and maximum time  $t_{\max}$ .
Output: Lists times,  $S$ ,  $I$ , and  $R$  giving number in each state at each time.

function fast_SIR( $G, \tau, \gamma$ , initial_infecteds,  $t_{\max}$ )
    times,  $S$ ,  $I$ ,  $R \leftarrow [0], [|G|], [0], [0]$ 
     $Q \leftarrow$  empty priority queue
    for  $u$  in  $G$ .nodes do
         $u$ .status  $\leftarrow$  susceptible
         $u$ .pred_inf_time  $\leftarrow \infty$ 
    for  $u$  in initial_infecteds do
        Event  $\leftarrow$  {node:  $u$ , time: 0, action: transmit}
         $u$ .pred_inf_time  $\leftarrow$  0
        add Event to  $Q$  ▷ ordered by time
    while  $Q$  is not empty do
        Event  $\leftarrow$  earliest remaining event in  $Q$ 
        if Event.action is transmit then
            if Event.node.status is susceptible then
                process_trans_SIR( $G$ , Event.node, Event.time,  $\tau, \gamma$ , times,  $S$ ,  $I$ ,  $R$ ,  $Q$ ,  $t_{\max}$ )
            else
                process_rec_SIR(Event.node, Event.time, times,  $S$ ,  $I$ ,  $R$ )
    return times,  $S$ ,  $I$ ,  $R$ 
```

Figura A.3 Epidemics on networks[1]

Esta función como su propio nombre indica permite simulaciones rápidas. Por ejemplo, para una grafo Erdős-Rényi de 1 millón de individuos la simulación tarda en torno a 1-2 minutos. El tiempo de simulación es similar al de generar la propia red por la librería de grafos.

Esta función nos permite especificar una fracción inicial de nodos infectado o un set de nodos Python.

También utilizaremos la librería **NetworkX**. Esta librería nos permite crear, manipular y estudiar de la estructura, dinámica y funciones de redes y grafos.

En la vida real, las redes de individuos son diferentes. Existen diferentes tipos de individuos con diferentes estilos de vida dependiendo de donde vivan y lo sociables que sean. Para poder simular esto, con esta librería crearemos diferentes topologías de grafos (equilibrados, con hubs, small worlds, etc). De esta manera el análisis y la simulación de las epidemias serán más precisas.

Además, nos apoyaremos en esta librería para manipular grafos previamente creados con ella. Por ejemplo, al simular unas medidas de estado durante una pandemia donde restringen la movilidad y el contacto, manipularíamos los grafos eliminando conexiones.

También se ha desarrollado código propio, principalmente para la realización y estudio de simulaciones en dos fases, donde se simula una restricción en las relaciones sociales modificando los enlaces del grafo una vez que se ha alcanzado un cierto número de contagios. Para más detalles consultar el anexo: *“Principal listado de funciones y código de implementación propia para el proyecto”*.

4.Topologías

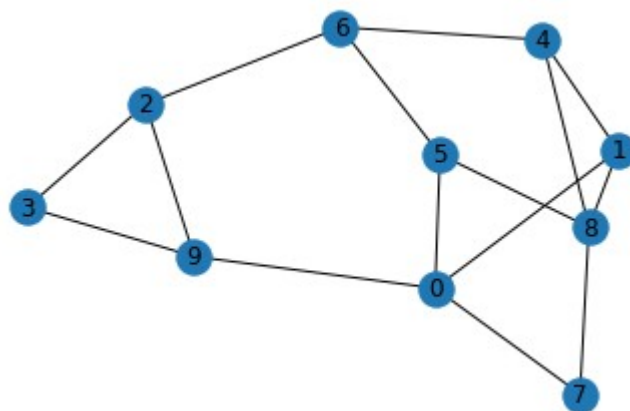
En este apartado nos centraremos en analizar diferentes topologías de grafos y redes que intentan representar las redes de individuos con sus relaciones con otros individuos en el mundo real.

Para que las comparaciones de las simulaciones y resultados tengan más sentido, todos los grafos que simularemos deberán tener el mismo número de nodos (N) y el mismo número de conexiones (o conexiones medias por nodo según la topología). Para ello fijaremos el número de nodos en $N=100.000$, y ajustaremos los parámetros de las funciones creadoras de grafos para que tengan el mismo número de conexiones, fijando el número medio de conexiones por nodo a 5(en las topologías pertinentes). Esto último se comprobará utilizando la función `Graph.number_of_edges()`.

Una vez fijados estos dos valores, describimos las diferentes topologías utilizadas.

4.1 ERDÖS-RÉNYI

Los grafos Erdős-Rényi son redes puramente aleatorias donde todos los nodos tienen aproximadamente el mismo número de conexiones. Además, estas conexiones también son puramente aleatorias.



Ejemplo red Enros-
Renyi con 10 nodos y una media de 3 conexiones por nodo.

Esta topología es la más aleatoria y se intenta asemejar a una sociedad equilibrada donde todos los individuos tienen aproximadamente el mismo número de relaciones y de forma arbitraria. También se puede

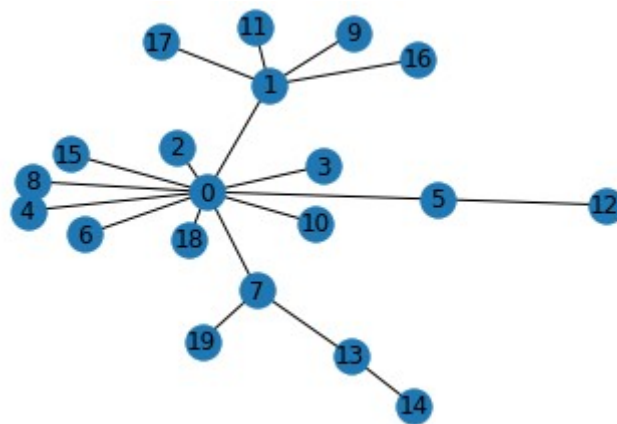
utilizar como una red estándar, ya que todos los modelos de redes reales son diferentes y esta se podría considerar la más genérica.

Por esto último, nos centraremos en este tipo de redes. Además, se pueden simular como un modelo SIR determinista de campo medio basado en ODEs. Posteriormente contrastaremos los dos tipos de simulaciones para validar el modelo de campo medio y poder hacer predicciones de una manera más sencilla.

Aun así, interesante simular otras topologías de redes para ver cómo evolucionan las pandemias dependiendo de las distribuciones de las conexiones entre los nodos.

4.2 BARABASI-ALBERT

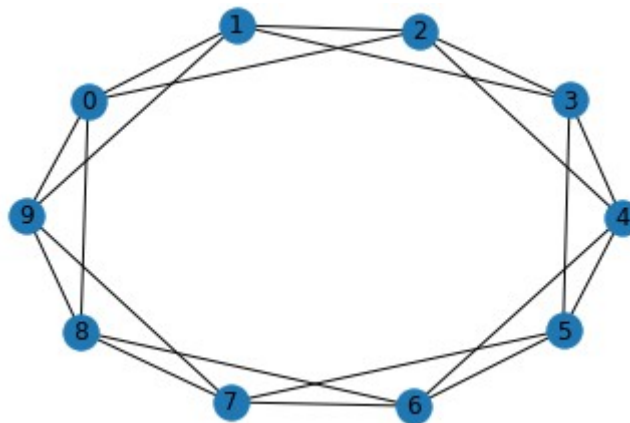
En este caso simularemos una red Barabasi-Albert, donde debido a su algoritmo que sigue el mecanismo de la libre escala, las conexiones se concentran en algunos nodos debido a su crecimiento exponencial, a los que llamaremos *hubs*.



Ejemplo red Barabasi-Albert con 20 nodos y con 1 conexión para cada nodo nuevo.

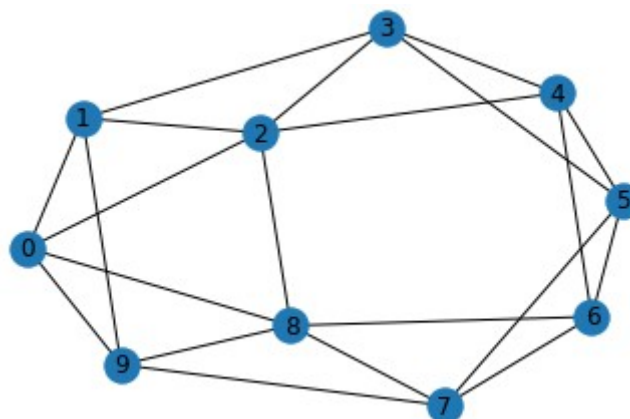
4.3 NEWMAN-WATTS-STROGATZ

También simularemos redes de Newman-Watts-Strogatz. Estas redes se caracterizan por ser redes circulares de mundo pequeño (small-world) de manera que cada nodo (individuo) solo está en contacto con sus k (en nuestro caso $k = 10$) nodos vecinos. De esta manera, cada pareja de vecinos directos está en contacto con casi la misma gente (comparten $k-1$ de sus k vecinos).



Ejemplo red circular con 10 nodos y 4 conexiones con vecinos

No obstante, el algoritmo para generar redes Newman-Watts-Strogatz, permite crear conexiones aleatorias según la probabilidad que decidamos entre dos nodos aleatorios. De esta manera, también hemos generado un grafo Newman-Watts-Strogatz con 9 vecinos por nodo, además de una probabilidad del 25% de crear conexiones aleatorias entre nodos aleatorios. Cumpliendo así que el número medio de conexiones por nodo sea 10.



Ejemplo red Newman-Watts-Strogatz con 10 nodos, 4 conexiones con vecinos y una conexión aleatoria (2 - 8)

5. Simulaciones

Para todas las simulaciones se utilizarán los siguientes parámetros:

Tasa de transmisión:

$$\text{Tau } (\tau) = 0.3$$

De acuerdo a la literatura, en ocasiones nos referimos a el como beta β .

Tasa de recuperación:

$$\text{Gamma } (\gamma) = 1.0$$

Tamaño de la población:

$$N = 100.000 \text{ nodos}$$

Fracción inicial de contagiados

$$\text{Rho} = 0.005$$

5.1 Simulaciones con diferentes topologías

En esta sección estudiaremos cómo se comporta la curva de contagios para las diferentes topologías que hemos simulado.

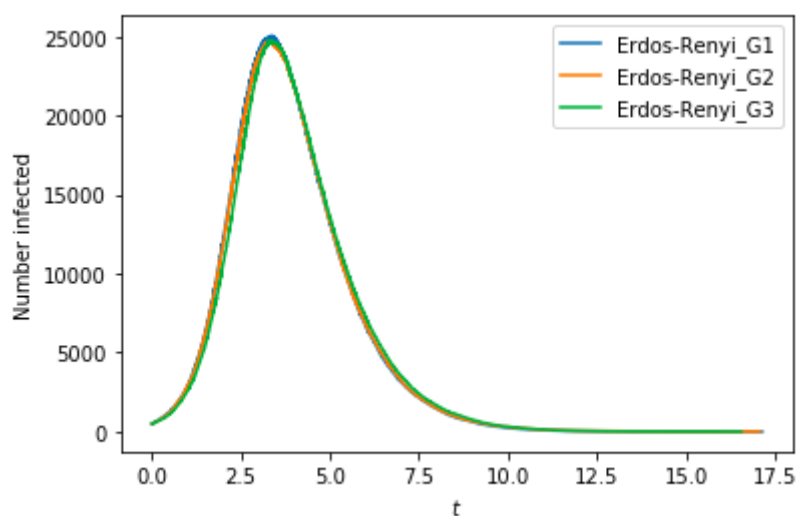
El proceso de simulación para las diferentes topologías será simplemente crear el grafo correspondiente para que tengan el mismo número de nodos y conexiones, aplicar la simulación SIR con la función **Eon.fast_SIR()** y graficar los resultados.

Ya que las simulaciones empiezan infectando a ciertos nodos aleatorios, y las topologías analizadas son de grafos aleatorios (excepto en el caso del grafo circular), realizaremos varias simulaciones sobre el mismo y sobre diferentes grafos.

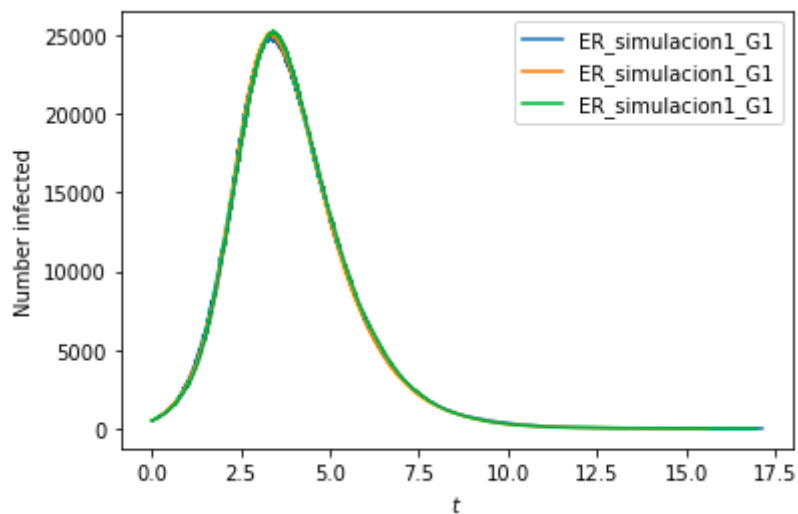
ERDÖS-RÉNYI

Para crear las redes de tipo Erdős-Rényi, utilizaremos la función **fast_gnp_random_graph()**. Esta función nos creará el grafo con un número N de nodos (100 000) y una probabilidad de creación de nueva conexión (0.001) que obtendremos partiendo del número esperado de vecinos (10) entre el número de nodos menos 1.

Al realizar varias simulaciones sobre tres grafos de esta topología con el mismo número promedio de conexiones y de nodos, podemos observar que la evolución de la pandemia es muy similar.



Tres simulaciones para tres grafos Erdos-Renyi diferentes.



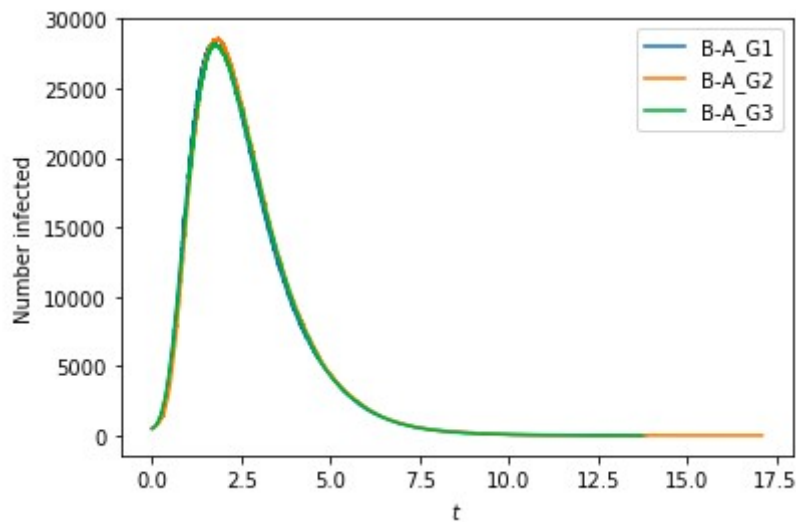
Tres simulaciones para el mismo grafo E-R.

Para esta topología no afectan a nivel significativo la creación aleatoria del grafo ni cuales sean los nodos iniciales infectados. Esto se debe a que, para un gran número de nodos, el grafo se estabiliza debido a su aleatoriedad en las conexiones y a que todos los nodos tienen en promedio el mismo número de conexiones.

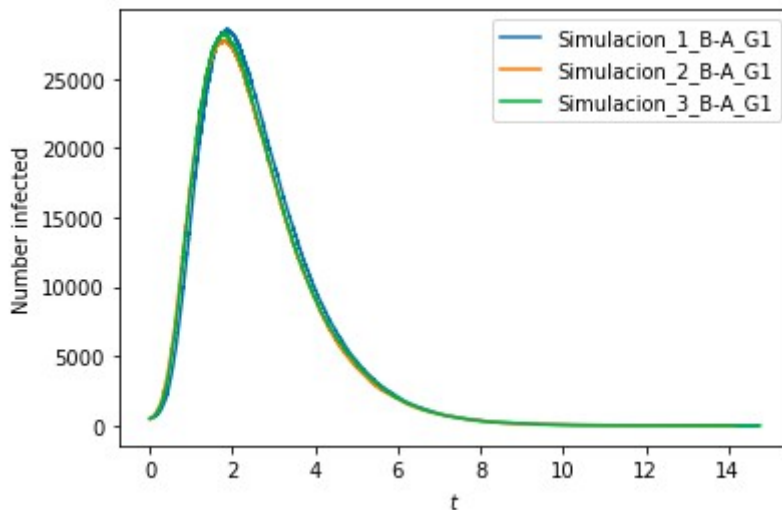
BARABASI-ALBERT

Para las redes Barabasi-Albert, será necesario llamar a la función **barabasi_albert_graph()** de la librería **networkx**. Indicaremos que el tamaño de la red será 100 000 nodos, y le indicaremos que unas 5 conexiones desde el nodo nuevo a los nodos existentes. De esta manera el numero promedio de conexiones por nodo será 5 (comprobado llamando a la función *Graph.number_of_edges() / N*).

Una vez creados tres grafos con los mismos parámetros realizamos las mismas simulaciones.



Tres simulaciones para tres grafos Barabasi-Albert diferentes.



Tres simulaciones para el mismo grafo Barabasi-Albert.

Como podemos observar, para la topología Barabasi-Albert tampoco aparecen grandes cambios ni al simular varias veces sobre una misma red ni en las simulaciones de redes diferentes con mismo tamaño y número de vecinos a añadir. Esto se debe al gran tamaño de la red, que acaba convergiendo a un equilibrio siempre y cuando tengan el mismo tamaño y mismas conexiones.

Como podemos observar en la simulación, la evolución de la pandemia SIR sucede mucho más rápida y más agresiva.

Llegando a máximo número de infectados significativamente más grande al de la simulación en grafos puramente aleatorios como el modelo Erdős-Rényi. Sin embargo, la pandemia dura menos tiempo ya que se llega pronto

al pico de infectados que es muy elevado, por lo que transmitirán la enfermedad al resto de la red y se recuperarán antes.

NEWMAN-WATTS-STROGATZ

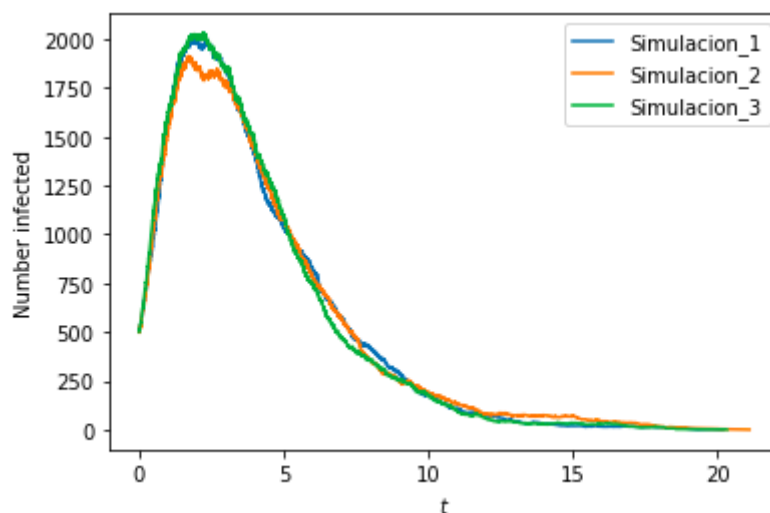
El análisis de simulaciones de esta topología las partiremos en dos: una sobre la red circular d mundo pequeño (small-world) sin conexiones aleatorias; y la otra con conexiones aleatorias.

Para ambos análisis, para la creación de grafos, utilizaremos la misma función. La función es **newman_watts_strogatz_graph()** de la librería **networkx**. Ambas con un tamaño de red de 100 000 nodos.

Sin conexiones aleatorias (Circular)

Para crear este grafo usaremos un $k = 10$ vecinos, para que cada nodo tenga exactamente 5 conexiones.

En cuanto a las simulaciones, en este caso solo crearemos un grafo. Pues los grafos circulares no son grafos aleatorios y, para los mismos parámetros tendríamos los mismos grafos. Sin embargo, sí que simularemos varias veces para un grafo creado. Este es el resultado:



Tres simulaciones para el mismo grafo Newman-Watts-Strogatz.

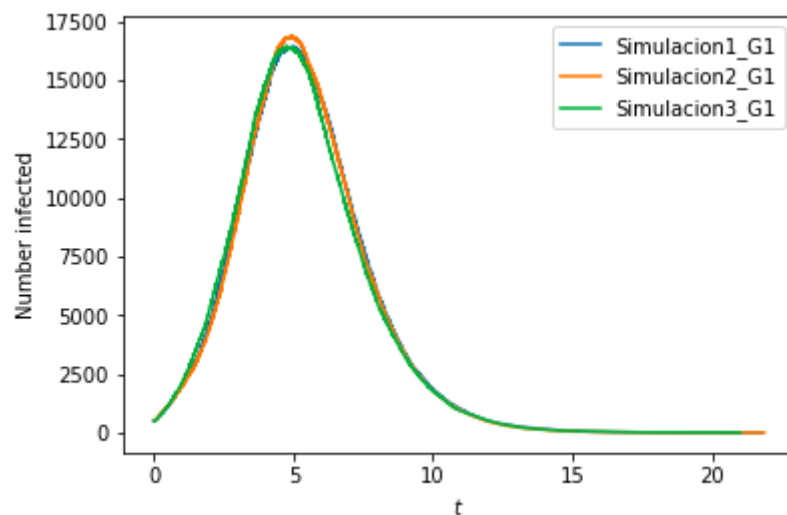
En este caso si se pueden apreciar leves diferencias en las diferentes simulaciones. Esto se debe a que la pandemia evoluciona lentamente. Lo que provoca que el pico con el máximo número de infectados no sea muy

elevado y la escala de la gráfica sea menor, por lo que se aprecian más las diferencias. Además, debido a la forma de este tipo de red, la epidemia avanza desde los nodos infectados hacia sus vecinos, sin saltar a otras zonas de la red. Esto provoca que la aleatoriedad de las variables aleatorias de contagio afecten más al desarrollo de la epidemia.

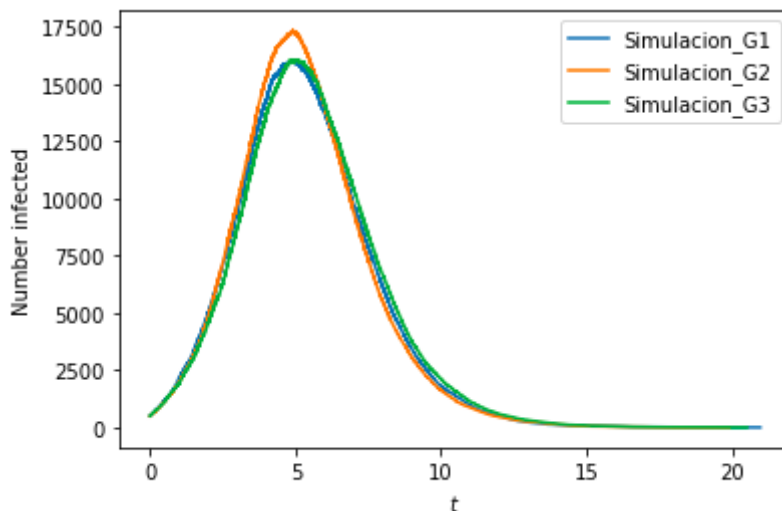
Con conexiones aleatorias (Newman-Watts-Strogatz)

En este caso, al ser grafos aleatorios, sí que crearemos varios grafos con los mismos parámetros llamando a la función indicada anteriormente. El número de vecinos será 9 en vez de 10, para que al incluir conexiones aleatorias (con probabilidad de 0.25) se siga cumpliendo que el número promedio de conexiones por nodo sea 5 (comprobado llamando a la función *Graph.number_of_edges() / N*).

Los resultados de las simulaciones son los siguientes:



Tres simulaciones para el mismo grafo Newman-Watts-Strogatz.



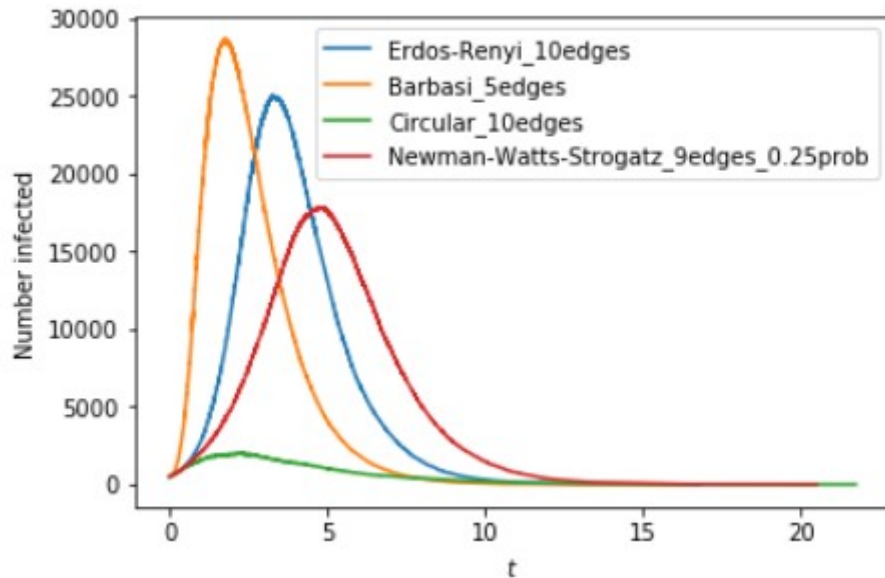
Tres simulaciones para tres grafos Barabasi-Albert diferentes.

Al igual que en las redes circulares, también se aprecian leves diferencias en las simulaciones. Sobre todo, en el caso en el que los grafos sean diferentes. Esto se debe a que al incluir pocas conexiones aleatorias y al ser previamente un grafo circular, cada vez que la pandemia se expande por una de estas conexiones puede llegar a duplicar el crecimiento. Este aumento de crecimiento depende de cómo de distribuidas hayan quedado las conexiones aleatorias, siendo más notable cuando se expande a zonas más alejadas de la epidemia.

Al simular estas redes podemos apreciar cambios muy significativos. En este caso la epidemia avanza más rápidamente. También aparece un número máximo de infectados claramente superior al del grafo circular sin conexiones aleatorias. Esto sucede ya que la epidemia se disgrega hacia otras zonas del grafo sana, aumentando el crecimiento de infectados. Sin embargo, es menos agresiva que las redes Barabasi-Albert o Erdős-Rényi, ya que los nodos infectados al mismo tiempo suelen ser vecinos y comparten muchas conexiones.

Comparativa de las diferentes topologías

Una vez analizadas cada una de las simulaciones por separado, es interesante contrastarlas representándolas en una misma grafica.



Comparativa de los cuatro tipos de topologías analizadas.

Como podemos apreciar, las redes que siguen una topología con hubs, como la de Barabasi-Albert, son en las que más rápido avanza la pandemia. Tienen el pico de infectados más alto que el resto, ya que al infectarse un hub, se contagian muchísimos nodos vecinos en un corto periodo de tiempo.

En segundo lugar, estarían las redes Erdős-Rényi, donde debido a su pura aleatoriedad, la pandemia se va desarrollando por toda la red, con mucha facilidad de llegar a cualquier nodo rápidamente.

En tercer lugar, estarían las redes Newman-Watts-Strogatz, donde la el crecimiento de infectados es más lento. Dependiendo de cuales sean los primeros nodos infectados, la red se ira infectando poco a poco en algunas zonas llegando a otras nuevas por las conexiones aleatorias.

Y finalmente están las redes circulares. Estas son claramente las pandemias más lentas. Los primeros nodos infectados empezarán a infectar a sus vecinos por zonas, de manera que los nodos más alejados de esos puntos serán los últimos en infectarse.

Es interesante simular para conocer cómo se propaga la enfermedad según el tipo de topología. Podría ser interesante para la población acelerar la propagación de la enfermedad para causar el mínimo número de pérdidas económicas; o intentar frenarla de manera que no se llenen los hospitales; o una mezcla hibrida. Según los intereses de la población, se podrían tomar medidas o restricciones para adaptar nuestras relaciones reales de manera que sigan una topología u otra.

5.2 Simulaciones variando la topología del grafo durante la simulación

5.2.1 Introducción

La motivación de este apartado es analizar el comportamiento de la epidemia cuando se incluye la reacción de los individuos ante una epidemia. En los modelos tradicionales el estado de los individuos no afecta a la dinámica del modelo. Sin embargo, hemos visto como cuando el número de infectados crece se adoptan, tanto desde la administración como individualmente, diferentes medidas defensivas para mitigar la propagación de la enfermedad (medidas de confinamiento, reducción de la actividad social). Estas medidas tienen un efecto equivalente a modificar los parámetros en un modelo SIR o adaptar la topología de la red en un modelo de propagación de la epidemia en grafos.

Se utilizará para estas simulaciones el modelo “base” del proyecto, esto es el modelo Erdos-Renyi, ya que implementa una red uniforme donde todos los nodos tiene en promedio el mismo número de conexiones.

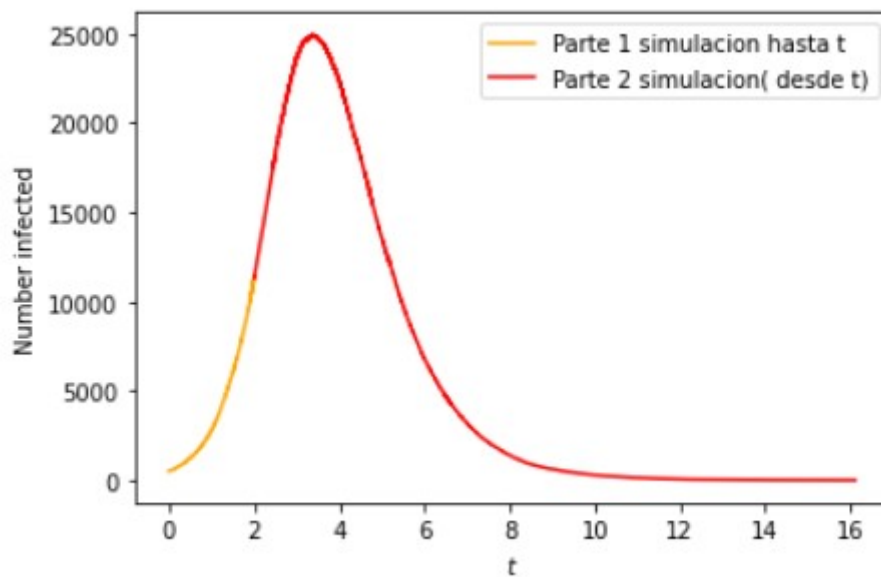
Para la reproducibilidad de los resultados de las simulaciones se ha empleado el uso de la semilla 123:

```
np.random.seed (123) #Semilla para la simulacion
```

Dado que las diferentes medidas se toman en un determinado momento t de la pandemia, esperando que esta evolucione con una tendencia descendente, nosotros haremos lo mismo. Para ello hemos desarrollado la función ***get_listados_nodos_a_tiempo(simulacion, tiempo)***, que nos permite a través de la librería de EoN utilizada, obtener el estado del grafo para cualquier instante. Esto es el estado de cada nodo, y no solo el número de nodos en cada estado. Gracias a ella lo que haremos es simular como de habitual, y llegado el instante t en el que se toman medidas, parar la simulación, editando la topología del grafo a modo de simulación de una “reducción de las relaciones entre nodos” como la disminución del número de enlaces entre nodos, y re-arrancar la simulación para ver el efecto en el cambio de la topología.

En la siguiente figura, vemos un **ejemplo**. En este caso se trata de la simulación en el grafo Erdos-Renyi, con los parámetros habituales usados en el artículo (ver sección de parámetros.) En naranja vemos la simulación de la primera parte hasta tiempo t , en la que no se han tomado medidas para restringir las interacciones sociales.

A partir de t , en la segunda parte en rojo, equivaldría al punto a partir del cual se empiezan a tomar medidas que restringen las interacciones sociales tomadas por la administración. En la figura de ejemplo, aún no se ha modificado los enlaces del grafo, por lo que como resultado vemos que la simulación final se corresponde a la simulación del grafo Erdos-Renyi del apartado anterior (esto ha sido validado para comprobar su correcta implementación).



Para dichos cambios de topología solo se modificarán los enlaces del grafo, dado que el número de nodos (población), suponemos se mantiene constante. Eliminaremos enlaces, simulando una restricción en las interacciones sociales. La eliminación de dichos enlaces, según el porcentaje de eliminados, simulará unas medidas más restrictivas que otras. Por ejemplo, una eliminación del 95% de enlaces de cada nodo podría equivalerse a una cuarentena impuesta por la administración muy restrictiva, restringida a las interacciones entre personas que tengan únicamente carácter esencial (por ejemplo, emergencias, tareas de reposición de alimentos y productos de primera necesidad).

Una eliminación de, por ejemplo, un 30% de los enlaces por nodo, podría equivalerse a unas restricciones más ligeras, donde a la población tuviese

limitadas las reuniones en interiores de más de 10 personas, o el uso de mascarillas en ámbitos públicos.

5.2.2 Parámetros

Para un primer estudio de estas características, se ha implementado la función ***elimina_enlaces(grafo,prob)***, que elimina cada enlace con una probabilidad *prob* de manera uniforme.

Con dicha implementación podemos jugar con dos parámetros que nos modifiquen el modelo y por tanto la simulación final, que son:

El tiempo *t* a partir del cual se empiezan a tomar las medias, que separa la parte 1 y 2 de la simulación final.

Podemos suponer que este parámetro, depende de un número *X*, umbral de contagiados alcanzados en cada instante *t*. Es lógico pensar que una vez alcanzado un % de contagios respecto a la población se tomen medidas globales a la población. En cambio, si ese umbral no se alcanza, el parámetro *t* no converge, y por tanto no se llegan a tomar medidas.

El porcentaje % de eliminación de cada enlace, que representa la supresión de ciertas interacciones sociales a nivel individual, según la dureza de las medidas tomadas para frenar la epidemia.

Este porcentaje depende de las durezas impuestas, dada la naturaleza de la enfermedad. Es lógico pensar que las medidas serán duras si la enfermedad es mortal, con lo que es muy probable que este porcentaje sea superior al 90% (equivalente a una cuarentena prácticamente total), mientras que, si es una gripe con efectos malignos bajos o mínimos, este parámetro tal vez sea más bien próximo a 0 (donde se le permite a la población hacer vida normal sin restricciones).

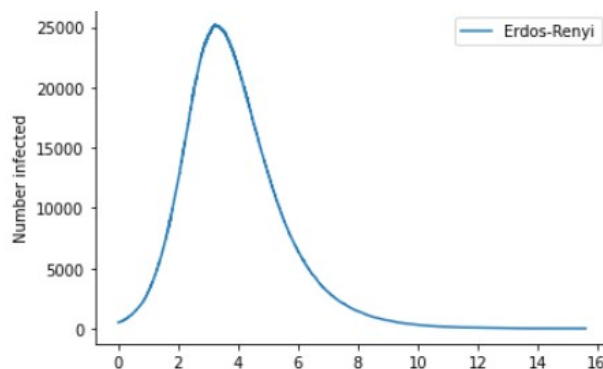
5.2.3 Simulación fijando el tiempo *t* a partir del que se toman medidas restrictivas sociales(umbral de nº de contagios).

(El código de este apartado puede encontrarse en el notebook "[Simulacion fijando t](#)")

Siguiendo con nuestro grafo base, con una población de $N = 100.000$ nodos. Podemos suponer que el instante t elegido será uno, en el que se haya alcanzado un umbral X de número de contagios. Por ejemplo, este umbral podría suponer un 10% de la población, es decir, 10.000 individuos.

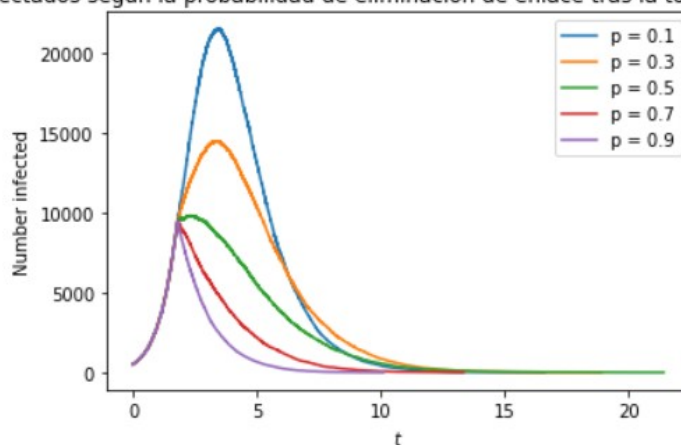
Para localizar el punto en la “simulación base” con la topología de Erdos-Renyi, utilizamos nuestra función auxiliar:

encuentra_tiempo_limite_alcanzado(tabla_tiempos, tabla_infectados, limite), llamada con un límite de 10.000 infectados, que nos arroja un resultado de $t=1.80$. Se puede observar en la gráfica de la simulación:



Por lo que fijamos el tiempo de corte en **$t = 1.80$** . Vemos cómo afecta la probabilidad de eliminación de enlace tras la toma de medidas en $t=1.8$ para diferentes probabilidades.

Numero de infectados segun la probabilidad de eliminacion de enlace tras la toma de medidas en $t=1.8$



Como observamos una probabilidad alta (medidas drásticas) tienen un efecto muy notable en el descenso de la curva de infectados. Se consigue llegar a cero en un tiempo t mucho menor (casi la mitad) y con un pico de contagios mucho menor, tanto, que de hecho esta medida hace que el pico de la curva sea el ya alcanzado en el tiempo $t=1.80$. Vemos como medidas drásticas consiguen resultados más que notables y evidentes.

Con medias menos drásticas, o lo que es lo mismo, con probabilidades de 0.7 y 0.5 observamos como el tiempo no se reduce demasiado hasta llegar a 0 en comparación con la probabilidad de 0.9, sin embargo, el pico sigue siendo prácticamente el mismo que con esta. Es decir, que con unas medidas la mitad de drásticas que las extremas, podríamos decir moderadas, conseguimos reducir el pico de contagios a más de la mitad (de 10.000 infectados a 25.000 que se corresponderían con la primera gráfica).

Con unas medidas de reducción de interacciones entre los individuos que podríamos describir como relajadas ($p=0.3$) observamos que siguen siendo bastante efectivas. Se consigue un pico de 15.000 infectados, 10.000 menos que si no se interviniera.

Vemos como estas características en un grafo que distribuye con los enlaces uniformemente son muy efectivas. Sin embargo, como sabemos, en la vida real las personas (nodos) tienden a concentrar unas más enlaces (relaciones sociales) que otras de media. Es decir, que hay personas más populares que otras, que se relacionan más que otras, y eso puede ser un factor determinante a la hora de modelar la transmisión de una enfermedad. Tal vez usar un grafo con una distribución normal de enlaces por nodo sería más acertado.

Además, también hemos supuesto que las relaciones se reducen uniformemente, cosa que en la vida real es de esperar que tampoco sea así, lo cual puede ser un factor determinante. Tal vez sería más razonable pensar que esta reducción se hace de acuerdo a una distribución exponencial.

5.2.4 Simulación fijando el porcentaje de eliminación aplicado a diferentes tiempos t de la pandemia.

5.2.4.1 Aplicando medidas sociales restrictivas moderadas (50% de reducción)

(El código de este apartado puede encontrarse en el notebook "[Simulacion fijando porcentaje eliminacion](#)")

En el apartado anterior suponíamos que el instante t era elegido una vez que se alcanzaba un umbral de un 10% de la población. ¿Qué pasaría si ese

umbral se hubiera elegido con un 5%, o tal vez con un 30%? Vamos a estudiarlo.

Para ello vamos a fijar unas **medidas restrictivas moderadas**, que podríamos definir como limitar las relaciones de los individuos al 50%(es decir, probabilidad de eliminar un enlace del 50%).

Primero vamos a encontrar el umbral t para el cual se alcanzan los mínimos a partir de los cuales las administraciones aplicarían dichas medidas moderadas limitando los contactos sociales. Para ello vamos a hacer uso de nuestra función

encuentra_tiempo_limite_alcanzado(tabla_tiempos,tabla_infectados, limite),

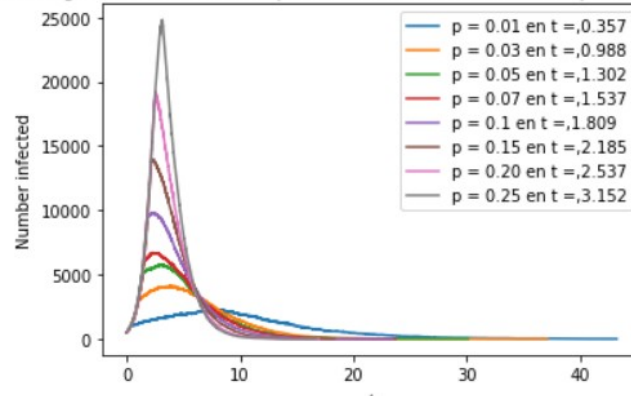
Llamándola con un límite que represente a un porcentaje de la población determinado, encontraremos en qué tiempo t se alcanza en la pandemia de forma natural dadas las condiciones iniciales base que utilizamos en todas las simulaciones del proyecto.

Porcentaje sobre la población total	Número de individuos sobre N=100.000	Tiempo t en el que se alcanza esse número de contagios
1%	1000	0.357
3%	3000	0.988
5%	5000	1.302
7%	7000	1.537
10%	10000	1.809
15%	15000	2.185
20%	20000	2.537
25%	25000	3.152
30%	30000	Limite no alcanzado nunca

(Tabla obtenida sobre la simulación original base del grafo Erdos-Renyi, tiempos redondeados a 3 decimales)

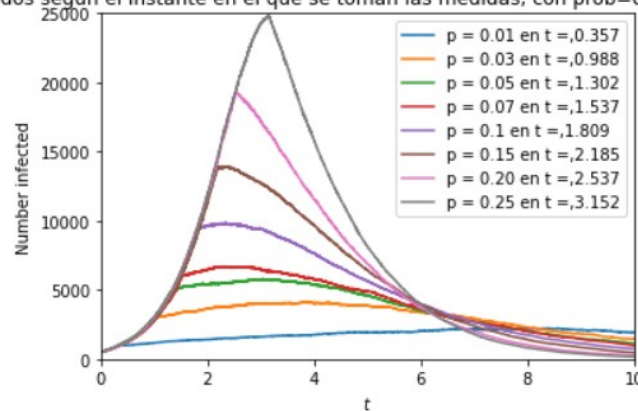
OJO: No confundir p = porcentaje de la población infectada, con la probabilidad de eliminación de enlaces en el grafo

Numero de infectados segun el instante en el que se toman las medidas, con prob=05 de eliminacion de enlaces



Ampliamos los tiempos en $t=[0,10]$ para apreciar mejor la comparativa:

Numero de infectados segun el instante en el que se toman las medidas, con prob=05 de eliminacion de enlaces



En las gráficas no se ha representado $p=30\%$ de la población, dado que el límite de los 30.000 individuos nunca se llega a alcanzar, y el resultado es el mismo que la simulación sobre el grafo baso de Erdos-Renyi (ver apartados anteriores). Es decir, que si fijamos el límite para actuar solo si se superan los 30.000 contagios, dado que nunca se alcanzan, no se llegan a tomar medidas sociales.

Vemos que, si ponemos ese límite cerca, sobre las 25.000 personas, apenas tienen efecto unas medidas moderadas, ya que el pico de la pandemia ya se ha alcanzado, y las diferencias de tiempos en lo que dura la pandemia son también inapreciables.

En cambio, vemos que, si empezamos a tomar medidas moderadas que afectan a las relaciones sociales con un número muy bajo de individuos, como podrían ser los 1.000 o 3.000 individuos (1% y 3% de la población), se consigue una curva de contagios prácticamente plana. **Vemos como empezar a tomar medidas lo antes posible, tiene un efecto muy importante a la hora de estabilizar la pandemia, y es un factor muy determinante.**

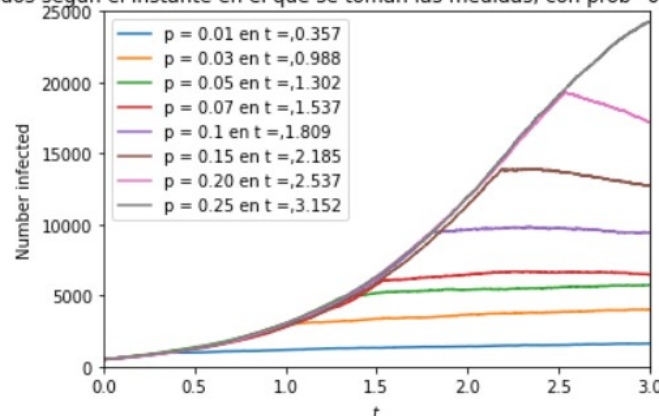
Como es de esperar, si tomamos medidas cuando el porcentaje de la población es más alto, mayores picos de contagio alcanzaremos. Observamos como a medida que este límite es mayor, la curva de infectados pasa de ser plana, a tener forma de curva normal, cada vez con un comportamiento más picudo. **Es decir, tomar medidas tempranas tiene como ventaja reducir el número de contagios, pero en consecuencia la pandemia se prolonga más en el tiempo. En cambio, dejar el límite de infectados sobre el que se toman medidas más alto, consigue un mayor número de contagios, pero que la pandemia se extienda menos en el tiempo.**

Si tomamos unos porcentajes de contagios medios, como podrían ser el 10% o el 15% de la población contagiada, vemos como es de esperar un compartimento intermedio, como es de esperar.

Como podemos apreciar en la siguiente gráfica ampliada, en todas las curvas se cumple también la siguiente propiedad: En el instante en el que se empiezan a tomar medidas, el número de contagios empieza a disminuir y nunca vuelve a alcanzar ese valor, siendo este máximo. O dicho en otras palabras, **una vez se empiezan a tomar medidas**, en este caso moderadas, se consigue que el pico máximo de contagios sea el mismo que se tenían en el momento en que se aplican dichas medidas. Es decir, una vez aplicadas las medidas, la curva de contagios no sigue creciendo, sino todo lo contrario, **empieza siempre a decrecer, sea cual sea el estado de la pandemia.**

Esto quiere decir, que dichas medidas de reducción de interacciones sociales moderadas (reduciendo cada individuo a la mitad sus relaciones sociales), es un factor muy determinante para frenar los contagios.

Numero de infectados segun el instante en el que se toman las medidas, con prob=05 de eliminacion de enlaces

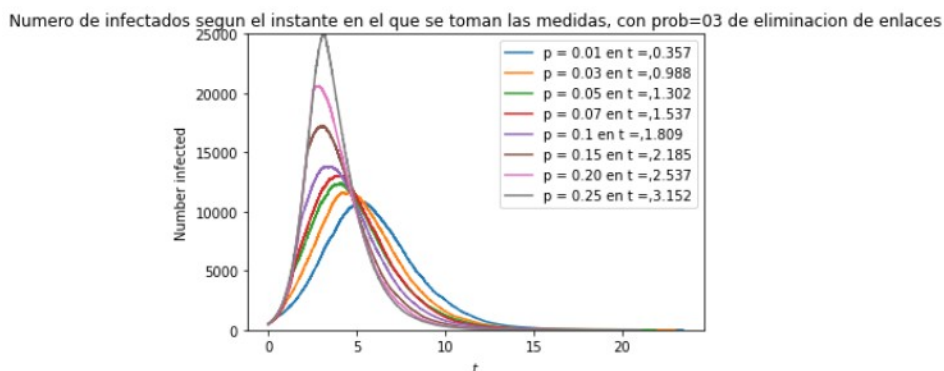


En la gráfica anterior podemos verlo con un ejemplo. Véase la curva marrón, donde ese empiezan a tomar medidas cuando el 15% de la población ha sido infectada. Esto ocurre en el tiempo $t = 2.185$. A partir de ahí se toman medidas desde la administración, y la población reduce sus interacciones sociales a la mitad, dado el estado de la pandemia. Con estas medidas lo que se consigue es que el número máximo de infectados a lo largo de la pandemia nunca crezca por encima de ese 15% de la población, dado que el crecimiento de infectados se revierte y se consigue que dicha curva se vuelva decreciente en el tiempo hasta llegar a 0.

5.2.4.2 Aplicando medidas sociales restrictivas ligeras(30% de reducción)

(Los resultados de dicha simulación se encuentran en el notebook: "Simulacion_fijando_porcentaje Eliminacion_2")

Probamos a repetir la simulación, pero ahora con unas medidas más "ligeras" de restricciones sociales, en la que se toman medidas que solo llegan a **reducir las relaciones sociales en un 30%**(esto es 30% de probabilidad de eliminar un enlace). Este es el resultado de la simulación:

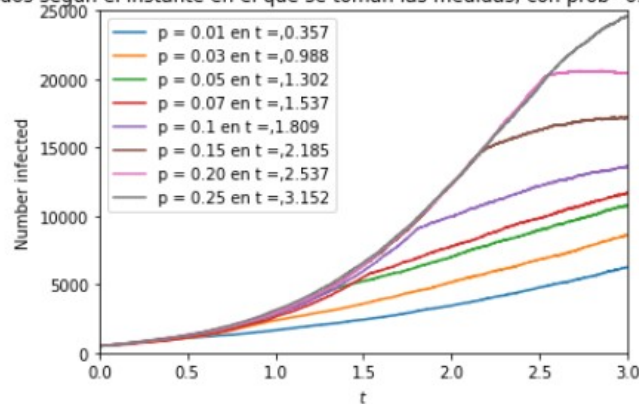


Observamos que ahora incluso para porcentajes de población más bajos para los que se empiezan a tomar medidas más bajas, las curvas ya no son tan planas, como era de esperar, y en cada curva se alcanzan picos de infectados bastante más altos. Es decir, a la hora de decaer la curva, lo hace con una pendiente menor.

Por ejemplo, si se empiezan a tomar medidas cuando el 1% de la población ha sido infectada, con las relaciones sociales reducidas al 50% el pico máximo alcanzado está por debajo de 2500 infectados (curva casi plana),

mientras si las relaciones sociales se reducen un 30%, el número de infectado supera los 10.000. Esta diferencia es más que razonable.

Numero de infectados segun el instante en el que se toman las medidas, con prob=03 de eliminacion de enlaces



También observamos que, en este caso, a la hora de aplicar restricciones sociales, **la curva ya no decae directamente, si no que ahora el número de contagios sigue creciendo un poco más hasta llegar a un pico y luego decae**. Véase el caso de la curva marrón con $p=15\%$ de la población contagiada, descrita anteriormente.

Observamos que este **parámetro** es muy determinante a la hora de detener el crecimiento de la curva de contagios y de **determinar la velocidad con la que empieza a decaer esta hasta llegar finalmente a 0**, cuando se detiene la pandemia.

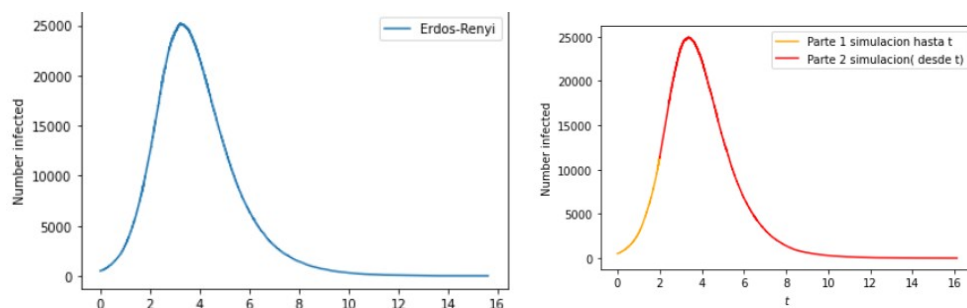
Observamos como ambos parámetros que afectan a la topología de nuestro modelo, son bastante influyentes en la curva final de infectados de la pandemia.

6. Validación

6.1 Validación de la implementación del modelo

Dado que nos hemos mayormente en librerías ya implementadas, como son EoN(Epidemics on Networks), que a su vez trabaja nativamente con NetworkX para los grafos, obviamos la validación de esta parte, de acuerdo a los ejemplos ya vistos, ya que en la validación del modelo además obtenemos los resultados esperados.

Sin embargo, si debemos validar los modelos de simulación partida, acorde a tomar medidas en las relaciones sociales individuales entre individuos. Esto ya lo hicimos en el apartado 5.2 de la memoria, en la que comprobábamos tanto gráficamente como a nivel numérico, que nuestra simulación partida coincidía con los resultados de la librería, si no modificábamos ningún enlace del grafo:



Comparativa gráfica de ambas salidas

Para consultar esta validación con mayor detalle, consultar el apartado 5.2.

Obviaremos el desarrollo de este apartado por considerarlo correcto, y de acuerdo a la extensión del proyecto.

6.2 Validación del modelo

Se validará en primer lugar el modelo “base” utilizado para la práctica, esto es, la simulación del grafo Erdos-Renyi, sin parada para tomar medidas sociales, modificando los enlaces del grafo.

En segundo lugar, se intentará validar el modelo, con modificaciones en los enlaces del grafo, es decir, cuando cambia la topología de éste.

Para validar nuestros modelos, hemos utilizado el modelo **ODE SIR** de campo medio, basado en ecuaciones diferenciales:

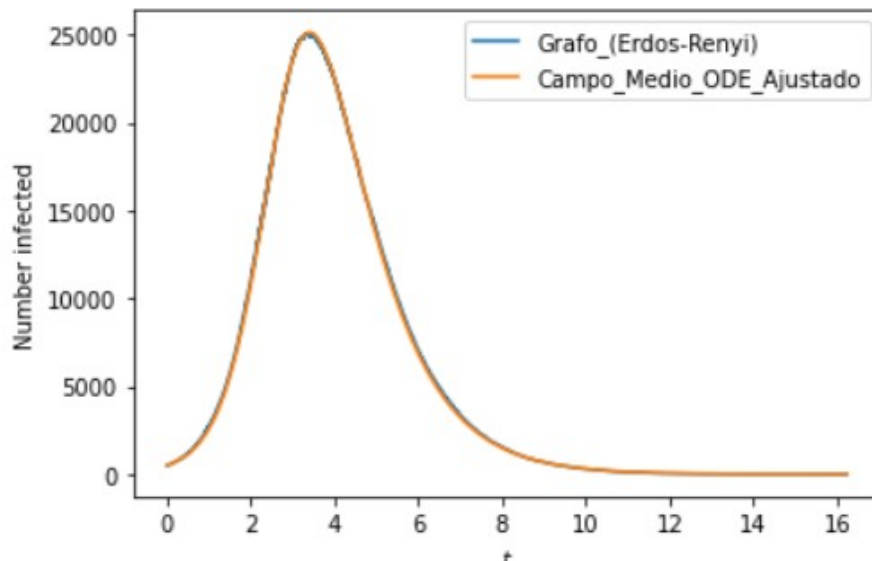
$$\begin{aligned}\frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= +\beta SI - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

Donde S,I,R corresponden a los Susceptibles, Infectados y recuperados, y los parámetros beta y gamma a la tasa de transmisión y de recuperación.

6.2.1 Validación del modelo base, sin modificaciones de las relaciones individuales.

(El código de este apartado puede encontrarse en el notebook "[Validacion_modelo_ODES_SIR](#)")

Utilizando para ello el ejemplo base del proyecto, esto es, el grafo Erdos-Renyi, encontramos primero los parámetros óptimos para el ODE-SIR, con la librería Scipy y la función `curve_fit` (`scipy.optimize.curve_fit`)[3]. Obteniendo una curva de infectados tal que así:



Que de manera visual se ajusta muy bien. Con unos valores de los parámetros beta y gamma tal que:

```
beta_ajustado: 2.7488424431027747e-05  
gamma_ajustado: 1.0530871519673743
```

6.2.2 Test de Kolmogorov-Smirnov

Para validar como tal el modelo, utilizamos el test no paramétrico de **Kolmogorov-Smirnov**, con la hipótesis H_0 , de que ambos modelos se pueden ajustar de acuerdo a la misma distribución para el número de contagiados. Obtenemos los siguientes valores (estadístico y pvalue):

```
KstestResult(statistic=0.053291390766842195, pvalue=1.7770591056720215e-213)
```

Observamos cómo hemos obtenido un estadístico pequeño y en consecuencia un pvalue alto. Es decir, que la distancia máxima entre ambas curvas es pequeña. Indicación de que hemos pasado el test de Kolmogorov,

como intuíamos de manera gráfica, **y podemos aceptar nuestra hipótesis H0 de que ambas curvas siguen la misma distribución.**

6.2.3 Valor esperado e intervalo de confianza.

Podemos considerar la variable aleatoria “número de infectados que hay en un instante t desde el inicio de la pandemia”. Para esa v.a. no interesaría saber cuál es su valor esperado, es decir, su esperanza. De acuerdo a la distribución de los datos, esta sería su media.

Dado que la media muestral es un buen estimador de la media de la distribución de la v.a, obtenemos empíricamente su valor a partir de la simulación:

$$\text{Media_muestral} = 16530.956093548146$$

Sobre un tamaño de muestra de $n = 172617$. Para dicha media muestral ($\mu_{\text{estimado_sim}}$), obtenemos un intervalo de confianza tal que:

$$16495.965732979348 \leq \mu_{\text{estimada_sim}} \leq 16565.946454116944$$

Para el caso del modelo ODE-SIR, obtendríamos para el mismo tamaño de muestra, un estimador situado en el intervalo de confianza tal que:

$$16448.875936645803 \leq \mu_{\text{estimada_ODE}} = 16484.346252448788 \leq 16519.816568251772$$

6.2.4 Validación del modelo base, con modificaciones de las relaciones individuales.

(El código de este apartado puede encontrarse en los notebooks:

-Simulacion_fijando_t_validacion_p_20

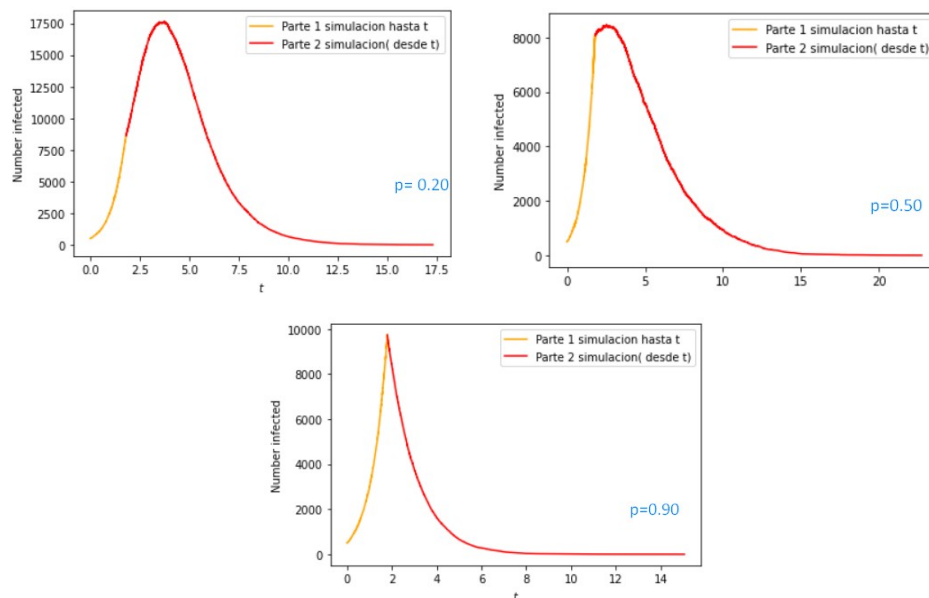
-Simulacion_fijando_t_validacion_p_50

-Simulacion_fijando_t_validacion_p_90)

Vamos a validar, también con el modelo ODE-SIR, como se ajusta a las curvas de infectados de nuestro modelo, cuando se toman medidas de restricciones sociales sobre la población. Dado el alcance del proyecto, lo vamos a hacer para el único caso, donde el instante de corte de la simulación va a ser **$t=1.80$** , el mismo instante que se hizo para el estudio en el apartado 5.2.3 de este documento.

Elegimos unas **probabilidades de eliminación de enlaces de 20%, 50% y 90%** (de acuerdo a los nombres de los notebooks de esta sección, en los que se encuentran los respectivos códigos.) El motivo de la elección de varios porcentajes se explica a continuación.

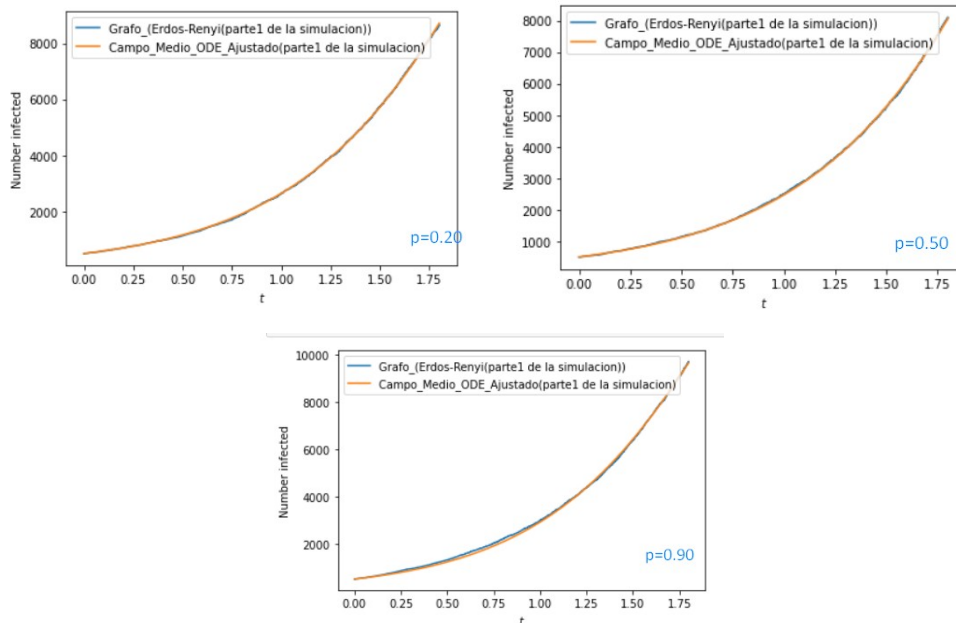
Partiendo de una simulación en $t = 1.80$, obtenemos una curva final de infectados tal que así, para los distintos porcentajes:



Para validar dichas curvas, buscamos unos parámetros que se ajusten con Scipy, de la misma manera que se hizo en el apartado anterior. Esto lo hacemos con dos curvas distintas para las dos partes de las simulaciones, dado que, en el corte de dichas simulaciones, al cambiar la topología del grafo, es como si modificásemos los parámetros del modelo ODE-SIR.

6.2.4.1 Validación parte 1 de la simulación.

Intentamos ajustar la primera parte de la simulación (línea naranja en las gráficas superiores):



En las 3 gráficas, vemos como conseguimos encontrar unos parámetros óptimos para el modelo ODE-SIR, que se ajustan a nuestro modelo en los tiempos de la parte 1 de la simulación, como era de esperar, ya que en esta parte aún no se han hecho modificaciones sobre el grafo, y ya vimos en el apartado superior, como obteníamos un resultado satisfactorio.

En ambas obtenemos el visto bueno por parte del test de Kolmogorov, obteniendo en los 3 resultados muy similares, con un estadístico muy pequeño. Incluimos únicamente por ello los resultados con $p=0.20$:

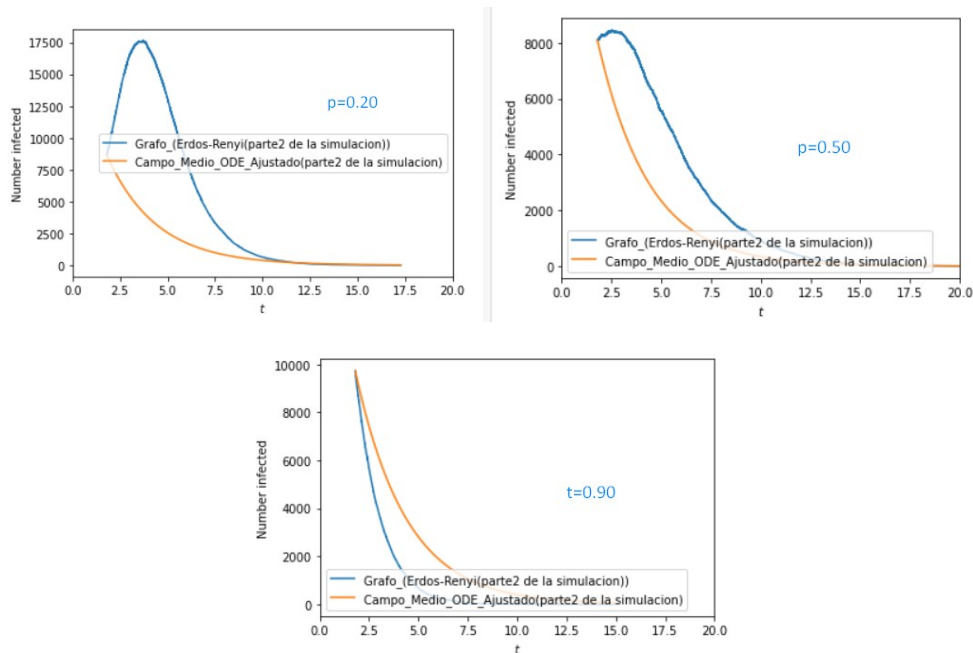
```
KstestResult(statistic=0.009984679592160206, pvalue=0.3000140620991052)
```

6.2.4.2 Validación parte 2 de la simulación.

Intentamos encontrar otros posibles parámetros β y γ , que consigan ajustar una curva ODE-SIR, a nuestro modelo.

Para ello inicializamos el modelo ODE-SIR con el número de Susceptibles, Infectados y Recuperados que deja cada uno de los 3 ejemplos en $t=1.80$. Posteriormente integramos el modelo desde $t=1.80$ a $t=20$.

Obtenemos las siguientes gráficas:



Ya podemos observar de manera gráfica, como en ninguno de los 3 casos conseguimos encontrar unos parámetros óptimos que consigan validar nuestro modelo. En consecuencia, obtenemos los siguientes resultados para el test de Kolmogorov:

$P=0.20$: `KstestResult(statistic=0.8514630282742542, pvalue=0.0)`

$P=0.50$: `KstestResult(statistic=0.4273636529885914, pvalue=0.0)`

$P=0.90$: `KstestResult(statistic=0.26318819719323494, pvalue=0.0)`

Con un estadístico tan alto observamos que ninguno de ellos nos sirve para validar nuestro modelo.

Sin embargo, como ya podíamos apreciar gráficamente, observamos que cuanto más alto es el porcentaje de eliminación, mejor se ajusta el modelo ODE-SIR. Observamos como la curva de $p=0.90$ es mucho más parecida a la de validación que la de $p=0.20$. Un valor del estadístico mucho más bajo (0.26), para el primer caso y más alto para el primero (0.85), nos confirma este comportamiento.

Es decir, podemos ver como a mayor porcentaje de enlaces eliminados, más rápido decae la curva de contagios, como ya habíamos comprobado en apartados anteriores. Sin embargo, observamos como **un cambio más brusco de tendencia en el crecimiento de dicha curva, es captado mejor por el modelo ODE-SIR de campo medio**. Es decir, aprecia mejor un cambio más evidente y brusco (90% de eliminación) mientras que no lo

hace con uno mucho más sutil (20%). Esto es lógico, debido a que dicho modelo de ecuaciones diferenciales se base en “medir” la velocidad de cambio en el crecimiento de la curva de infectados. Cambios más bruscos, son captados mejor, que cambios más suaves.

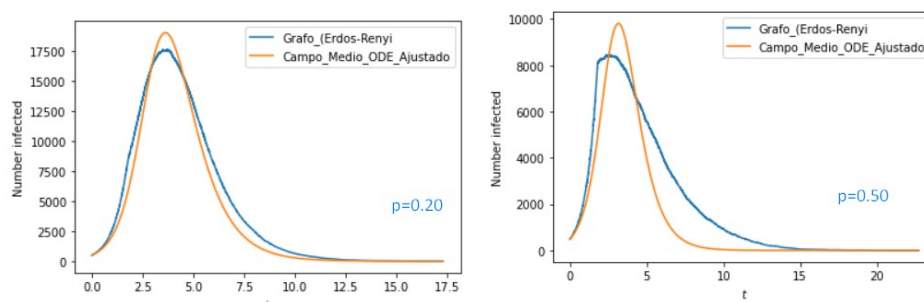
Dado que estos cambios tienen que ver con modificaciones a nivel individual de las relaciones sociales, es lógico esperar que un modelo de campo medio no se ajuste bien, dado que tiende a ver de manera global, un cambio que afecta a nivel local de cada individuo, y por ello para estos casos, no sea la mejor opción de validación, cuando muchos de estos cambios producidos sobre el conjunto son más bien sutiles. Podemos observar esto con mayor detalle en el siguiente apartado.

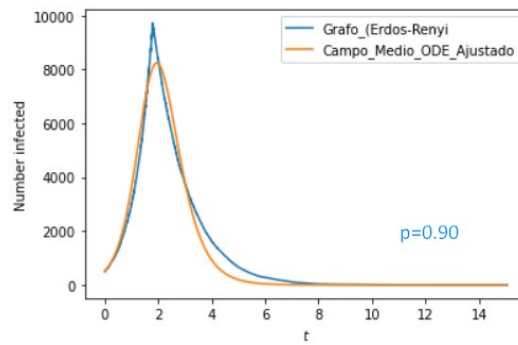
6.2.4.3 Validación sobre la simulación total.

En los 2 apartados anteriores, intentábamos ajustar dos curvas distintas para cada una de las dos partes de la simulación (antes de tomar medidas y después de tomar medidas sociales).

Llegamos a la conclusión de que, con nuestro un modelo que maneja las relaciones sociales entre los individuos, no encaja bien como modelos deterministas basados en ODES, en este caso, el modelo ODE-SIR. Esto es debido a que los modelos **ODES tienen una visión global sobre el conjunto de individuos, y no captan bien lo que las interacciones individuales hacen, si no tienen un efecto bastante brusco y evidente sobre el conjunto de la población.**

Vamos a ver esto sobre la curva total de infectados, siguiendo el flujo del apartado anterior. Para ello vamos a ajustar la curva del ODE-SIR a la curva final de nuestra simulación, “sin cortes”:





Con unos estadísticos del test de Kolmogorov de 0.22 , 0.22 y 0.16 respectivamente. Siendo el de 0.16 para la probabilidad de $p=0.90$. Observamos de manera global, como el mejor ajuste se consigue cuando hay cambios bruscos, como es este último caso, y a pesar de ello, no es un ajuste suficientemente bueno como para poder utilizarlo de validador de nuestro modelo.

7. Conclusiones

En este proyecto analizamos las dinámicas de un SIR estándar con dinámicas Markovianas. Utilizando para ello un modelo basado en grafos para simular el efecto de las relaciones sociales individuales en la propagación de una epidemia entre la población. De esta manera estudiamos el efecto de diferentes topologías de red sobre la curva de infectados de la población, además del comportamiento de dicha curva cuando se incluye la reacción de los individuos ante una epidemia.

Con unos parámetros de Tau y gamma fijados en unos valores de 0.3 y 1.0 respectivamente, sobre un grafo de una población $N=100000$ nodos.

En el estudio de las diferentes topologías, se puede apreciar en las gráficas las diferencias en las evoluciones de las pandemias. Fijando un mismo numero de nodos y conexiones, observamos que el factor mas relevante en el aumento del crecimiento de infectados y de la pandemia, es la aparición de hubs. Por otro lado, las redes en las que los nodos comparten varios vecinos ralentizan el crecimiento de infectados, sin llegar a picos elevados pero alargando la duración de la epidemia. Y en medio tendríamos las redes puramente aleatorias, donde la pandemia crece y termina de una manera intermedia.

En las simulaciones en las que variábamos la topología del grafo (en un grafo Erdos-Renyi), simulando restricciones sociales, nos encontrábamos con dos parámetros principales ajustables:

- Tiempo a partir del cual se aplican medidas sociales.
- Dureza de las medidas en las restricciones sociales

Resultando ambos parámetros ser muy determinantes en la evolución de la pandemia: afectando de manera importante al pico máximo alcanzado de infectados, a la velocidad con la que empezaba a decaer dicha curva y a la prolongación en el tiempo de la pandemia.

Finalmente, conseguimos validar nuestro modelo, con uno de campo medio ODE-SIR, únicamente cuando no se tomaban medidas que afectaras a las relaciones individuales, debido a que dicho modelo de validación no captaba bien los cambios que se producen a nivel individual sobre el conjunto, a no ser que estos fueran muy brutos y evidentes.

8. Bibliografía

[1] Keeling Matt J and Eames Ken T.D, Networks and epidemic models, J. R. Soc. Interface. (2005) 2295–307 <http://doi.org/10.1098/rsif.2005.0051> T. Gross, C.J.D. D’Lima, B. Blasius, Epidemic dynamics on an adaptive network Phys. Rev. Lett., 96 (20) (2006), p. 208701

[2] Leitch, J., Alexander, K.A. & Sengupta, S. Toward epidemic thresholds on temporal networks: a review and open questions. Appl Netw Sci 4, 105 (2019). <https://doi.org/10.1007/s41109-019-0230-4>

[3] scipy.optimize.curve_fit — SciPy v1.6.0 Reference Guide
https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

[4] Graph generators — NetworkX 1.10 documentation <https://networkx.org/documentation/networkx-1.10/reference/generators.html>

[5] The SIR epidemic model
<https://scipython.com/book/chapter-8-scipy/additional-examples/the-sir-epidemic-model/>

[6] Welcome to Epidemics on Networks’s documentation! — Epidemics on Networks 1.2rc1 documentation
<https://epidemicsonnetworks.readthedocs.io/en/latest/>

[7] <https://arxiv.org/pdf/2001.02436.pdf>

9. Anexos

[1] proyecto.ipynb

[2] Simulacion_fijando_porcentaje Eliminacion.ipynb

[3] Simulacion_fijando_porcentaje Eliminacion_2.ipynb

[4] Simulacion_fijando_t.ipynb

[5] Simulacion_fijando_t_validacion1.ipynb

[6] Simulacion_fijando_t_validacion_p_20.ipynb

[7] Simulacion_fijando_t_validacion_p_50.ipynb

[8] Simulacion_fijando_t_validacion_p_90.ipynb

9.1 Principal listado de funciones y código de implementación propia para el proyecto.

Descartando el código necesario de llamadas a diferentes librerías python ya implementadas, el siguiente desarrollo complementario de código de autoría propia ha sido necesario para realizar las simulaciones y estudiar los resultados(cabe destacar que todo el código perteneciente a los notebooks anexos se ha tenido que escribir igualmente para la realización del proyecto):

1. Get_listado_nodos_a_tiempo(simulacion, tiempo)

Necesaria dada la naturaleza de la librería de simulaciones empleada, necesitamos obtener qué nodos son los que se encuentran en estado S,I o R en el instante determinado que deseemos, necesario para partir la simulación, modificar el grafo y reanudarla.

2. Elimina_enlaces(grafo,prob)

Esta función, elimina enlaces del grafo de acuerdo a una probabilidad dada. Es la única forma que hemos implementado de modificar enlaces para las simulaciones. Se ajusta a la librería utilizada.

3. Simulación Partida

En este caso no se trata de una función como tal, pero dicho código de simulación partida está repartido y repetido con diferentes parámetros y opciones sobre los diversos notebooks. El hecho de no hacerlo función era para simplificar el tiempo dedicado a ello, para no quitárselo a otros apartados igualmente importantes del proyecto.

Básicamente, parte la simulación en 2 partes (2 simulaciones), recoge y organiza los datos de manera coherente para su uso posterior en los estudios y representaciones. para representar. Esto ha sido necesario hacer así, dada la naturaleza de la librerías utilizadas (EoN y NetworkX). Se puede ver un ejemplo de su uso en el notebook de “proyecto”.

4. Simulacion_fijando_porcentaje Eliminacion

En estos notebooks se modifica el código anterior para poder realizar varias simulaciones simultaneas y recogiendo los datos requeridos. Todo ello utilizando como base el código descrito arriba.

5. Simulacion_fijando_t

Similar al caso anterior. Algunos cambios concretos eran necesarios para poder fijar el tiempo y poder hacer varias simulaciones simultáneas junto a su representación.

6.

Encuentra_tiempo_limite_alcanzado(tabla_tiempos,tabla_infectados_limite)

Encuentra el primer tiempo t a partir del cual el número de infectados \geq limite. Utilizada para el estudio de las simulaciones, en el momento en el que se decide que, una vez alcanzado un porcentaje de contagios entre la población, se toman medidas sociales.