

PRÁCTICA 6**1 Modelo Gilbert (1959)**

1.1 Implementar en una función de python el algoritmo de generación de redes aleatorias de **Gilbert $G(N,p)$** :

- 1) Start with N isolated nodes.
- 2) Select a node pair and generate a random number between 0 and 1. If the number is lower than p , connect the selected node pair with a link, otherwise leave them disconnected.
- 3) Repeat step (2) for each of the $N(N-1)/2$ node pairs

1.2 Para una red generada mediante el modelo anterior $G(100,0.005)$ calcular los estadísticos:

- Número de nodos, número de enlaces
- Grado máximo, mínimo y medio
- Clustering medio de los nodos (ver Networkx function [average_clustering](#))

2 Distribución de grado

2.1 Represente la distribución de grado de distintas realizaciones del modelo $G(100,p)$ para los siguientes valores de $p=\{0.005,0.05,0.5\}$. ¿Qué distribución de probabilidad sigue la distribución de grado?

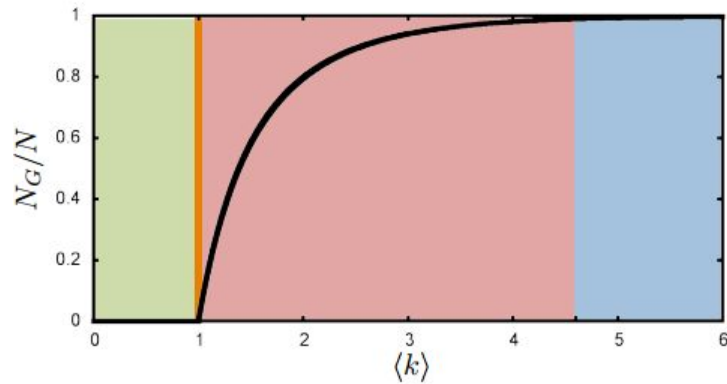
Recuerde la función: [matplotlib.pyplot.hist\(\)](#)

3 Aparición del componente gigante

3.1 Calcule la variación de N_G/N (tamaño del componente mayor N_G dividido por el número de nodos N) cuando variamos el grado medio $\langle k \rangle$ de una red $G(1000,p)$ y represéntelo gráficamente de forma semejante a la figura. Para ello:

- 1) Defina un conjunto de valores de $\{\langle k \rangle\}$ entre 0 y $\ln(1000)^1$. La función de numpy [linspace](#) permite dividir un rango en partes iguales.
- 2) Para cada valor $\langle k \rangle$ genere la red $G(1000,p)$ y calcule el tamaño del componente mayor N_G ; replique este proceso un número de veces y calcule el valor medio de N_G
- 3) Represente gráficamente los valores de $\{\langle k \rangle\}$ frente a los N_G/N calculados.

¹ Es el límite para el que la red aleatoria está completamente conectada, donde $N_G/N=1$



Tenga en cuenta los siguiente:

- En redes aleatorias de Erdős-Rényi/Gilbert el grado medio esperado de la red es $\langle k \rangle = Np$, por lo que conocido k_j **la probabilidad p_j** resulta ser $p_j = k_j/N$.
- Recuerde que un modelo $G(1000, p)$ es un modelo estocástico, por lo que para definir con precisión estadística la gráfica anterior cada modelo concreto $G(1000, p_j)$ **deberá replicarse un número de veces** y computar los valores medios de N_G/N para las diferentes realizaciones.
- La función de Networkx [connected_component_subgraphs\(G\)](#) devuelve un iterador sobre los componentes de la red. Una forma sencilla de obtener una lista del tamaño de los componentes de una red G podría ser:
`[Gc.number_of_nodes() for Gc in connected_component_subgraphs(G)]`