

# **Transició de fase i components connexes en grafs aleatoris.**

Grup 8:

QUINTANA TORRES, JOSEP

BOYANO IVARS, PAULA

RUBINSTEIN PÉREZ, FEDERICO

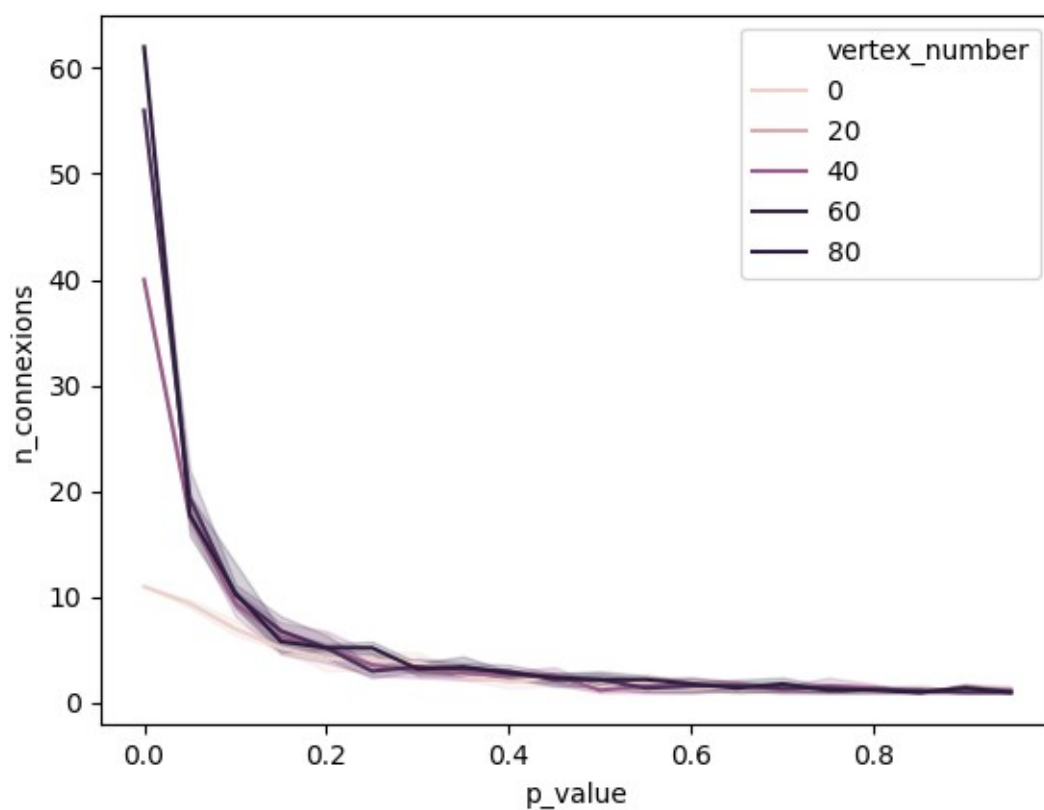
MARQUÉS GODÓ, JORDI

## Experiments :

Per obtenir totes les dades hem realitzat experiments molt similars.

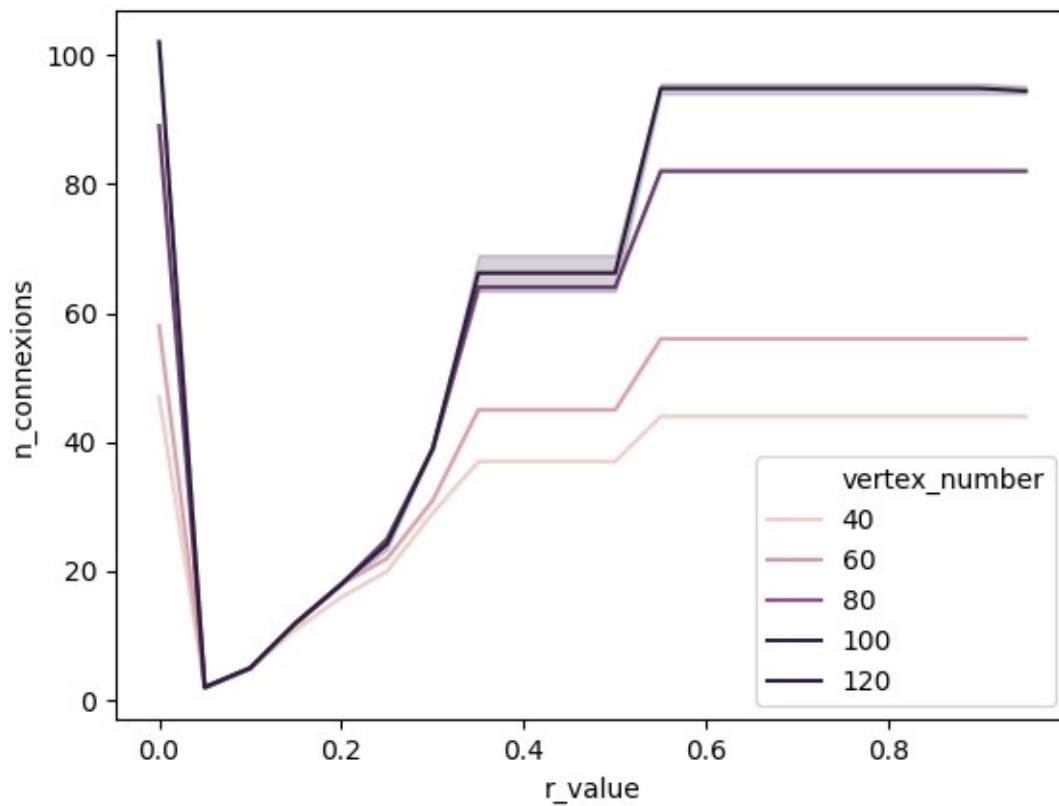
Per tots els casos hem agafat 5 numeros de vertex aleatoris i hem probat diferents valors variables, com ara la probabilitat de adjacencia o el radi. A més, per cada valor del vertex hem repetit multiples vegades el mateix experiment per minimitzar els resultats inesperats causats per l'aleatorietat.

## Graf(n,p):



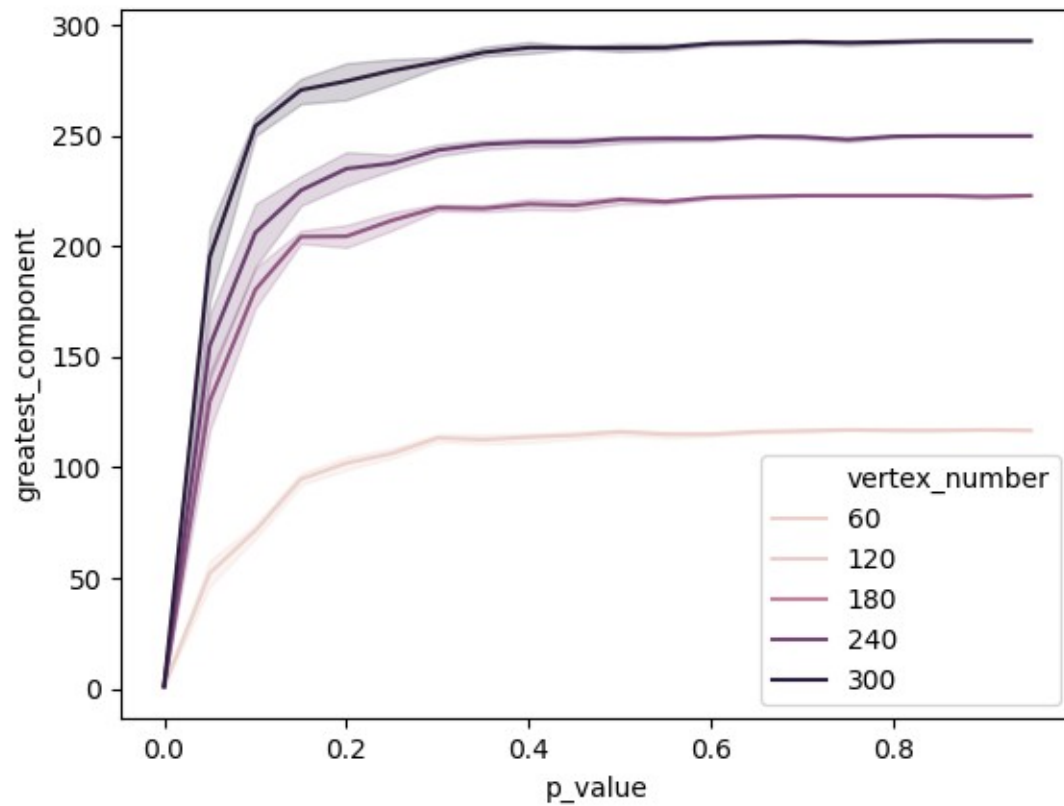
En aquest grafic podem observar el resultat de la mostra, on podem veure que tot i les lleugeres diferencies, les diferents mostres practicamente tenen un valor de transició de fase molt similar.

Graf(n,r):



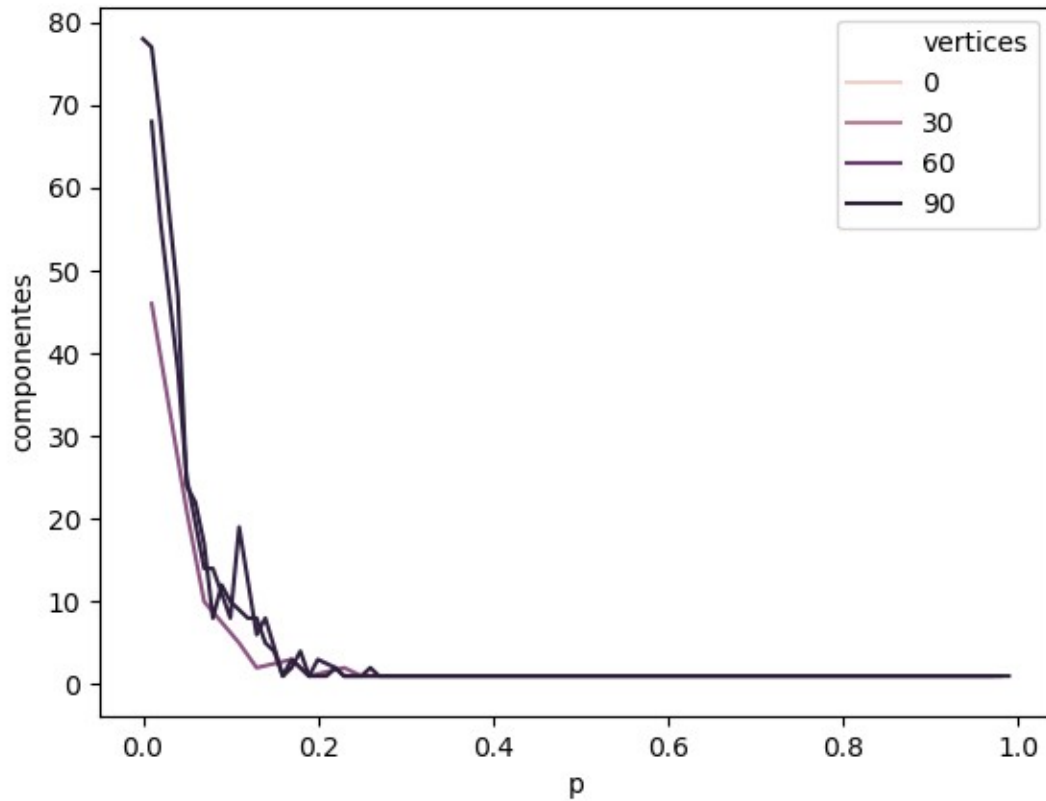
En aquest cas podem observar que el numero de components connexes i el valor de  $r$  son inversament proporcionals. Com major es la distancia entre vertex, més numeros de components. Pero novament podem veure que la transició de fase es molt similar independentment del tamany del graf.

## Components Connexes Gegants:



En aquesta ocasió em probat grafs més grans. Tot hi així seguim en lo mateix, la transició de fase es molt similar, inclus en les components connexes gegants.

## Graf Extra:



En aquest cas hem decidit provar els grafs creat per "*uniform\_random\_intersection\_graph*" de la llibreria *networkx*. Com era d'espera tornem a trobar una transició de fase molt similar als casos anteriors.

## Replicar Experiments:

Per realitzar una replica dels experiments que hem realitzat es necessari contar amb el compilador g++, i en la carpeta realitzar primer un `#make`, i despres executar el `#main.x`.

Per obtenir els experiments extra i els grafics, es necessari python3, pandas, seaborn, matplotlib i networkx.

Per executar cal la comanda `#python3 grafics.py`