Network Science Homework 2

Arghya Kannadaguli (ak5357)

2025-02-09

Assignment Description: Run the random graph model, vary the connection probability p, examine the impact of p on the size of the largest component and the number of components.

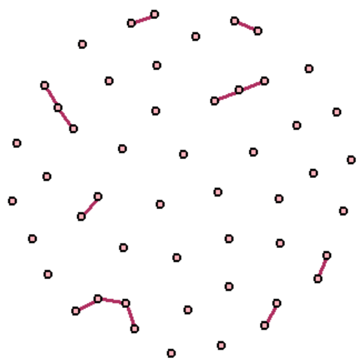
Generate Random Graphs

The following code block generates random graphs using the function `sample_gnp()` , which creates random graphs according to the $G(n, p)$ Erdős-Rényi model. I will use this function to create several networks, varying the parameter p which denotes probability for drawing an edge between two arbitrary vertices.

```
set.seed(2025)

plot_random_graph = function(p, n=50){
  sample_gnp(n, p) |>
    plot(vertex.label = NA, vertex.size = 5, vertex.color = "lightpink", edge.width = 1.5, edge.
color = "maroon")
}
```

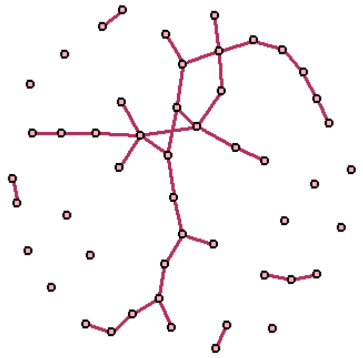
Here is a random graph with $n = 50$ and $p = 0.01$.

```
plot_random_graph(0.01)
```



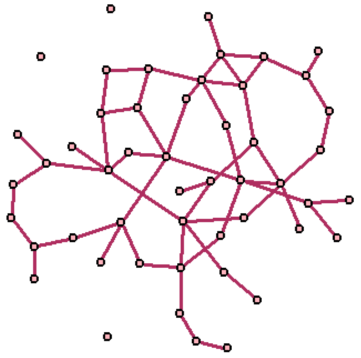
Here is a random graph with $n = 50$ and $p = 0.03$.

```
plot_random_graph(0.03)
```



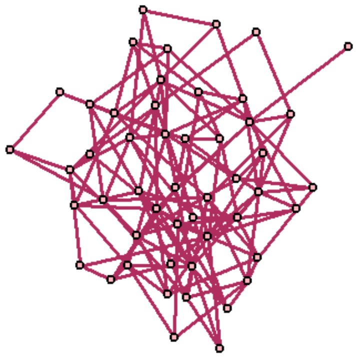
Here is a random graph with $n = 50$ and $p = 0.05$.

```
plot_random_graph(0.05)
```



Here is a random graph with $n = 50$ and $p = 0.1$.

```
plot_random_graph(0.1)
```



Based on these initial visualizations, I would hypothesize that as the value of p increases, the size of the largest component in a network increases and the number of components decreases. Let's check this out on a larger scale.

Varying p

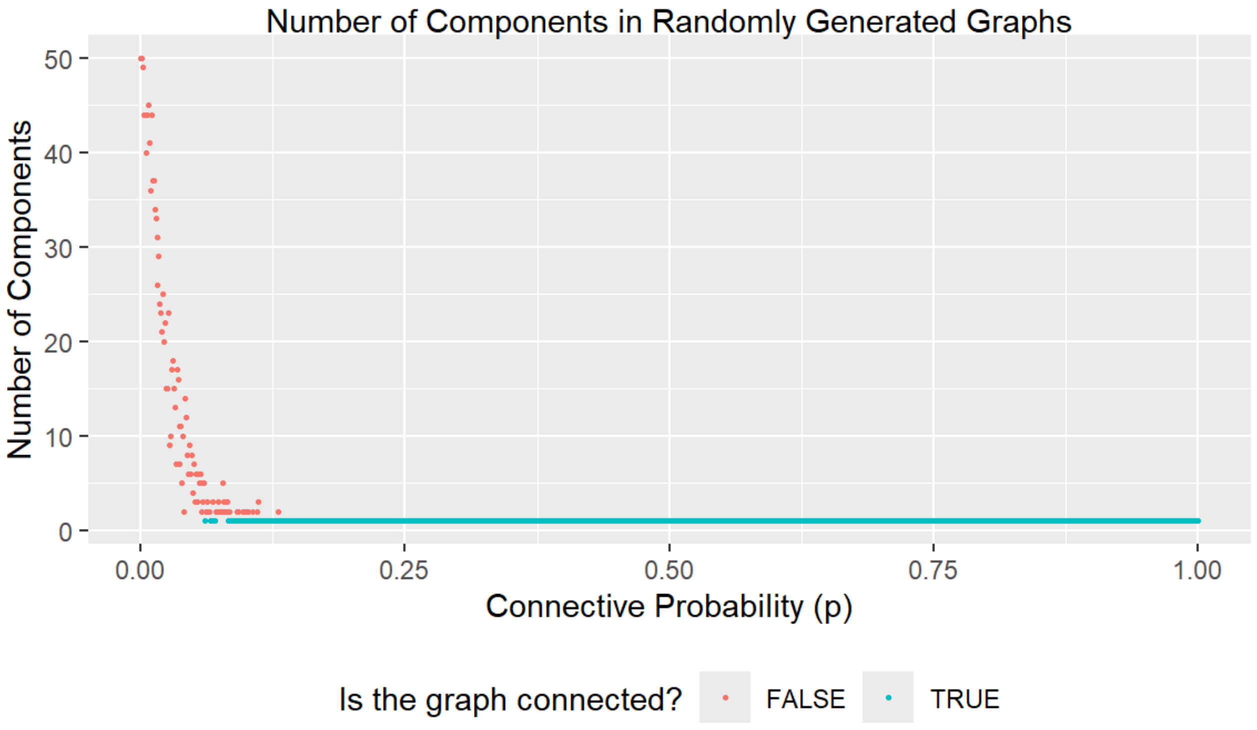
The code block below generates a dataframe containing random graphs with 50 nodes each and values of p that vary from 0 to 1. For each graph, the dataframe contains the number of components and size of its largest component.

```
# Setting network size at 50 nodes
n = 50

# Generating dataframe of graphs with varying values of p
gnp_df = tibble(
    p = seq(0, 1, 0.001), # varying values of p
    graph = map(p, ~sample_gnp(n, .x)), # generate random graph for each p
    connected = map(graph, is_connected),
    comps = map(graph, decompose),
    n_comp = map(comps, length), # get number of components
    comp_sizes = map(comps, ~sapply(.x, vcount)),
    largest_comp = map(comp_sizes, max) # get size of largest component
  ) |>
  unnest(c(connected, n_comp, largest_comp)) |>
  select(-c(comps, comp_sizes))

gnp_df |> head(10)
```

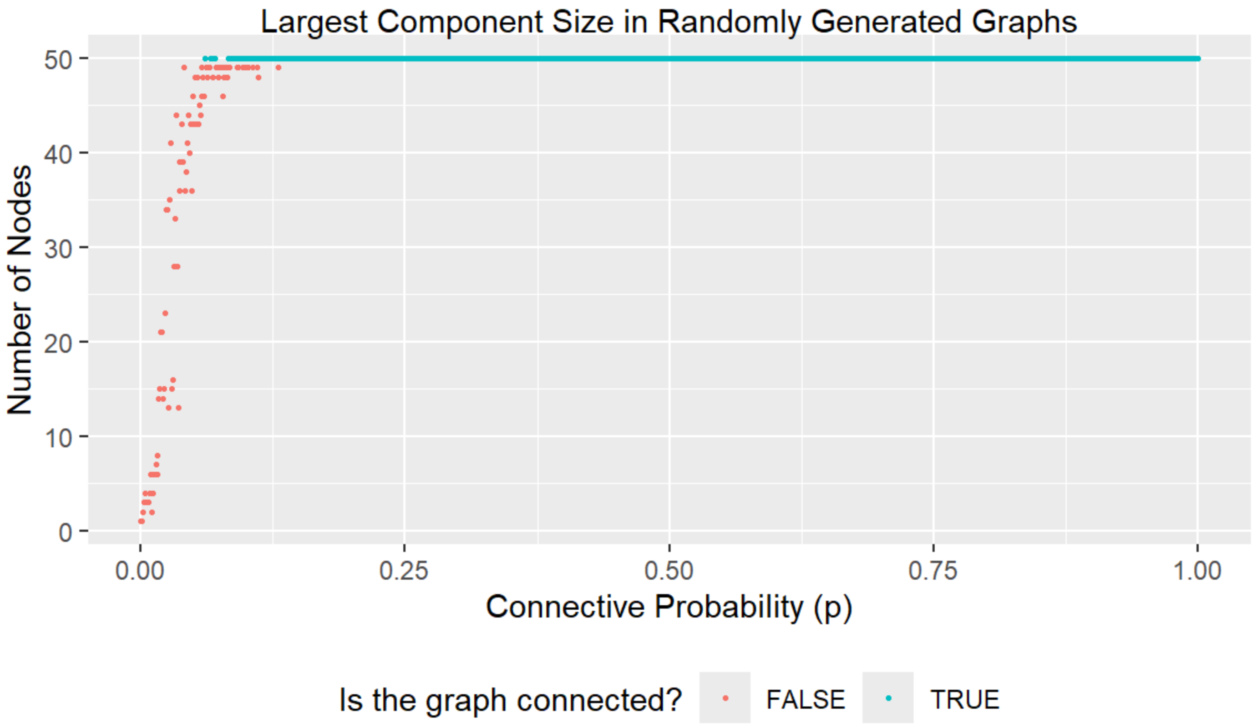
```
## # A tibble: 10 × 5
##         p graph    connected n_comp largest_comp
##     <dbl> <list>   <lgl>      <int>        <dbl>
##   1 0     <igraph> FALSE         50            1
##   2 0.001 <igraph> FALSE         50            1
##   3 0.002 <igraph> FALSE         49            2
##   4 0.003 <igraph> FALSE         44            3
##   5 0.004 <igraph> FALSE         44            4
##   6 0.005 <igraph> FALSE         40            3
##   7 0.006 <igraph> FALSE         44            3
##   8 0.007 <igraph> FALSE         45            3
##   9 0.008 <igraph> FALSE         41            4
## 10 0.009 <igraph> FALSE         36            6
```

Now let's visualize how the number of components and size of largest component change with the value of p.

```
gnp_df |>
  ggplot(aes(x = p, y = n_comp, colour = connected)) +
  geom_point(size = 0.5) +
  labs(
    title = "Number of Components in Randomly Generated Graphs",
    x = "Connective Probability (p)",
    y = "Number of Components",
    color = "Is the graph connected?"
  )
```

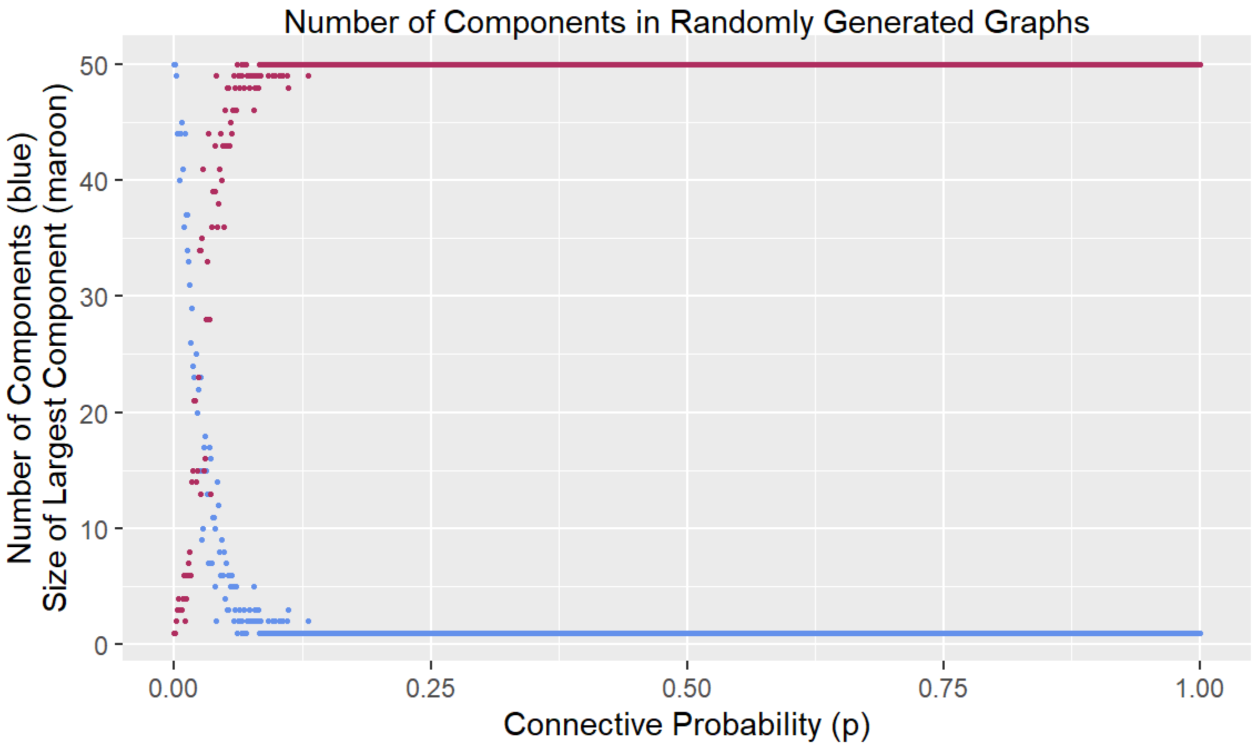


```
gnp_df |>
  ggplot(aes(x = p, y = largest_comp, colour = connected)) +
  geom_point(size = 0.5) +
  labs(
    title = "Largest Component Size in Randomly Generated Graphs",
    x = "Connective Probability (p)",
    y = "Number of Nodes",
    color = "Is the graph connected?"
  )
```



Now, just for fun, let’s plot both the number of components and size of largest component on the same axes.

```
gnp_df |>
  ggplot(aes(x = p, y = n_comp)) +
  geom_point(size = 0.5, col = "cornflowerblue") +
  geom_point(aes(x = p, y = largest_comp), data = gnp_df, size = 0.5, col = "maroon") +
  labs(
    title = "Number of Components in Randomly Generated Graphs",
    x = "Connective Probability (p)",
    y = "Number of Components (blue)\nSize of Largest Component (maroon)"
  )
```



It seems that increasing the value of p is indeed linked to an increase in the largest component's size and decrease in the number of components of a randomly generated graph.