MATH 168: HW #3

DHRUV C.

PROF. PORTER

FALL 2020

2) Problem 6.7

(a) Total number of ingoing edges at nodes $1, \dots, r$

$$= \sum_{i=1}^{r} k_i^{in}$$

total number of outgoing edges at nodes $1, \dots, r$

$$= \sum_{i=1}^{r} k_i^{out}$$

(b) Since our network is s.t. edges only run from nodes w higher labels to nodes w lower labels, the number of edges incoming from nodes $\{r+1, \dots, n\}$ to $\{1, \dots, r\}$ is the total number of ingoing edges at $1, \dots, r$ minus the total number of ingoing edges from $\{1, \dots, r\}$ to $\{1, \dots, r\}$. Note that this latter quantity is actually just the total number of outgoing edges at $1, \dots, r$ since edges must originate at nodes w/ higher labels in our network.

$\therefore$ Total num of edges running to nodes $1, \dots, r$ from $r+1, \dots, n$ $= \sum_{i=1}^{r} k_i^{in} - \sum_{i=1}^{r} k_i^{out}$

$$= \sum_{i=1}^{r} (k_i^{in} - k_i^{out})$$

(c) Since there are at most $\sum_{i=1}^{r} (k_i^{in} - k_i^{out})$ edges running from $r+1, \dots, n$ to $1, \dots, r$, we note that

$\{r+1\} \subsetneq \{r+1, \dots, n\}$ is just an element of the full originating node set, and thus we get $k_{r+1}^{out} \leq \sum_{i=1}^{r} (k_i^{in} - k_i^{out})$.

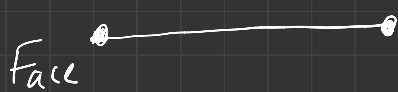
Furthermore noting that $\sum_{r+1}^{n} (k_i^{out} - k_i^{in}) = \sum_{r+1}^{n} k_i^{out} - \sum_{r+1}^{n} k_i^{in}$ represents the number of edges leaving nodes $\{r+1, \dots, n\}$ to $\{1, \dots, r\}$. As above, since $\{r\} \subsetneq \{1, \dots, r\}$ is simply an element of the receiving set, we must have

$$k_r^{in} \leq \sum_{r+1}^{n} (k_i^{out} - k_i^{in}) \quad \forall r \in [1, n] \quad \square$$

3) Problem 6.11

(a) For a single node with no edges we only have the exterior face. $\therefore$ $n=1$, $m=0$ & $f=1$.

(b) Upon adding a node with an edge we first note from the diagram below that there is still just one face, the exterior face.



So $n=2$, $m=1$ & $f=1$.

(c) If we add a self-edge or an edge between two extant nodes then to avoid losing planarity our edge must split the existing region into two faces $\Rightarrow$ num of edges increase by 1, num of faces increases by 1 while num of nodes stays the same.

(d) We will use induction to show that $n+f-m=2$ for all connected planar networks. We induct on m.

For $m=0$: we must have $n=1$ and $f=1$ for connectivity $\Rightarrow$ $n+f-m = 1+1-0=2$ as required.

<u>Inductive step</u>: Sp using strong induction that $n+f-m=2$ is true for all $m \le k$. We want to show this relation holds for $k+1$.

Let $G=(V,E)$ st $|E|=k+1$ and $G_k=(V, E-\{e\})$ such that G_k has k edges. Then,

case (i) either e is part of a cycle:

Note if e is part of a cycle in G then removing it results in a G_k that is still planar and connected. Note that G_k must have $f-1$ faces (where f is the num of faces in G) since e lies on the boundary of 2 faces as it is part of a cycle in G

(much like in part (c)). The number of nodes in G and G_k is the same. Since G_k has k edges, by the inductive hypothesis, we know our relation holds i.e.:

$$n + (f-1) - k = 2$$

$$\Rightarrow n + f - (k+1) = 2$$ i.e. our relation holds for G with $k+1$ edges.

case (ii) e connects 2 separate components (bridge):

In this case we note that the removing e does not change the num of faces. Note however that G_k then has 2 disconnected components with $m \leq k$ i.e. the relation holds for these components. Labelling the components a and b we then have $n_a + f_a - m_a = 2$ and $n_b + f_b - m_b = 2$. Moreover, $n = n_a + n_b$, $f = f_a + f_b - 1$ and

$$k = m_a + m_b.$$

Adding the relations for both these components we get:

$$(n_a + n_b) + (f_a + f_b) - (m_a + m_b) = 4$$

$$\Rightarrow n + f + 1 - k = 4 \Rightarrow n + f - k = 3$$

Subtracting 1 from both sides, we get
$n+f-(k+1)=2$ as desired,
proving the relation for G.

$\therefore$ Using the principle of strong induction we get that $n+f-m=2$

(e) For a simple graph each face must have at least 3 boundary edges. Moreover we know that $2m$ provides an upper bound for the sum of num of edges appearing in boundaries (since at most an edge is double-counted) $\Rightarrow$ $2m \geq 3f$.

Substituting $f = m+2-n$ from (d) we get

$$2m \geq 3(m+2-n)$$
$$\Rightarrow 2m \geq 3m+6-3n$$
$$\Rightarrow 3n-6 \geq m$$
$$\Rightarrow m < 3n \quad (\because 3n > 3n-6)$$
$$\Rightarrow \frac{m}{n} < 3 \Rightarrow \boxed{\frac{2m}{n} < 6}$$

where $\frac{2m}{n}$ is the mean degree c $\square$

4) Problem 10.2

(a) If the distribution is properly normalized then we have $\int_0^\infty p_k \, dk = 1$.

$\because \quad \int_0^\infty c a^k \, dk = 1 \Rightarrow c\left[\frac{a^k}{\ln(a)}\right]_0^\infty = 1$

$\Rightarrow c\left(\frac{a^\infty}{\ln(a)} - \frac{a^0}{\ln(a)}\right) = 1$

But $\because \; 0 < a < 1, \; a^\infty \longrightarrow 0$

$\Rightarrow \frac{-c}{\ln(a)} = 1 \Rightarrow c = -\ln(a)$

(b) $P = \dfrac{\int_k^\infty c a^k \, dk}{\int_0^\infty c a^k \, dk} = c\left[\frac{a^k}{\ln(a)}\right]_k^\infty$

$= c\left(\frac{a^\infty - a^k}{\ln(a)}\right)$

$= -\ln(a)\left(\frac{-a^k}{\ln(a)}\right)$ from (a) and $\because 0 < a < 1 \Downarrow a^\infty \longrightarrow 0$

$P = a^k$ where $0 < a < 1$.

(c) $W = \dfrac{\int_k^\infty k p_k \, dk}{\int_0^\infty k p_k \, dk} = \dfrac{\cancel{c}\int_k^\infty k a^k \, dk}{\cancel{c}\int_0^\infty k a^k \, dk}$

$= \left[\frac{a^k(k\log(a) - 1)}{\log^2(a)}\right]_k^\infty \Big/ \left[\frac{a^k(k\log(a) - 1)}{\log^2(a)}\right]_0^\infty$

$= \dfrac{\left(\frac{a^\infty(\infty\log(a) - 1)}{\log^2(a)} - \frac{a^k(k\log(a) - 1)}{\log^2(a)}\right)}{\left(\frac{a^\infty(\infty\log(a) - 1)}{\log^2(a)} - \frac{a^0(0\log(a) - 1)}{\log^2(a)}\right)}$

Noting that $\because 0 < a < 1, \; a^\infty \longrightarrow 0$, we have

$W = -\left(\frac{a^k(k\log(a) - 1)}{\log^2(a)}\right) \Big/ -\frac{1(-1)}{\log^2(a)}$

$$= \frac{a^k(1 - k\log(a))}{\cancel{\log^2(a)}} \cdot \cancel{\log^2(a)}$$

$$\therefore W = a^k(1 - k\log(a))$$

(d) Substituting $P = a^k$ from (b),

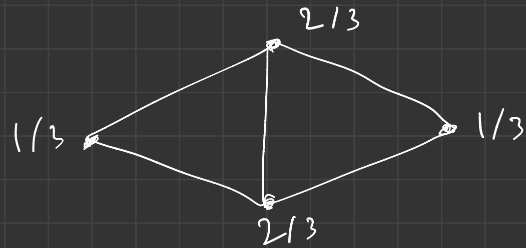
$$P - \frac{1 - 1/a}{\log(a)} P \log P = a^k - \frac{1 - 1/a}{\log(a)} a^k \log(a^k)$$

$$= a^k\left(1 - k\log(a)\left(\frac{1-1/a}{\log(a)}\right)\right)$$

$$\simeq W$$

(e) We observe that for example when $P = 0.1$ and $a = 0.01$, $W = 5.05 > 1$. These "unphysical" values occur primarily when $P > a$ since then for $P = a^k$ to hold ~~k~~ would have to negative (as $0 < a < 1$).

5) Clustering coefficients



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Then global clustering coefficient $C = \frac{\text{num } \Delta\text{s} \times 3}{\text{num connected triples}}$

$$= \frac{2 \times 3}{8} = \frac{3}{4}$$

I have written the local clustering coeffi of each node next to it above.

Averaging these to get the mean clustering coeff, we get $C_{WS} = \frac{1}{n}\sum_{i=1}^{n} C_i = \frac{1}{4}\left(1 + \frac{1+2+2}{3}\right)$

$$= \frac{6}{4\cdot 3} = \frac{1}{2}$$

6) ER graphs

(a) $$C = \frac{(\text{number of triangles}) \times 3}{(\text{number of connected triples})}$$

For an ensemble $G(n,p)$ of Erdos-Renyi graphs we first note that the expected number of triangles are $\binom{n}{3}p^3$ since p is the probability with which two randomly chosen nodes are connected and there are $\binom{n}{3}$ distinct triangles in the network.

For the num of connected triplets there are also $\binom{n}{3}$ distinct triplets but these may either all be connected with probability p^3 (i.e. triangle) or one is disconnected from one of the other nodes in the triplet i.e. $p^2(1-p)$. We adjust both. We must multiply both these by 3 to ensure counting each distinct triplet.

$\therefore$ We have expected global clustering coeff

$$C = \frac{3\cancel{\binom{n}{3}}p^3}{\cancel{\binom{n}{3}}(3p^3+3p^2(1-p))} = \frac{\cancel{3}p^3}{\cancel{3}(p^3+p^2(1-p))}$$

$$C = \frac{p^3}{p^3+p^2-p^3} = p$$ //

(b) An ER network is not a good model for social networks since it doesn't have many properties typically observed in social networks — short path lengths (small world effect), high clustering, assortative mixing & homophily i.e. the tendency of nodes to be connected w/ nodes like itself.

Since p is a constant in ER graphs, they are fully random and not a good model. Using a function of p dependent on the k_i's of the nodes it is connecting (and other factors) would lead to improvements.

7)

(a) Plots with code and observations can be found attached below part (b).

(b) The Watts-Strogatz model, although relatively simple (doesn't have many parameters), is a good ~~model~~ model for social networks. It was discovered with express purpose of a model that typically has high clustering and short path lengths to depict the small world effect. Assortive mixing by degree, an important ~~compon~~

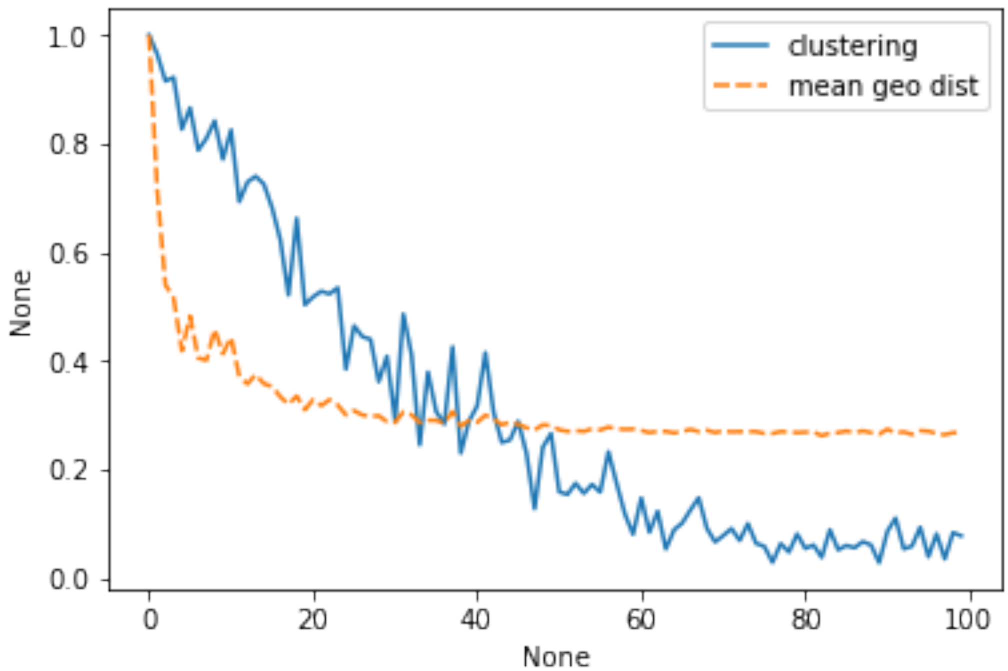
HW 3

October 28, 2020

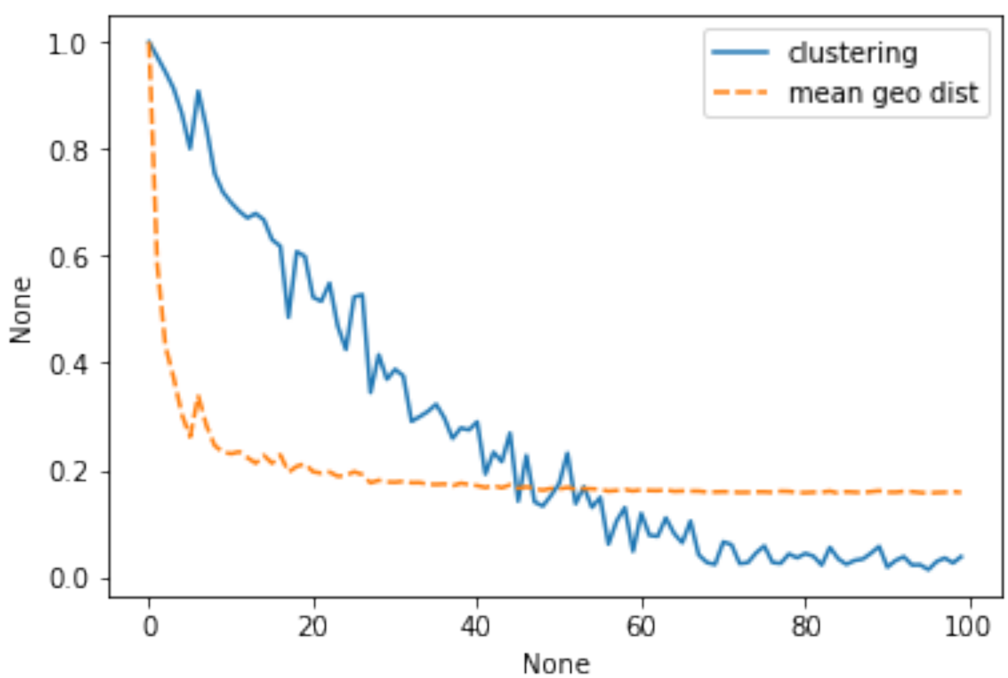
[1]:
```
import numpy as np
import pandas as pd
import networkx as nx
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

[4]:
```
def sample_WS_graphs_on_p(num_graphs, n, k=4):
    clustering, mean_geo_dis = [], []
    p_range = np.arange(0, 1, 1/num_graphs)
    for p in p_range:
        G = nx.watts_strogatz_graph(n, k, p)
        clustering.append(nx.average_clustering(G))
        mean_geo_dis.append(nx.average_shortest_path_length(G))

    clustering = np.divide(clustering, max(clustering))
    mean_geo_dis = np.divide(mean_geo_dis, max(mean_geo_dis))

    data = pd.concat([pd.Series(clustering), pd.Series(mean_geo_dis)], axis = 1)

    data.rename(columns = {0: 'clustering', 1:'mean geo dist'}, inplace = True)
    sns.lineplot(data = data)
    plt.show()
```

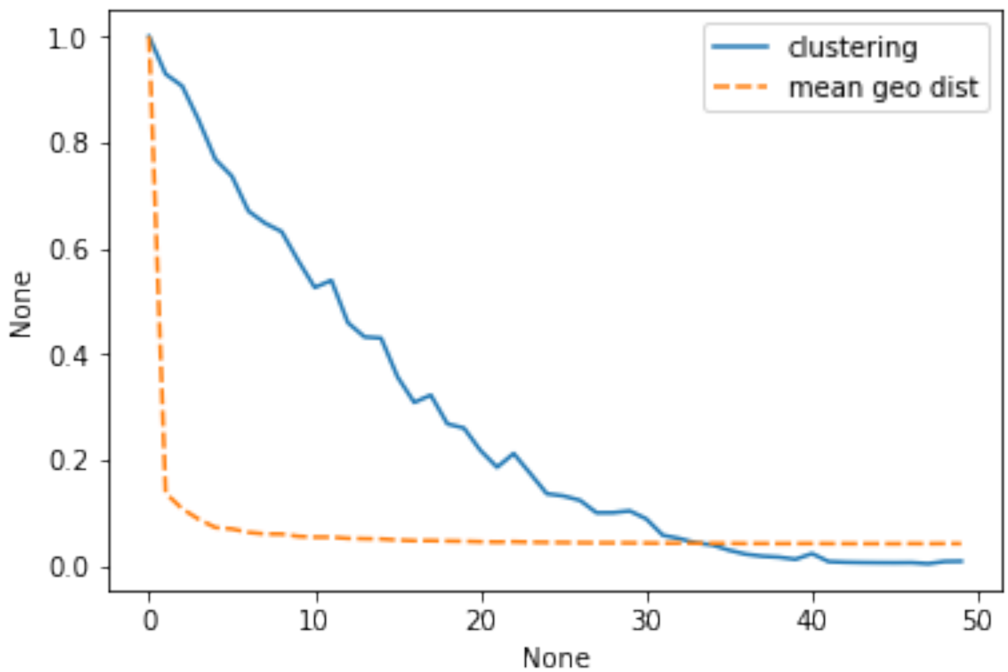
[5]:
```
sample_WS_graphs_on_p(100, 100)
```



[7]:
```
sample_WS_graphs_on_p(100, 200)
```



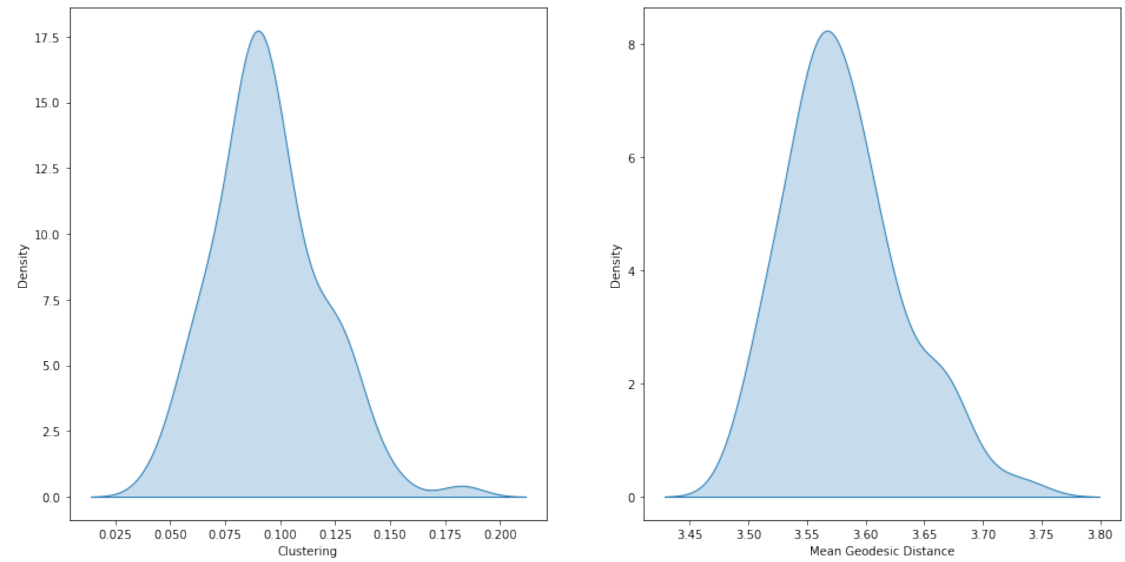
[8]:
```
sample_WS_graphs_on_p(50, 1000)
```



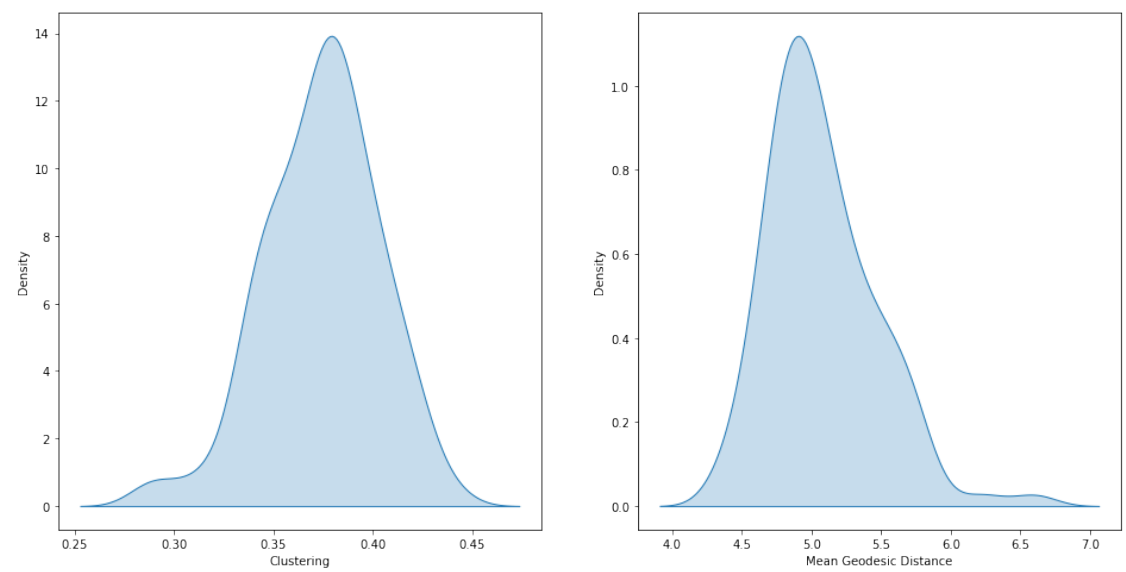
We observe that the mean geodesic distance drops immediately for higher values on p (as x increases, p increases from 0 to 1) and then abruptly levels out. The higher the value of N, the smaller the distance it levels out on. Moreover, clustering also drops more significantly with higher values of Ns but the trend of normalized clustering staying over normalized mean geodesic distance remains (except for really high values of p).

[9]:
```
def sample_WS_graphs(num_graphs, n, k=4, p=0.5):
    clustering, mean_geo_dis = [], []

    for _ in range(num_graphs):
        G = nx.watts_strogatz_graph(n, k, p)
        clustering.append(nx.average_clustering(G))
        mean_geo_dis.append(nx.average_shortest_path_length(G))

    fig, axes = plt.subplots(1,2, figsize=(16,8))
    sns.kdeplot(ax=axes[0], x=clustering, fill=True)
    axes[0].set(xlabel="Clustering")
    sns.kdeplot(ax=axes[1], x=mean_geo_dis, fill=True)
    axes[1].set(xlabel="Mean Geodesic Distance")
    plt.show()
```

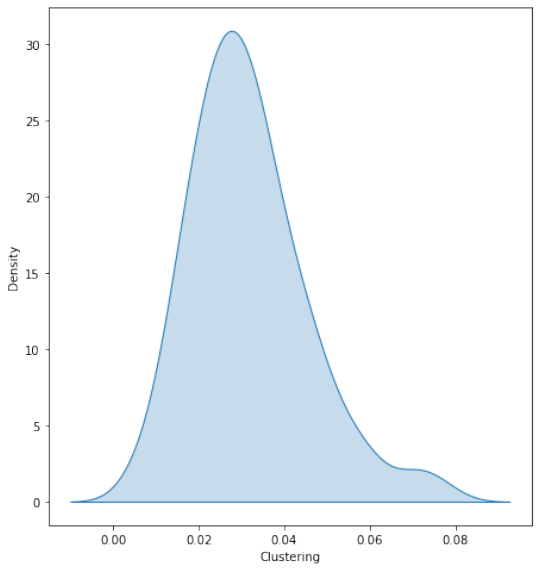
[10]:
```
sample_WS_graphs(100, 100)
```

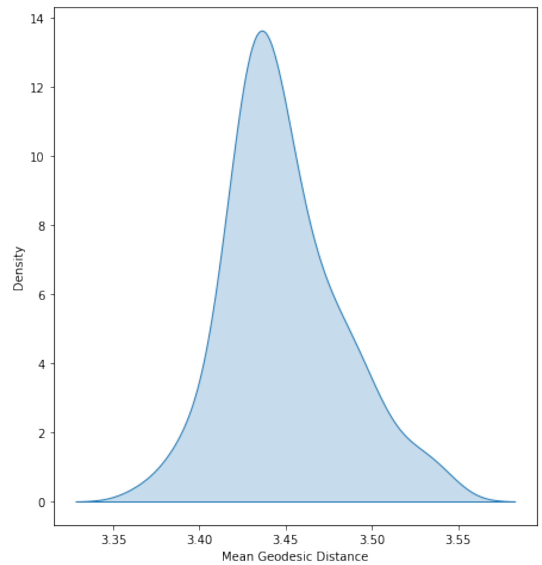


[11]:
```
sample_WS_graphs(100, 100, p=0.1)
```



[12]:
```
sample_WS_graphs(100, 100, p=0.9)
```





Above we observe sample means for clustering and mean geodesic distance for particular realizations of networks for given values of p. An interesting note is that the mean geodesic distance seems to be a slightly right-tailed distribution for every value of p we check. The curve resembles a normal distribution centered around the true value as expected.