CSCI - 5352 (Network Analysis and Modeling) - Problem Set 1

Name of Student : Rajarshi Basak

September 12, 2018

[1] (a) The adjacency matrix for network (A) is given by

A	1	2	3	4	5
1	0	1	0	0	1
2	0	0	1	0	0
3	1	0	0	0	0
4	1	0	0	0	1
5	0	0	1	1	0

[1] (b) The adjacency list for network (A) is given by

$$\begin{aligned} 1 &\rightarrow \{2,5\} \\ 2 &\rightarrow \{3\} \\ 3 &\rightarrow \{1\} \\ 4 &\rightarrow \{1,5\} \\ 5 &\rightarrow \{3,4\} \end{aligned}$$

[1] (c) For network B, let the two classes for one-mode projections be N_1 for the 5 dark vertices on the top, and N_2 for the 6 light vertices on the bottom.

Then the adjacency matrices for both one-mode projections of network (B) are given by

N_1	1	2	3	4	5
1	0	1	1	1	1
2	1	0	0	1	1
3	1	0	0	0	0
4	1	1	0	0	0
5	1	1	0	0	0

N_2	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	0	0	0	1
3	1	0	0	1	1	0
4	0	0	1	0	0	0
5	1	0	1	0	0	0
6	0	1	0	0	0	0

[2] Given $\mathbf{A} \equiv$ adjacency matrix of the graph (unwieghted, undirected edges with no self-loops), $A_{ij} \equiv$ elements of the adjacency matrix, $\mathbf{1} \equiv$ column vector, then

(a) The vector $\mathbf{k}$ whose elements are the degrees k_i of the vertices are given by

$$k_i = \sum_i A_{ij} = \sum_j A_{ji}$$

where $A_{ij} and A_{ji}$ represent the elements of the adjacency matrix $\mathbf{A}$.
In vector notation, the vector $\mathbf{k}$ is given by

$$\mathbf{k} = \mathbf{A1}$$

(b) Since each edge in an undirected network contributes twice to some degree (once for each endpoint or stub), the sum of all degrees in a network must be equal to twice the total number of edges in the network m. Hence the number m of edges in the network is given by

$$2m = \sum_i k_i = \sum_i \Big(\sum_j A_{ij} \Big)$$
$$m = \frac{1}{2} \sum_i k_i = \frac{1}{2} \sum_i \sum_j A_{ij}$$

(c) The matrix $\mathbf{N}$ whose elements N_{ij} is equal to the number of common neighbors of vertices i and j is given by

$$N_{ij} = \sum_k A_{ik} A_{kj}$$

(d) Let the total number of triangles be given by T_{net}. Since we have

$$3 * T_{net} = \sum_i \sum_j \sum_k A_{ij} A_{jk} A_{ki}$$

Thus,

$$T_{net} = \frac{1}{3} \sum_i \sum_j \sum_k A_{ij} A_{jk} A_{ki}$$

[3] Given there are n_1 vertices of type 1 and n_2 vertices of type 2, and the mean degrees of these two types of vertices are represented by c_1 and c_2. For the first type of vertices, the mean degree of a node is

$$c_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} k_i = \frac{2m_1}{n_1}$$

For the second type of vertices, the mean degree of a node is

$$c_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} k_i = \frac{2m_2}{n_2}$$

Now for a bipartite graph, there are no edges/connections within the same class of nodes. All the edges are from one-type of node to another. This implies that the total number of edges in the first one-mode projection equals the total number of edges in the second one-mode projection which in turn is equal to the number of edges in the graph. In other words, $m_1 = m_2 = m$. Hence,

$$c_1 = \frac{2m}{n_1}$$

and

$$c_2 = \frac{2m}{n_2}$$

Thus equating the values of m from the two equations above, we have

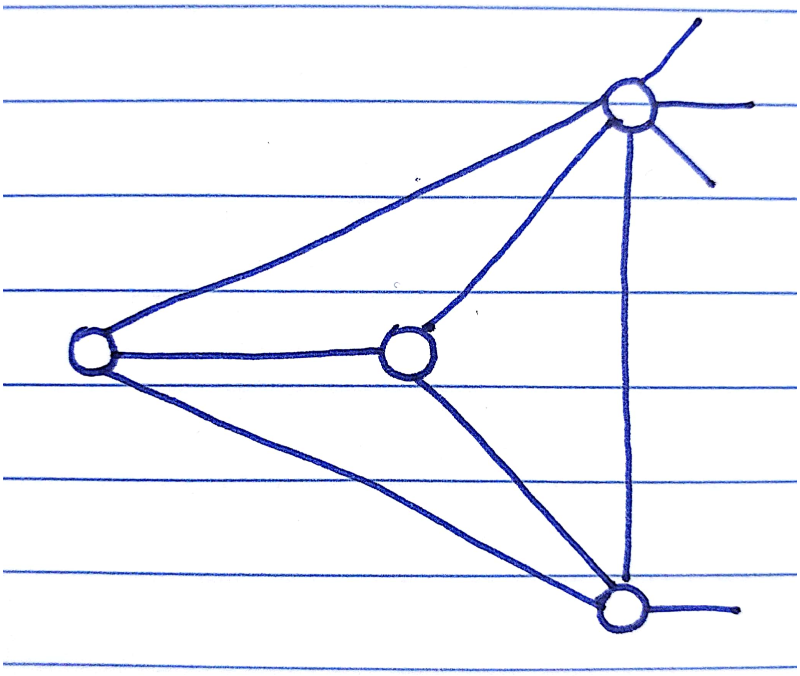
$$m = \frac{c_1 n_1}{2} = \frac{c_2 n_2}{2}$$

Rearranging which, we have

$$c_2 = \frac{n_1}{n_2} c_1$$

Thus proved.

[4] **(a)** One 3-core in network (A) has been re-drawn below:



[4] **(b)** The reciprocity of a network r is given by

$$r = \frac{1}{m} \sum_{i,j} A_{ij} A_{ji}$$

where A_{ij} is the adjacency matrix and m represents the total number of edges in the network. The total number of edges in the network is given by

$$m = \frac{1}{2} \sum_{i=1}^{n} k_i = \frac{1}{2} [5 + 4 + 4 + 3] = 8$$

Hence the reciprocity of this network is given by

$$\frac{1}{8} \sum_{i,j} A_{ij} Aji = \frac{1}{8} [1 + 1 + 1]$$

or,

$$r = \frac{3}{8} = 0.375$$

[4] **(c)** The cosine similarity of the vertices A and B is given by

$$\sigma_{AB} = \frac{\sum_k A_{Ak} A_{kB}}{\sqrt{k_A k_B}} = \frac{n_{AB}}{\sqrt{k_A k_B}}$$

where k_A and k_B are the degrees of the vertices A and B respectively.

Since the vertices A and B have two common vertices, $n_{AB} = 2$, $k_A = 4$ and $k_B = 5$. Hence the required cosine similarity is

$$\sigma_{AB} = \frac{2}{\sqrt{4 * 5}} = \frac{2}{\sqrt{10}} = \frac{2}{\sqrt{2}\sqrt{5}}$$

or

$$\sigma_{AB} = \sqrt{\frac{2}{5}}$$

[5] (a) For the given Cayley tree (with $k = 3$), the number of vertices accesible from the central vertex at the first step is 3, at the second step is $3 * 2 = 6$, at the third step is $3 * 2 * 2 = 12$ and so on. In terms of the step d and degree $k = 3$, the number of vertices accessible from the central vertex at the first step is k, at the second step is $k(k-1)^1$, at the third step is $k(k-1)(k-1) = k(k-1)^2$ and so on.

Generalizing the above observations, we conclude that the number of vertices accessible from the central vertex at the dth step is given by $k(k-1)(k-1)(k-1)...$ with $(k-1)$ multiplied by itself $d-1$ times. This is because at the first step k vertices are accessible, and at each of the following steps the number of vertices accessible grows by a factor of $(k-1)$. Hence the total number of such vertices is $k(k-1)^{d-1}$.

This has been summarized in the table below:

Step(d)	Number of vertices accessible	Number of vertices accessible in terms of k and d
1	3	k
2	$3 * 2 = 6$	$k(k-1)$
3	$3 * 2 * 2 = 12$	$k(k-1)^2$
4	$3 * 2 * 2 * 2 = 24$	$k(k-1)^3$
d	-	$k(k-1)^{d-1}$

[5] (b) Given $k = 3$, the diameter of the network can be deduced as follows: For a level 1 Cayley tree, the diameter is $2 * 1 = 2$ units, for a level 2 Cayley tree, the diameter is $2 * 2 = 4$ units, for a level 3 Cayley tree, the diameter is $3 * 2 = 6$ units, and so on. Thus for the dth level, the diameter of the Cayley tree is given by $d(k-1)$. This is summarized in the table below:

Level	Diameter
1	$1 * 2 = 2$
2	$2 * 2 = 4$
3	$3 * 2 = 6$
d	$d(k-1)$

So far we have the diameter in terms of d and k, and we want it in terms of k and n. To find this, we first find the total number of vertices, given that we know the number at each level from the previous part of this problem.

The total number of vertices is given by:

$$1 + k + k(k-1) + k(k-1)^2 + k(k-1)^3 + ... + k(k-1)^{d-1}$$

$$= 1 + k\big[(k-1) + (k-1)^2 + (k-1)^3 + ... + (k-1)^{d-1}\big]$$

$$= 1 + k\sum_{i=0}^{d-1}(k-1)^i$$

which is a geometric series in $(k-1)$.

Now the sum of the first n terms geometric series, given a is the first term and r is the common ratio, is

$$s = a\left(\frac{1-r^n}{1-r}\right) \quad (\text{if } r \neq 1)$$

In our case, $a = 1$, $r = (k-1)$ (here we assume $k > 2$ to allow $r \neq 1$) and $n = (d-1)$, and the sum of our geometric series becomes

$$\left[\frac{1-(k-1)^d}{1-(k-1)}\right]$$

Hence the total number of vertices becomes:

$$1 + k\left[\frac{1-(k-1)^d}{1-(k-1)}\right]$$

But the total number of vertices in the Cayley tree in n. Hence equating the above eqaution with n, we have

$$1 + k\left[\frac{1-(k-1)^d}{1-(k-1)}\right] = n$$

$$=> \frac{1-(k-1)^d}{1-(k-1)} = \frac{n-1}{k}$$

$$=> 1-(k-1)^d = (\frac{2}{k}-1)(n-1) = \frac{2n}{k} - n - 2$$

$$=> n = \frac{2n}{k} - 2 = (k-1)^d$$

$$=> d = log_{k-1}\left[n - \frac{2n}{k} - 2\right]$$

Replacing the expression for d we found above into the expression for the diameter we found before, we have for the diameter D of a Cayley tree:

$$D = log_{k-1}\left[n - \frac{2n}{k} - 2\right](k-1)$$

[5] (c) Since the Diameter D of the Cayley tree is proportional to $log_{k-1}\left[n - \frac{2n}{k} - 2\right]$, the diameter essentially increases as $log(n)$ or slower. In other words, the Diameter $D \propto O(logn)$, and hence the network displays the "small-world effect".

[6] (a) Given the average degree of the network $\langle k \rangle = \frac{1}{n}\sum_{i=1}^{n} k_i$, the average squared-degree of a network $\langle k^2 \rangle = \frac{1}{n}\sum_{i=1}^{n} k_i^2$, and the mean neighbour degree (MND) of a network

$$\langle k_v \rangle = \frac{1}{2m}\sum_{u=1}^{n}\sum_{v=1}^{n} k_v A_{uv}$$

we have, shuffling the order of the summations

$$=> \langle k_v \rangle = \frac{1}{2m}\sum_{v=1}^{n}\sum_{u=1}^{n} k_v A_{uv}$$

$$=> \langle k_v \rangle = \frac{1}{2m}\sum_{v=1}^{n} k_v \sum_{u=1}^{n} A_{uv}$$

but since $\sum_{u=1}^{n} A_{uv} = k_v$, we have

$$=> \langle k_v \rangle = \frac{1}{2m}\sum_{v=1}^{n} k_v * k_v = \frac{1}{2m}\sum_{v=1}^{n}(k_v)^2 = \frac{1}{2m}\sum_{u=1}^{n}(k_u)^2$$

Now $\frac{1}{n}\sum_{i=1}^{n}(k_i)^2 = \langle k^2 \rangle$, and hence

$$=> \langle k_v \rangle = \frac{1}{2m} n \langle k^2 \rangle$$

rearranging which, we have

$$=> \langle k_v \rangle = \frac{\langle k^2 \rangle}{\frac{2m}{n}}$$

but $\langle k \rangle = \frac{1}{n}\sum_{i=1}^{n} k_i = \frac{2m}{n}$, so

$$=> \langle k_v \rangle = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

which is the expression for $\langle k_v \rangle$ in terms of the average squared-degree $\langle k^2 \rangle$ and the average degree $\langle k \rangle$

[6] (b)

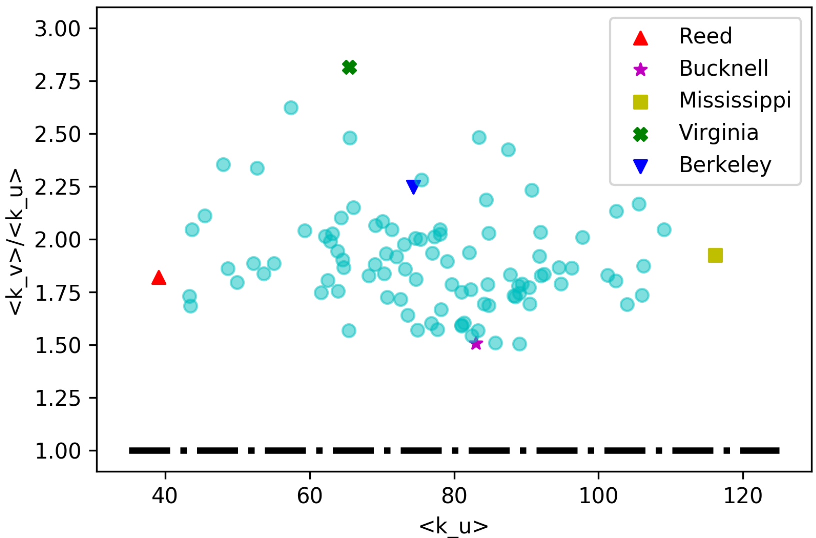


Figure 1: Scatter-Plot of $\frac{\langle k_v \rangle}{\langle k_u \rangle}$ vs. $\langle k_u \rangle$

As shown in the scatter-plot above, we do observe a 'friendship paradox' across these networks as a group. The degree to which we observe a 'friendship paradox' is bounded by Colgate, which has the smallest value of $\langle k_v \rangle \div \langle k_u \rangle$ at 1.51 and Virigina, which has the highest value at 2.82.

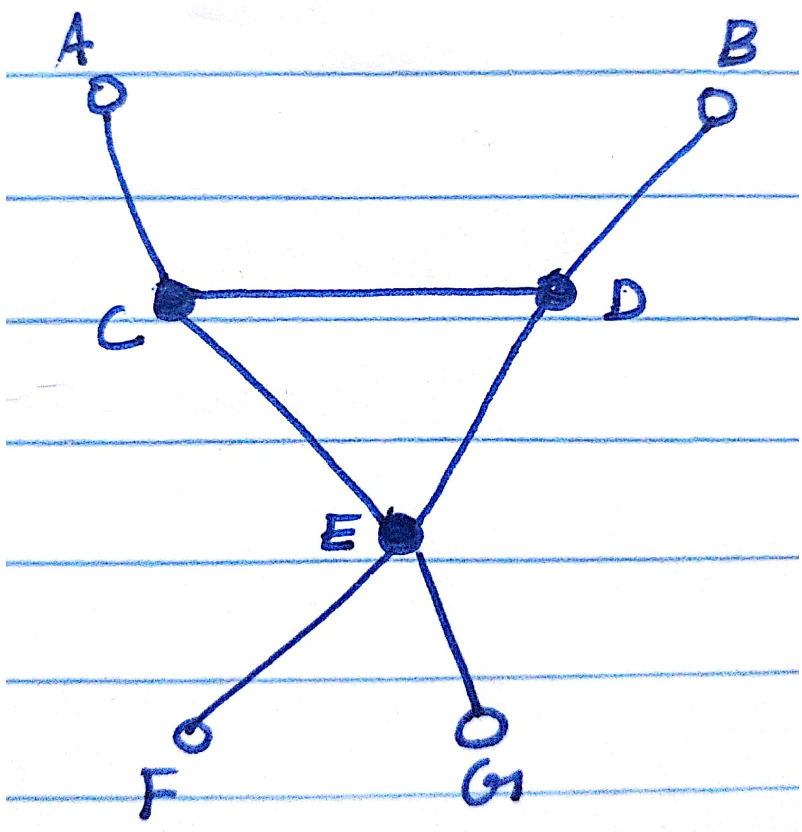
Why should we expect to see this paradox? When the ratio is more than 1, it means that $\langle k_v \rangle \div \langle k_u \rangle$ is more than 1, which in turn means that $\langle k^2 \rangle \div \langle k \rangle^2$ is more than 1. This implies that $\langle k^2 \rangle$ is greater than $\langle k \rangle^2$.

Since $\langle k^2 \rangle - \langle k \rangle^2$ is the variance of the network with respect to the vetices, this means that the network has a non-zero and non-negative variance - i.e. some vertices have more connections than others in the network. In other words, the ratio will be 1 when all the vertices have the same number of edges. Since it is extremely rare that all the vertices in a network will have equal number of connections, we can expect to see a value of the ratio different from 1 for all such networks which do not have identical number of edges/connections for all its vertices.

[6] (c) To illustrate the 'majority illusion', we consider two example networks and examine how the quantities of interest change as we alter the properties of the network.

Let our binary variable be $x_u \in$ (vertex is light, vertex is dark) i.e. a vertex can be either lightly or darkly colored. When a vertex is lightly colored, the binary variable $x_u = 0$, and when it is darkly colored, the variable is $x_u = 1$.

In the first network, we have 7 vertices, and we have vertices C,D and E darkly colored, while vertices A, B, F and G are darkly colored (shown below).



Here the fraction of vertices that are dark (i.e. which exhibit the property $x_u = 1$) is

$$q = \frac{1}{n}\sum_u x_u = \frac{1}{7}3 = \frac{3}{7}$$

and hence $q < 0.5$

Now if we consider the neighbors of each vertex then the value of x_v for each of the vertices are as summarized in the table below:

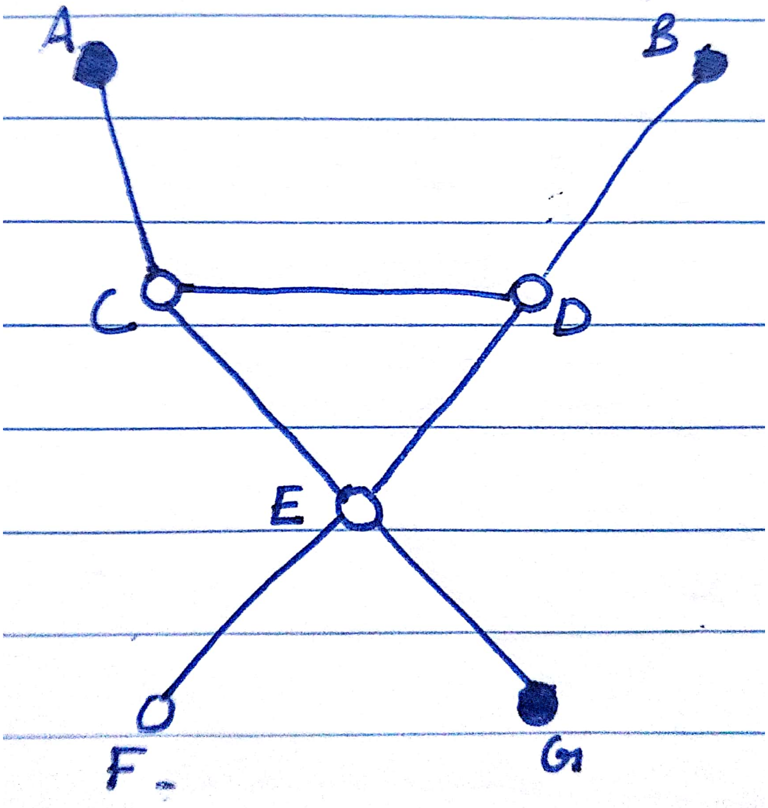
Node	A	B	C	D	E	F	G
Value of x_v	1	1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	1	1

Hence the average value of the binary variable for the node's neighbors is given by

$$\langle x_v \rangle = \frac{1}{7}\left[1 + 1 + \frac{2}{3} + \frac{2}{3} + \frac{1}{2} + 1 + 1\right] = \frac{5}{6} = 0.83$$

and hence $\langle x_v \rangle > 0.5$.

The second network has the same number of vertices, but here it is A, B and G that are darkly colored, while C, D, E and F are lightly colored (shown below).



Again the the fraction of vertices that are dark here (i.e. which exhibit the property $x_u = 1$) is

$$q = \frac{1}{n}\sum_u x_u = \frac{1}{7}3 = \frac{3}{7}$$

and hence $q < 0.5$, which was the case with the previous network.

If we consider the neighbors of each vertex then the value of x_v for each of the vertices are now different, and they have been summarized in the table below:

Node	A	B	C	D	E	F	G
Value of x_v	0	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{4}$	0	0

Hence the average value of the binary variable for the node's neighbors is given by

$$\langle x_v \rangle = \frac{1}{7}\Big[0 + 0 + \frac{1}{3} + \frac{1}{3} + \frac{1}{4} + 0 + 0\Big] = \frac{11}{84} = 0.13$$

and hence $\langle x_v \rangle < 0.5$.

Thus we observed that the majority of the node's neighbors exhibit the binary property in the first example, but not in the example, even though in both the examples, a minority of the actual nodes exhibit the property.

We note that in the first example the binary property is exhibited by nodes that have more degrees, while in the second example, it is the nodes with the lesser number of degrees that have the property. In other words, the more popular nodes (having comparatively more connections) have the property in the first network, while the less popular nodes (having comparatively lesser connections) have the property in the second example.

Therefore, we conclude that there is a strong positive correlation between the degree of the node in a network exhibiting the binary property, and the network showing a 'majority illusion'. The more the number of nodes with a higher degree that show a binary property in a network, the more the network will show a 'majority illusion' for that property.

The code for generating the scatter-plot from the data-set is shown below.

```
import matplotlib.pyplot as plt
import numpy as np
import networkx as nx
import glob

def getgraphparameters(filename):
    mean_degree = 0
    total_edges = 0
    total_vertices = 0
    total_squared_degree = 0
    mean_squared_degree = 0
    mean_neighbour_degree = 0
    ratio = 0

    graph = nx.read_edgelist(filename, nodetype=int, data=(('weight',float),))

    total_edges = len(graph.edges())
    total_vertices = len(graph.nodes())
    mean_degree = 2*(total_edges)/total_vertices

    for i in graph.nodes():
        total_squared_degree += len(graph.edges(i))**2

    mean_squared_degree = total_squared_degree/total_vertices
    mean_neighbour_degree = mean_squared_degree/mean_degree
    ratio = mean_neighbour_degree/mean_degree
    #print(ratio)
    return (mean_degree, ratio, mean_neighbour_degree)

parameters = {}
for filename in glob.glob('*.txt'):
    print(filename)
    ra, md, mnd = getgraphparameters(filename)
    print (ra, md, mnd)
    parameters[filename] = (ra, md, mnd)

print(parameters)

x = np.zeros((95))
y = np.zeros((95))
ind = 0
for j in parameters:
    temp=(parameters[j])

    if j=='Reed98.txt':
        xreed98 = temp[0]
        yreed98 = temp[1]
    elif j=='Bucknell39.txt':
        xbuck = temp[0]
        ybuck = temp[1]
    elif j=='Mississippi66.txt':
        xmiss = temp[0]
        ymiss = temp[1]
    elif j=='Virginia63.txt':
        xvirg = temp[0]
        yvirg = temp[1]
    elif j=='Berkeley13.txt':
        xberk = temp[0]
        yberk = temp[1]
    else:
        x[ind] = temp[0]
```

```
        y[ind] = temp[1]
        ind = ind+1

plt.scatter(xreed98,yreed98, alpha = 1.0, color = 'r',marker = '^')
plt.scatter(xbuck,ybuck, alpha = 1.0, color = 'm', marker = '*')
plt.scatter(xmiss,ymiss, alpha = 1.0, color = 'y', marker = 's')
plt.scatter(xvirg,yvirg, alpha = 1.0, color = 'g', marker = 'X')
plt.scatter(xberk,yberk, alpha = 1.0, color = 'b', marker = 'v')
plt.scatter(x,y, alpha=0.5, color = 'c')

plt.legend(( 'Reed', 'Bucknell', 'Mississippi', 'Virginia',
            'Berkeley'))

xx = np.linspace(35, 125, num=100)
yy = np.ones(np.size(xx))
plt.plot(xx,yy,linestyle='-.', color='k', linewidth=3)
plt.xlabel('<k_u>')
plt.ylabel('<k_v>/<k_u>')
plt.ylim((0.9,3.1))
plt.savefig('k_v_k_u-vs-k_u-plot.png', dpi = 300)
plt.show()
```

Note: I collaborated with Ganesh Chandra Satish on this assignment. I would also like to thank Subhayan De for helping me with the scatter-plot.