

```

1  #!/usr/bin/python
2  ...
3  BLG 517E - Modelling and Performance Analysis of Networks
4  Random Graphs Assignment
5  Tugrul Yatagan 504161551
6  ...
7  import igraph as ig
8  import numpy as np
9  import scipy as sp
10 import scipy.stats
11 import matplotlib.pyplot as plt
12
13 N=50
14 repeat=5
15 P list=[0.05, 0.2, 0.4, 0.6, 0.8]
16 confidence=0.95
17
18 edge list=[[] for x in range(repeat)] for y in range(len(P list))
19 giant size list=[[] for x in range(repeat)] for y in range(len(P list))
20 diameter list=[[] for x in range(repeat)] for y in range(len(P list))
21 avg degree list=[[] for x in range(repeat)] for y in range(len(P list))
22
23 def confidence interval(data, conf):
24     a = 1.0*np.array(data)
25     n = len(a)
26     m, se = np.mean(a), sp.stats.sem(a)
27     h = se * sp.stats.t.ppf((1+conf)/2., n-1)
28     #return m, m-h, m+h, h
29     return m, h
30
31
32 for k, P in enumerate(P list):
33     for i in range(repeat):
34         g = ig.Graph.Erdos_Renyi(n=N, p=P)
35         ig.plot(g, 'graph p{0} {1}.png'.format(P, i+1))
36         print "Random Graph #{0}, n={1}, p={2}".format(i+1, N, P)
37         edge list[k][i] = g.ecount()
38         print "    Number of edge = {0}".format(edge list[k][i])
39         giant size list[k][i] = max(g.components().sizes())
40         print "    Giant component size = {0}".format(giant size list[k][i])
41         diameter list[k][i] = g.diameter()
42         print "    Diameter = {0}".format(diameter list[k][i])
43         avg degree list[k][i] = np.mean(g.degree())
44         print "    Average degree = {0}".format(avg degree list[k][i])
45         print g.degree distribution()
46     print "****"
47
48 print edge list
49 print giant size list
50 print diameter list
51 print avg degree list
52
53
54 avg degree mean list=[confidence interval(x, confidence)[0] for x in avg degree list]
55 avg degree confidence list=[confidence interval(x, confidence)[1] * 2 for x in
56 avg degree list]
57 plt.bar(range(len(P list)), avg degree mean list, yerr=avg degree confidence list,
58 alpha=0.2, align='center', width=0.4)
59 plt.xticks(range(len(P list)), [str(p) for p in P list])
60 for m, c, k in zip(avg degree mean list, avg degree confidence list,
61 range(len(P list))):
62     plt.annotate("{0:.1f} +- {1:.1f}".format(m, c/2), (k, m+2), va='bottom',
63 ha='center', fontsize=10)
64 plt.grid()
65 plt.title("Average degree")
66 plt.ylabel("Average degree")
67 plt.xlabel("p")
68 plt.savefig('avg degree.png')
69 plt.qcf().clear()
70
71 edge mean list=[confidence interval(x, confidence)[0] for x in edge list]
72 edge confidence list=[confidence interval(x, confidence)[1] * 2 for x in edge list]
73 plt.bar(range(len(P list)), edge mean list, yerr=edge confidence list, alpha=0.2,
74 align='center', width=0.4)

```

```

70 plt.xticks(range(len(P list)), [str(p) for p in P list])
71 for m, c, k in zip(edge mean list, edge confidence list, range(len(P list))):
72     plt.annotate("{0:.1f} +- {1:.1f}".format(m, c/2), (k, m+10), va='bottom',
73                 ha='center', fontsize=10)
74 plt.grid()
75 plt.title("Number of edge")
76 plt.ylabel("Number of edge")
77 plt.xlabel("p")
78 plt.savefig('edge.png')
79 plt.gcf().clear()
80 giant size mean list=[confidence interval(x, confidence)[0] for x in giant size list]
81 giant size confidence list=[confidence interval(x, confidence)[1] * 2 for x in
82     giant size list]
83 plt.bar(range(len(P list)), giant size mean list, yerr=giant size confidence list,
84         alpha=0.2, align='center', width=0.4)
85 plt.xticks(range(len(P list)), [str(p) for p in P list])
86 for m, c, k in zip(giant size mean list, giant size confidence list,
87     range(len(P list))):
88     plt.annotate("{0:.1f} +- {1:.1f}".format(m, c/2), (k, m+0), va='bottom',
89                 ha='center', fontsize=10)
90 plt.grid()
91 plt.title("Giant component size")
92 plt.ylabel("Giant component size")
93 plt.xlabel("p")
94 plt.savefig('giant.png')
95 plt.gcf().clear()
96 diameter mean list=[confidence interval(x, confidence)[0] for x in diameter list]
97 diameter confidence list=[confidence interval(x, confidence)[1] * 2 for x in
98     diameter list]
99 plt.bar(range(len(P list)), diameter mean list, yerr=diameter confidence list,
100         alpha=0.2, align='center', width=0.4)
101 plt.xticks(range(len(P list)), [str(p) for p in P list])
102 for m, c, k in zip(diameter mean list, diameter confidence list, range(len(P list))):
103     plt.annotate("{0:.1f} +- {1:.1f}".format(m, c/2), (k, m+0.2), va='bottom',
104                 ha='center', fontsize=10)
105 plt.grid()
106 plt.title("Diameter")
107 plt.ylabel("Diameter")
108 plt.xlabel("p")
109 plt.savefig('diameter.png')
110 plt.gcf().clear()

```