

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def printPrettyMatrix(objStr, matrix):
5     objStr+="\t\t["
6     for row in matrix:
7         objStr+="["
8         for col in row:
9             objStr+=f" {col:.02f},"
10        objStr= objStr[:-1]+"],"
11        objStr+="\n \t\t"
12    objStr=objStr[:-5]+"]"
13    return objStr
14
15 def showPlot(PlotSS, PlotII, PlotRR, node, ts, main_title):
16     plt.title(f"{main_title}\nNode {node} Infctd:{np.max(PlotII)}")
17     S_i = plt.plot(PlotSS[node,:], label=f'S_{node}({ts})') # just plot node 0's
18     I_i = plt.plot(PlotII[node,:], label=f'I_{node}({ts})') # just plot node 0's
19     R_i = plt.plot(PlotRR[node,:], label=f'R_{node}({ts})') # just plot node 0's
20     plt.legend()
21     plt.show()
22
23 def SIR(beta, gamma, A, I0, timesteps, show_plot=False, node=1, main_title="Question
24 00") :
25     ...
26     Notes:
27     - assumes that everything is dimensioned correctly
28     - all S/I/R are fractions of node's population
29     - R0 = 0
30     - S0 = 1-I0
31     ...
32     n = A.shape[0]
33
34     # initialize:
35     SS = np.zeros([n,timesteps]) # each column in a list of S_i's; there is one
36     column per timestep
37     II = np.zeros_like(SS) # copy the shape of SS to create II
38     RR = np.zeros_like(SS) # nobody starts out with immunity
39     II[:,0] = I0 # set the initial infection
40     SS[:,0] = np.ones(n)-II[:,0] # if you're not infected, you're susceptible
41
42     for t in range(1,timesteps):
43         bigS = np.diag(SS[:,t-1])
44         new_infections = beta * bigS @ A @ II[:,t-1]
45         heals = gamma*II[:,t-1]
46
47         SS[:,t] = SS[:,t-1] - new_infections
48         II[:,t] = II[:,t-1] + new_infections - heals
49         RR[:,t] = RR[:,t-1] + heals
50
51     if show_plot:
52         showPlot(SS, II, RR, node, timesteps, main_title)
53
54     return SS,II,RR
55
56 #####

```

```

55 # Homework 05: Question 1
56 #####
57 print("\nQuestion 1:")
58 beta = 0.8
59 gamma = 0.4
60 Q1_og_infection = [0.01, 0.0, 0.0, 0.0, 0.0]
61 Q1_graph_adj = np.array([[0.8, 0.0, 0.0, 0.05, 0.15],
62                           [1.0, 0.0, 0.0, 0.0, 0.0],
63                           [0.3, 0.2, 0.5, 0.0, 0.0],
64                           [0.0, 0.0, 0.05, 0.95, 0.0],
65                           [0.0, 0.0, 0.0, 0.2, 0.8]])
66 q1_timesteps = 2
67
68 Q1_SS, Q1_II, Q1_RR = SIR(beta, gamma, Q1_graph_adj, Q1_og_infection, q1_timesteps,
69 show_plot=True, main_title="Question 01")
70 print("\tS_i(t):\t")
71 print(printPrettyMatrix("", Q1_SS))
72 print("\tI_i(t):\t")
73 print(printPrettyMatrix("", Q1_II))
74 print("\tR_i(t):\t")
75 print(printPrettyMatrix("", Q1_RR))
76 #####
77 # Homework 05: Question 2
78 #####
79 print("\nQuestion 2:")
80 Q2_og_infection = [0.01, 0.0, 0.0, 0.0, 0.0]
81 Q2_graph_adj = np.array([[0.8, 0.0, 0.0, 0.05, 0.15],
82                           [1.0, 0.0, 0.0, 0.0, 0.0],
83                           [0.3, 0.2, 0.5, 0.0, 0.0],
84                           [0.0, 0.0, 0.05, 0.95, 0.0],
85                           [0.0, 0.0, 0.0, 0.2, 0.8]])
86 q2_timesteps = 0
87
88 while True:
89     Q2_SS, Q2_II, Q2_RR = SIR(beta, gamma, Q2_graph_adj, Q2_og_infection,
90 q2_timesteps+1, show_plot=False)
91     if np.all(Q2_II[:, q2_timesteps] != 0.0):
92         print(f"all nodes are infected at time step {q2_timesteps} \t{Q2_II[:,
93 q2_timesteps]}")
94         break
95     if q2_timesteps % 2 == 0:
96         showPlot(Q2_SS, Q2_II, Q2_RR, 2, q2_timesteps, main_title="Question 02")
97
98     q2_timesteps += 1
99
100 print("\tS_i(t):\t")
101 print(printPrettyMatrix("", Q2_SS))
102 print("\tI_i(t):\t")
103 print(printPrettyMatrix("", Q2_II))
104 print("\tR_i(t):\t")
105 print(printPrettyMatrix("", Q2_RR))
106
107 #####
108 # Homework 05: Question 3
109 #####
110 print("\nQuestion 3:")
111 print("\tSuppose that the connection between node 2 and node 1 is removed. If\n",

```

```

112         "\tnothing else changes in the network, would this change the spread of\n",
113         "\tthe epidemic? Can you predict how many people in node 2 would eventually\n",
114         "\tget sick?\n")
115 Q3_og_infection = [0.01, 0.0, 0.0, 0.0, 0.0]
116 Q3_graph_adj = np.array([[0.8, 0.0, 0.0, 0.05, 0.15],
117                          [0.0, 0.0, 0.0, 0.0, 0.0],
118                          [0.3, 0.2, 0.5, 0.0, 0.0],
119                          [0.0, 0.0, 0.05, 0.95, 0.0],
120                          [0.0, 0.0, 0.0, 0.2, 0.8]])
121 q3_timesteps = 0
122
123 while True:
124     Q3_SS, Q3_II, Q3_RR = SIR(beta, gamma, Q3_graph_adj, Q3_og_infection,
125                               q3_timesteps+1, node=2, show_plot=False)
126     if (Q3_II[:, q3_timesteps][1] != 0.0):
127         print(f"Node 2 is infected at time step {q3_timesteps} \t{Q3_II[:,
128 q3_timesteps]}")
129         break
130     if q3_timesteps % 5 == 0:
131         print(f"Step:\t{q3_timesteps}")
132         print(f"Node 2 infection %:{Q3_II[:, q3_timesteps][1]}")
133         print(f"The rest of the network {Q3_II[:, q3_timesteps]}")
134         showPlot(Q3_SS, Q3_II, Q3_RR, 2, q3_timesteps, "Question 03")
135     if q3_timesteps > 50:
136         break
137     q3_timesteps += 1
138
139 print(f"Last ten time steps of {q3_timesteps-1}")
140 print("\tS_i(t):\t")
141 print(printPrettyMatrix("", Q3_SS[:, q3_timesteps-10:q3_timesteps]))
142 print("\tI_i(t):\t")
143 print(printPrettyMatrix("", Q3_II[:, q3_timesteps-10:q3_timesteps]))
144 print("\tR_i(t):\t")
145 print(printPrettyMatrix("", Q3_RR[:, q3_timesteps-10:q3_timesteps]))
146
147 #####
148 # Homework 05: Question 4
149 #####
150 print("\nQuestion 4:")
151 print("\tRepeat that same thought experiment, but this time let the initial\n",
152       "\tinfection start on node 2 (so S_i(0)=0.1 and S_2(0)=0.99, but all other\n",
153       "\tnodes have S_i(0)=1, I_i(0)=0 and R_i(0)=0. Now can you predict how
154       many\n",
155       "\tpeople in node 2 would eventually get sick?\n")
156 Q4_og_infection = [0.0, 0.01, 0.0, 0.0, 0.0]
157 Q4_graph_adj = np.array([[0.8, 0.0, 0.0, 0.05, 0.15],
158                          [0.0, 0.0, 0.0, 0.0, 0.0],
159                          [0.3, 0.2, 0.5, 0.0, 0.0],
160                          [0.0, 0.0, 0.05, 0.95, 0.0],
161                          [0.0, 0.0, 0.0, 0.2, 0.8]])
162 q4_timesteps = 0
163
164 while True:
165     Q4_SS, Q4_II, Q4_RR = SIR(beta, gamma, Q4_graph_adj, Q4_og_infection,
166                               q4_timesteps+1, node = 2, show_plot=False, main_title="Question 04")
167     if q4_timesteps % 10 == 0:
168         print(f"Step:\t{q4_timesteps}")
169         print(f"Node 2 infection %:{Q4_II[:, q4_timesteps][1]}")
170         print(f"The rest of the network {Q4_II[:, q4_timesteps]}")

```

```

168     showPlot(Q4_SS, Q4_II, Q4_RR, 2, q4_timesteps, main_title="Question 04")
169     if q4_timesteps > 50:
170         break
171     q4_timesteps += 1
172
173 print(f"Last ten time steps of {q4_timesteps-1}")
174 print("\tS_i(t):\t")
175 print(printPrettyMatrix("", Q4_SS[:, q4_timesteps-10:q4_timesteps]))
176 print("\tI_i(t):\t")
177 print(printPrettyMatrix("", Q4_II[:, q4_timesteps-10:q4_timesteps]))
178 print("\tR_i(t):\t")
179 print(printPrettyMatrix("", Q4_RR[:, q4_timesteps-10:q4_timesteps]))
180
181
182 #####
183 # Homework 05: Question 5
184 #####
185 print("\nQuestion 5:")
186 print("\tInstead, suppose you remove the connection from node 2 to node 1\n",
187       "\tand replace it with a self-loop on 2 with weight 1. If nothing else\n",
188       "\tchanges in the network (and the infection starts at node 1), would \n",
189       "\tthis change the spread of the epidemic? Can you predict how many\n",
190       "\tpeople in node 2 would eventually get sick?\n")
191 Q5_og_infection = [0.0, 0.01, 0.0, 0.0, 0.0]
192 Q5_graph_adj = np.array([[0.8, 0.0, 0.0, 0.05, 0.15],
193                          [0.0, 1.0, 0.0, 0.0, 0.0],
194                          [0.3, 0.2, 0.5, 0.0, 0.0],
195                          [0.0, 0.0, 0.05, 0.95, 0.0],
196                          [0.0, 0.0, 0.0, 0.2, 0.8]])
197
198 q5_timesteps = 0
199
200 while True:
201     Q5_SS, Q5_II, Q5_RR = SIR(beta, gamma, Q5_graph_adj, Q5_og_infection,
202                               q5_timesteps+1, node = 2, show_plot=False, main_title="Question 05")
203     if q5_timesteps % 10 == 0:
204         print(f"Step:\t{q5_timesteps}")
205         print(f"Node 2 infection %:{Q5_II[:, q5_timesteps][1]}")
206         print(f"The rest of the network {Q5_II[:, q5_timesteps]}")
207         showPlot(Q5_SS, Q5_II, Q5_RR, 2, q5_timesteps, "Question 05")
208
209     if q5_timesteps > 50:
210         break
211     q5_timesteps += 1
212
213 print(f"Last ten time steps of {q5_timesteps-1}")
214 print("\tS_i(t):\t")
215 print(printPrettyMatrix("", Q5_SS[:, q5_timesteps-10:q5_timesteps]))
216 print("\tI_i(t):\t")
217 print(printPrettyMatrix("", Q5_II[:, q5_timesteps-10:q5_timesteps]))
218 print("\tR_i(t):\t")
219 print(printPrettyMatrix("", Q5_RR[:, q5_timesteps-10:q5_timesteps]))

```