Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

По дисциплине «Теоретические интеллектуальные и информационные технологии»

Тема: «Имитационное моделирование»

Выполнил:

Студент 1 курса

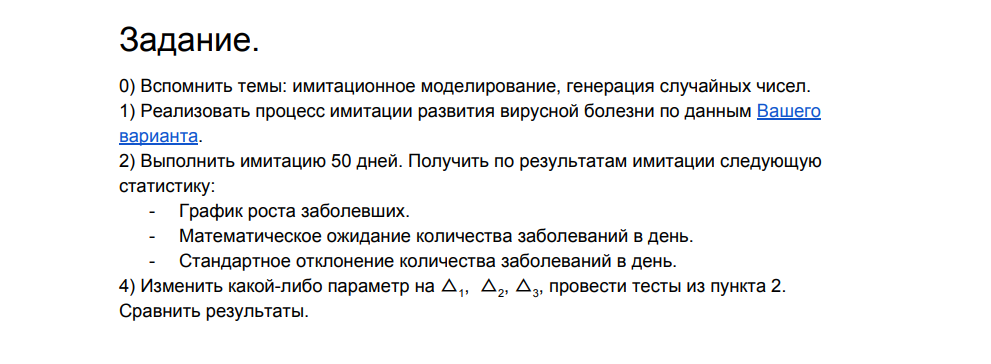
Группы ИИ-21(1)

Кирилович А.А.

Проверил:

Анфилец С. В.

Брест 2022

Вариант 5:





Код:

import numpy as np

import numpy.random as rand

import matplotlib.pyplot as plt

def N(m, scale):

    r = rand.normal(m, scale, size=None)

    if r < 0: r = 0

    return r

def U(c, d):

    r = rand.uniform(c-d, c+d, size=None)

    if r < 0: r = 0

    return r

def E(scale):

    r = rand.exponential(scale, size=None)

    if r < 0: r = 0

    return r

class Virus:

    def \_\_init\_\_(self):

        self.I = E(2.5)

corona = Virus()

class Environment:

    def \_\_init\_\_(self):

        self.AP = N(40, 5.5)

        self.M = N(10, 1.5)

        self.T = E(3.5)

region = Environment()

class Agent:

    day = None

    def \_\_init\_\_(self):

        self.m = rand.uniform(0, 50)

        self.is\_ill = False

        self.is\_know = False

        self.is\_on\_treatment = False

        self.is\_dead = False

        self.SC = int(N(self.m, 5))

        self.R = 1

        self.HA = N(4, 1.5)

        self.A = N(region.AP, 20)

        self.RT = int(2 + N(30 - 2.2 \* self.HA, 4 - 0.15 \* self.HA))

        self.DR = self.A \* 0.11 + U(5 - 0.05 \* self.HA - 0.2 \* region.M, 2 - 0.1 \* self.HA)

    def setRT(self):

        self.RT = int(2 + N(30 - 2.2 \* self.HA - 0.8 \* region.M,\

             4 - 0.15 \* self.HA - 0.12 \* region.M))

initial\_amount = int(rand.uniform(5, 11))

people = [Agent() for i in range(initial\_amount)]

for i in range(initial\_amount):

    people[i].is\_ill = True

    people[i].day = 0

data = []

k\_is\_dead = 0

k\_is\_cured = 0

data.append([0, len(people), k\_is\_dead, k\_is\_cured])

print("%3s %8s %6s %6s" % ("DAY", "Infected", "Died", "Cured"))

print("%3d %8d %6d %6d" % (0, len(people), k\_is\_dead, k\_is\_cured))

for day in range(30):

    for i in range(len(people)):

        if people[i].is\_know == True:

            continue

        probability\_T = rand.uniform(0, 100)

        if probability\_T < region.T:

            people[i].is\_know = True

            people[i].R = U(5, 5)

            people[i].SC /= people[i].R

            people[i].SC = int(people[i].SC)

    for i in range(len(people)):

        probability\_HA = rand.uniform(0, 100)

        if people[i].HA < 2:

            if probability\_HA < 0.1 \* 100:

                people[i].is\_on\_treatment = True

                people[i].setRT()

        else:

            if probability\_HA < 0.01 \* 100:

                people[i].is\_on\_treatment = True

                people[i].setRT()

    index = []

    for i in range(len(people)):

        probability\_DR = rand.uniform(0, 100)

        if probability\_DR < people[i].DR:

            people[i].is\_dead = True

            k\_is\_dead += 1

            index.append(i)

    for i in index:

        people[i] = None

    for i in index:

        people.remove(None)

    index = []

    for i in range(len(people)):

        if people[i].day + people[i].RT == day:

            people[i].is\_ill = False

            k\_is\_cured += 1

            index.append(i)

    for i in index:

        people[i] = None

    for i in index:

        people.remove(None)

    k\_new = 0

    for person in people:

        if person.is\_on\_treatment == True:

            continue

        k\_new += person.SC

    people += [Agent() for i in range(k\_new)]

    index = []

    for i in range(len(people)):

        if people[i].is\_ill == True:

            continue

        probability\_I = rand.uniform(0, 100)

        if probability\_I < corona.I:

            people[i].is\_ill = True

            people[i].day = day

        else:

            index.append(i)

    for i in index:

        people[i] = None

    for i in index:

        people.remove(None)

    data.append([day + 1, len(people), k\_is\_dead, k\_is\_cured])

    print("%3d %8d %6d %6d" % (day + 1, len(people), k\_is\_dead, k\_is\_cured))

data[0].append(0)

data[0].append(0)

data[0].append(0)

for i in range(1, len(data)):

    data[i].append(data[i][1] - data[i - 1][1])

    data[i].append(data[i][2] - data[i - 1][2])

    data[i].append(data[i][3] - data[i - 1][3])

print("\n\n")

print("%3s %9s %6s %6s %9s %6s %6s" % ("DAY", "InfectedG", "DiedG", "CuredG", "Infected+", "Died+", "Cured+"))

for i in range(len(data)):

    print("%3d %8d %6d %6d %9d %6d %6d" % (data[i][0], data[i][1], data[i][2], data[i][3], \

        data[i][4], data[i][5], data[i][6]))

mean = np.mean(data[4])

std = np.std(data[4])

print(f"\n\nMathematical expectation: {int(mean)}; Standard deviation: {int(std)}.")

fig = plt.figure(constrained\_layout=True)

gs = fig.add\_gridspec(nrows=3, ncols=3)

ax1 = fig.add\_subplot(gs[:-1, :])

ax1.set\_title('Infected')

ax2 = fig.add\_subplot(gs[-1, :-1])

ax2.set\_title('Died')

ax3 = fig.add\_subplot(gs[-1, -1])

ax3.set\_title('Cured')

day = [i for i in range(len(data))]

infectedG = [data[i][1] for i in range(len(data))]

ax1.bar(day, infectedG, color='red')

diedG = [data[i][2] for i in range(len(data))]

ax2.bar(day, diedG, color='black')

curedG = [data[i][3] for i in range(len(data))]

ax3.bar(day, curedG, color='green')

ax1.grid(), ax2.grid(), ax3.grid()

plt.show()

Вывод:   
