korona-Copy2

December 14, 2020

1

[1]:
```
#health status:
SUCCESS = 0
HEALTHY = 1
SICK = 2
INFECTIOUS = 3
IMMUNITY = 4
DEAD = 5
```

[2]:
```
#human role:
STUDENT = 0
TEACHER = 1
```

[3]:
```
from calendar import Calendar
import random
```

[4]:
```
class Human:
    def __init__( self, role, come_to_university, meeting_person_expected_value,
                  meeting_person_dispersion, incubation_period_expected_value,
                  incubation_period_dispersion, mortality,
                  illness_time_expected_value, illness_time_dispersion,
                  immunitet_period_expected_value, immunitet_period_dispersion):
        self.health = HEALTHY
        self.role = role
        self.come_to_university = come_to_university
        self.meeting_person_expected_value = meeting_person_expected_value
        self.meeting_person_dispersion = meeting_person_dispersion
        self.time_period = -1
        self.mortality = mortality
        self.incubation_period_expected_value = incubation_period_expected_value
        self.incubation_period_dispersion = incubation_period_dispersion
        self.illness_time_expected_value = illness_time_expected_value
        self.illness_time_dispersion = illness_time_dispersion
        self.immunitet_period_expected_value = immunitet_period_expected_value
        self.immunitet_period_dispersion = immunitet_period_dispersion
```

```
    def infectious(self, infectiouness):
        if self.health == HEALTHY and random.uniform(0, 1) < infectiouness:
            self.health = INFECTIOUS
            self.time_period = int(random.normalvariate(
                self.incubation_period_expected_value,
                self.incubation_period_dispersion))
            return True
        return False

    def process(self):
        if self.health == INFECTIOUS:
            self.time_period -= 1
            if self.time_period <= 0:
                self.health = SICK
                self.time_period = int(random.normalvariate(
                    self.illness_time_expected_value,
                    self.illness_time_dispersion
                ))
                return self.health
        if self.health == SICK:
            self.time_period -= 1
            if self.time_period <= 0:
                if random.uniform(0, 1) < self.mortality:
                    self.health = DEAD
                else:
                    self.health = IMMUNITY
                    self.time_period = int(random.normalvariate(
                        self.immunitet_period_expected_value,
                        self.immunitet_period_dispersion
                ))
                return self.health
        if self.health == IMMUNITY:
            self.time_period -= 1
            if self.time_period <= 0:
                self.health = HEALTHY
                self.time_period = -1
                return self.health
        return SUCCESS

    def is_come(self):
        if self.health != SICK and \
            random.uniform(0, 1) < self.come_to_university:
            return True
        return False

    def get_meetings(self):
        meetings = int(random.normalvariate(
```

```
            self.meeting_person_expected_value,
            self.meeting_person_dispersion))
        if meetings > 0:
            return meetings
        return 0
```

[5]:
```
class Population:
    def __init__( self, students_num, student_meeting_person_expected_value,
                  student_meeting_person_dispersion, student_come_to_university,
                  teachers_num, teacher_meeting_person_expected_value,
                  teacher_meeting_person_dispersion, teacher_come_to_university,
                  infectiousness, incubation_period_expected_value,
                  incubation_period_dispersion, mortality,
                  illness_time_expected_value, illness_time_dispersion,
                  immunitet_period_expected_value, immunitet_period_dispersion
                ):
        self.human_list = []
        #add students
        for i in range(students_num):
            self.human_list.append( Human(STUDENT,
                                          student_come_to_university,
                                          student_meeting_person_expected_value,
                                          student_meeting_person_dispersion,
                                          incubation_period_expected_value,
                                          incubation_period_dispersion,
                                          mortality,
                                          illness_time_expected_value,
                                          illness_time_dispersion,
                                          immunitet_period_expected_value,
                                          immunitet_period_dispersion)
                                    )
        #add teachers
        for i in range(teachers_num):
            self.human_list.append( Human(TEACHER,
                                          teacher_come_to_university,
                                          teacher_meeting_person_expected_value,
                                          teacher_meeting_person_dispersion,
                                          incubation_period_expected_value,
                                          incubation_period_dispersion,
                                          mortality,
                                          illness_time_expected_value,
                                          illness_time_dispersion,
                                          immunitet_period_expected_value,
                                          immunitet_period_dispersion)
                                    )
        self.dead = 0
```

```
        self.healthy = len(self.human_list)
        self.infectious = 0
        self.sick = 0
        self.immunity = 0
        self.mortality = mortality
        self.infectiousness = infectiousness

    def process( self ):
        #create comming list
        come_to_university = list()
        for human in self.human_list:
            result = human.process()
            if result == DEAD:
                self.dead += 1
                self.sick -= 1
                self.human_list.remove(human)
                continue

            elif result == SICK:
                self.infectious -= 1
                self.sick += 1
                continue

            elif result == IMMUNITY:
                self.sick -= 1
                self.immunity += 1
                continue

            elif result == HEALTHY:
                self.immunity -= 1
                self.healthy += 1
                continue

            if human.is_come():
                come_to_university.append(human)
        if len(come_to_university) == 0:
            return

        #meetings
        for human in come_to_university:
            if(human.health == INFECTIOUS):
                for i in range(human.get_meetings()):
                    if random.choice(come_to_university).infectious(self.
↪infectiousness):
                        self.healthy -= 1
                        self.infectious += 1
```

```
    def process_weekend( self ):
        for human in self.human_list:
            result = human.process()
            if result == DEAD:
                self.dead += 1
                self.sick -= 1
                self.human_list.remove(human)
            elif result == SICK:
                self.infectious -= 1
                self.sick += 1
            elif result == IMMUNITY:
                self.sick -= 1
                self.immunity += 1
            elif result == HEALTHY:
                self.immunity -= 1
                self.healthy += 1



    def first_infection( self ):
        human = random.choice(self.human_list)
        human.infectious(1.)
        self.healthy -= 1
        self.infectious += 1

    def add_infectious( self, human):
        human.infectious(self.infectiousness)
        self.healthy -= 1
        self.infectious += 1
```

[6]:
```
import datetime
```

[7]:
```
def autum_semestr_loop( population, drawer):
    try:
        calendar = Calendar()
        my_calendar = calendar.yeardayscalendar( year = 2020 )
        for season in range(2, 5):
            for month in range(3):
                if (season)*3+1+ month < 9:
                    continue
                if season == 4 and month > 0:
                    break
                for week in range(len(my_calendar[season][month])):
                    for day in range(7):
                        if not my_calendar[season][month][week][day] == 0:
                            if day == 6:
```

```
                                #print('sunday')
                                population.process_weekend()
                            else:
                                #print('{} - {}'.format(
                                #my_calendar[season][month][week][day],
                                #(season)*3+1+ month
                                #))
                                #print(str(season)+' '+str(month)+'␣
↪'+str(week)+' '+str(day))
                                population.process()
                                #print('{} {} {} {} {}'.format(
                                #    population.healthy,
                                #    population.infectious,
                                #    population.sick,
                                #    population.immunity,
                                #    population.dead
                                #    ))
                            date = datetime.date(year=2020,
                                                 month=(season)*3 +1 + month,
                                                 ␣
↪day=my_calendar[season][month][week][day])
                            drawer.add(population, date)
    except IndexError:
        return
```

2 1.

3

[8]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

[9]:
```
class Drawer2D:
    def __init__(self):
        self.healthy = []
        self.infectious = []
        self.sick = []
        self.dead = []
        self.date = []
        self.immunity = []

    def add(self, population, date):
        self.healthy.append(population.healthy+population.immunity)
        self.infectious.append(population.infectious)
        self.sick.append(population.sick)
```

```
        self.dead.append(population.dead)
        self.immunity.append(population.immunity)
        self.date.append(date)

    def draw(self):
        fig, ax = plt.subplots(figsize=(15, 9))
        ax.plot(self.date, self.healthy, 'g', label='    ')
        ax.plot(self.date, self.infectious, 'b', label='     ')
        ax.plot(self.date, self.sick, 'r', label='   ')
        ax.plot(self.date, self.dead, 'black', label='   ')
        ax.plot(self.date, self.immunity, 'c', label='    ')
        ax.legend()
        ax.set_xlabel('  ')
        ax.set_ylabel('          ')
        ax.set_title("                    ")
```

[10]:
```
def experiment1():
    students_num = 100
    student_meeting_person_expected_value = 10
    student_meeting_person_dispersion = 2
    student_come_to_university = 0.9
    teachers_num = 0
    teacher_meeting_person_expected_value = 0
    teacher_meeting_person_dispersion = 0
    teacher_come_to_university = 0
    infectiousness = 0.2
    incubation_period_expected_value = 14
    incubation_period_dispersion = 2
    mortality = 0.1
    illness_time_expected_value = 20
    illness_time_dispersion = 15
    immunitet_period_expected_value = 40
    immunitet_period_dispersion = 10
    population = Population(students_num, student_meeting_person_expected_value,
                        student_meeting_person_dispersion,␣
↪student_come_to_university,
                        teachers_num, teacher_meeting_person_expected_value,
                        teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                        infectiousness, incubation_period_expected_value,
                        incubation_period_dispersion,
                        mortality,
                        illness_time_expected_value, illness_time_dispersion,
                        immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
    population.first_infection()
    drawer2D = Drawer2D()
```

```
    autum_semestr_loop( population, drawer2D)
    return drawer2D
```

4 2.

4.1

[11]:
```
def experiment2():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    population = Population(students_num, student_meeting_person_expected_value,
                        student_meeting_person_dispersion,␣
↪student_come_to_university,
                        teachers_num, teacher_meeting_person_expected_value,
                        teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                        infectiousness, incubation_period_expected_value,
                        incubation_period_dispersion,
                        mortality,
                        illness_time_expected_value, illness_time_dispersion,
                        immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
    population.first_infection()
    drawer2D = Drawer2D()
    autum_semestr_loop( population, drawer2D)
    return drawer2D
```

5 3.

5.1 ,

[12]:
```
def experiment3():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799*0.7 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
                               #https://www.kommersant.ru/doc/4432704
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    population = Population(students_num, student_meeting_person_expected_value,
                        student_meeting_person_dispersion,␣
↪student_come_to_university,
                        teachers_num, teacher_meeting_person_expected_value,
                        teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                        infectiousness, incubation_period_expected_value,
                        incubation_period_dispersion,
                        mortality,
                        illness_time_expected_value, illness_time_dispersion,
                        immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
    population.first_infection()
    drawer2D = Drawer2D()
    autum_semestr_loop( population, drawer2D)
    return drawer2D
```

6 4.

6.1

[13]:
```
def experiment4():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799*0.7 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
                               #https://www.kommersant.ru/doc/4432704
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 20
    immunitet_period_dispersion = 10
    population = Population(students_num, student_meeting_person_expected_value,
                        student_meeting_person_dispersion,␣
↪student_come_to_university,
                        teachers_num, teacher_meeting_person_expected_value,
                        teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                        infectiousness, incubation_period_expected_value,
                        incubation_period_dispersion,
                        mortality,
                        illness_time_expected_value, illness_time_dispersion,
                        immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
    population.first_infection()
    drawer2D = Drawer2D()
    autum_semestr_loop( population, drawer2D)
    return drawer2D
```

7 ,

[14]:
```
import pylab
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
```

[15]:
```
class Drawer3D:
    def __init__(self, k_name):
        self.Y = []
        self.X = []
        self.date = []
        self.Zhealthy = []
        self.Zinfectious = []
        self.Zsick = []
        self.Zdead = []
        self.Zimmunity = []
        self.healthy = []
        self.infectious = []
        self.sick = []
        self.dead = []
        self.immunity = []
        self.k_name = k_name

    def setK(self, k):
        self.X.append([i for i in range(len(self.date))])
        self.Y.append([k for i in range(len(self.date))].copy())
        self.Zhealthy.append(self.healthy.copy())
        self.Zinfectious.append(self.infectious.copy())
        self.Zsick.append(self.sick.copy())
        self.Zdead.append(self.dead.copy())
        self.Zimmunity.append(self.immunity.copy())
        self.date = []
        self.healthy = []
        self.infectious = []
        self.sick = []
        self.dead = []
        self.immunity = []

    def add( self, population, date):
        self.date.append(date)
        self.healthy.append(population.healthy)
        self.infectious.append(population.infectious)
        self.sick.append(population.sick)
        self.dead.append(population.dead)
        self.immunity.append(population.immunity)

    def draw(self):
```

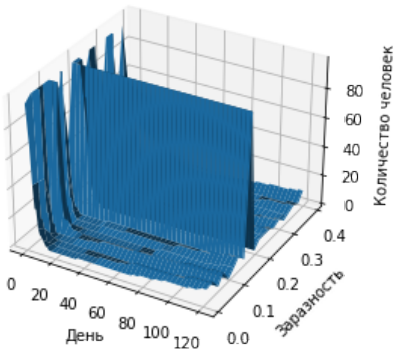
```
        x = np.array(self.X)
        y = np.array(self.Y)
        zHealthy = np.array(self.Zhealthy)
        zInfectious = np.array(self.Zinfectious)
        zImmunity = np.array(self.Zimmunity)
        zDead = np.array(self.Zdead)
        zSick = np.array(self.Zsick)

        fig = plt.figure(figsize=(15, 15))
        ax = fig.add_subplot(3, 2, 1, projection='3d')
        ax.set_title('    ')
        ax.set_xlabel('  ')
        ax.set_ylabel(self.k_name)
        ax.set_zlabel('          ')
        surf = ax.plot_surface(x, y, zHealthy)

        ax = fig.add_subplot(3, 2, 2, projection='3d')
        ax.set_title('     ')
        ax.set_xlabel('  ')
        ax.set_ylabel(self.k_name)
        ax.set_zlabel('          ')
        surf = ax.plot_surface(x, y, zInfectious)

        ax = fig.add_subplot(3, 2, 3, projection='3d')
        ax.set_title('    ')
        ax.set_xlabel('  ')
        ax.set_ylabel(self.k_name)
        ax.set_zlabel('          ')
        surf = ax.plot_surface(x, y, zSick)

        ax = fig.add_subplot(3, 2, 4, projection='3d')
        ax.set_title('     ')
        ax.set_xlabel('  ')
        ax.set_ylabel(self.k_name)
        ax.set_zlabel('          ')
        surf = ax.plot_surface(x, y, zImmunity)

        ax = fig.add_subplot(3, 2, 5, projection='3d')
        ax.set_title('    ')
        ax.set_xlabel('  ')
        ax.set_ylabel(self.k_name)
        ax.set_zlabel('          ')
        surf = ax.plot_surface(x, y, zDead)
```

[16]:
```
%%time
students_num = 100
student_meeting_person_expected_value = 150
```

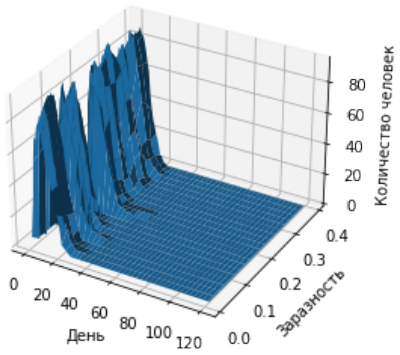
```
student_meeting_person_dispersion = 50
student_come_to_university = 0.5
teachers_num = 0
teacher_meeting_person_expected_value = 200
teacher_meeting_person_dispersion = 150
teacher_come_to_university = 0.3
incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
 ↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
incubation_period_dispersion = 5
mortality = 0.06
illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
 ↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
illness_time_dispersion = 5
immunitet_period_expected_value = 240
immunitet_period_dispersion = 100
drawer3D = Drawer3D('      ')
for infectiousness in np.arange(0.01, 0.4, 0.02):
    population = Population(students_num, student_meeting_person_expected_value,
                        student_meeting_person_dispersion,␣
 ↪student_come_to_university,
                        teachers_num, teacher_meeting_person_expected_value,
                        teacher_meeting_person_dispersion,␣
 ↪teacher_come_to_university,
                        infectiousness, incubation_period_expected_value,
                        incubation_period_dispersion,
                        mortality,
                        illness_time_expected_value, illness_time_dispersion,
                        immunitet_period_expected_value,␣
 ↪immunitet_period_dispersion)
    population.first_infection()
    autum_semestr_loop( population, drawer3D)
    drawer3D.setK(infectiousness)
drawer3D.draw()
```

```
CPU times: user 866 ms, sys: 177 µs, total: 866 ms
Wall time: 887 ms
```

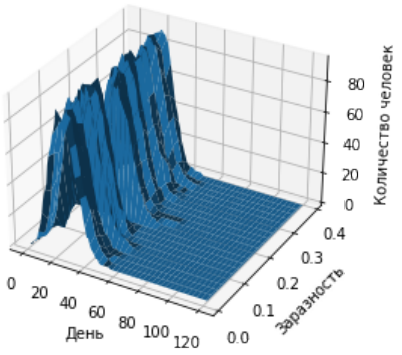
Здоровые



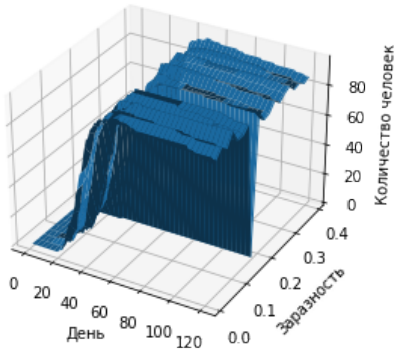
Зараженные



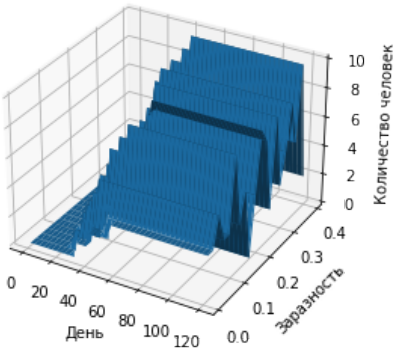
Больные



Иммунитет



Умерли



, . .

8 5.

9

[17]:
```
def experiment5():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("      ")
    for infectiousness in np.arange(0.0, 0.95, 0.05):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                        student_meeting_person_dispersion,␣
↪student_come_to_university,
                        teachers_num, teacher_meeting_person_expected_value,
                        teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                        infectiousness, incubation_period_expected_value,
                        incubation_period_dispersion,
                        mortality,
                        illness_time_expected_value, illness_time_dispersion,
                        immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(infectiousness)
    return drawer3D
```

10 6.

11

[18]:
```
def experiment6():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("                    ")
    for student_come_to_university in np.arange(0., 0.95, 0.05):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                            student_meeting_person_dispersion,␣
↪student_come_to_university,
                            teachers_num, teacher_meeting_person_expected_value,
                            teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                            infectiousness, incubation_period_expected_value,
                            incubation_period_dispersion,
                            mortality,
                            illness_time_expected_value, illness_time_dispersion,
                            immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(student_come_to_university)
    return drawer3D
```

12 7.

13

[19]:
```
def experiment7():
    students_num = 19000
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("                  ")
    for student_meeting_person_expected_value in np.arange(10, 200, 10):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                           student_meeting_person_dispersion,␣
↪student_come_to_university,
                           teachers_num, teacher_meeting_person_expected_value,
                           teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                           infectiousness, incubation_period_expected_value,
                           incubation_period_dispersion,
                           mortality,
                           illness_time_expected_value, illness_time_dispersion,
                           immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(student_meeting_person_expected_value)
    return drawer3D
```

14 8.

15

[20]:
```
def experiment8():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("                        ")
    for teacher_come_to_university in np.arange(0.05, 1, 0.05):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                            student_meeting_person_dispersion,␣
↪student_come_to_university,
                            teachers_num, teacher_meeting_person_expected_value,
                            teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                            infectiousness, incubation_period_expected_value,
                            incubation_period_dispersion,
                            mortality,
                            illness_time_expected_value, illness_time_dispersion,
                            immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(teacher_come_to_university)
    return drawer3D
```

16 9.

17

[21]:
```
def experiment9():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("                    ")
    for incubation_period_expected_value in np.arange(1, 20, 1):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                            student_meeting_person_dispersion,␣
↪student_come_to_university,
                            teachers_num, teacher_meeting_person_expected_value,
                            teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                            infectiousness, incubation_period_expected_value,
                            incubation_period_dispersion,
                            mortality,
                            illness_time_expected_value, illness_time_dispersion,
                            immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(incubation_period_expected_value)
    return drawer3D
```

18 10.

19

[22]:
```
def experiment10():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("      ")
    for mortality in np.arange(0, 1, 0.05):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                            student_meeting_person_dispersion,␣
↪student_come_to_university,
                            teachers_num, teacher_meeting_person_expected_value,
                            teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                            infectiousness, incubation_period_expected_value,
                            incubation_period_dispersion,
                            mortality,
                            illness_time_expected_value, illness_time_dispersion,
                            immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(mortality)
    return drawer3D
```

20 11.

21

[23]:
```
def experiment11():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("                        ")
    for immunitet_period_expected_value in np.arange(10, 100, 10):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                           student_meeting_person_dispersion,␣
↪student_come_to_university,
                           teachers_num, teacher_meeting_person_expected_value,
                           teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                           infectiousness, incubation_period_expected_value,
                           incubation_period_dispersion,
                           mortality,
                           illness_time_expected_value, illness_time_dispersion,
                           immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(immunitet_period_expected_value)
    return drawer3D
```

22 12.

23

[24]:
```
def experiment12():
    students_num = 19000
    student_meeting_person_expected_value = 150
    student_meeting_person_dispersion = 50
    student_come_to_university = 0.5
    teachers_num = 3297
    teacher_meeting_person_expected_value = 200
    teacher_meeting_person_dispersion = 150
    teacher_come_to_university = 0.3
    infectiousness = 0.799 #https://rg.ru/2020/05/04/
↪issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.
↪html
    incubation_period_expected_value = 11 #https://iz.ru/989894/2020-03-22/
↪nazvan-srednii-inkubatcionnyi-period-koronavirusa
    incubation_period_dispersion = 5
    mortality = 0.06
    illness_time_expected_value = 20 #https://iz.ru/981482/2020-02-28/
↪voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa
    illness_time_dispersion = 5
    immunitet_period_expected_value = 240
    immunitet_period_dispersion = 100
    drawer3D = Drawer3D("           ")
    for illness_time_expected_value in np.arange(3, 30, 2):
        population = Population(students_num,␣
↪student_meeting_person_expected_value,
                        student_meeting_person_dispersion,␣
↪student_come_to_university,
                        teachers_num, teacher_meeting_person_expected_value,
                        teacher_meeting_person_dispersion,␣
↪teacher_come_to_university,
                        infectiousness, incubation_period_expected_value,
                        incubation_period_dispersion,
                        mortality,
                        illness_time_expected_value, illness_time_dispersion,
                        immunitet_period_expected_value,␣
↪immunitet_period_dispersion)
        population.first_infection()
        autum_semestr_loop( population, drawer3D)
        drawer3D.setK(illness_time_expected_value)
    return drawer3D
```

[25]:
```
import multiprocessing
num_of_cpu = multiprocessing.cpu_count()
print (num_of_cpu)
```

```
32
```

[26]:
```
experiment_list = [experiment1,
                   experiment2,
                   experiment3,
                   experiment4,
                   experiment5,
                   experiment6,
                   experiment7,
                   experiment8,
                   experiment9,
                   experiment10,
                   experiment11,
                   experiment12]
```

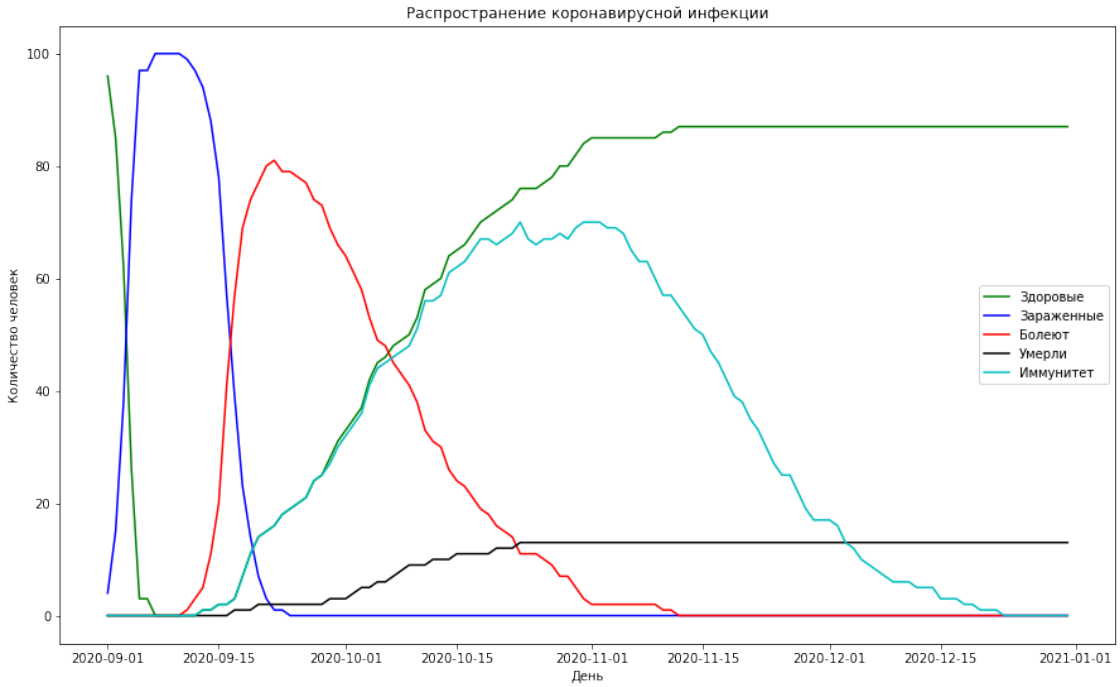
[27]:
```
def experiment_do(i):
    return experiment_list[i]()
```

[28]:
```
%%time
pool = multiprocessing.Pool(processes=num_of_cpu)
res = pool.map(experiment_do, range(len(experiment_list)))
```

```
CPU times: user 1.09 s, sys: 102 ms, total: 1.19 s
Wall time: 3min 11s
```

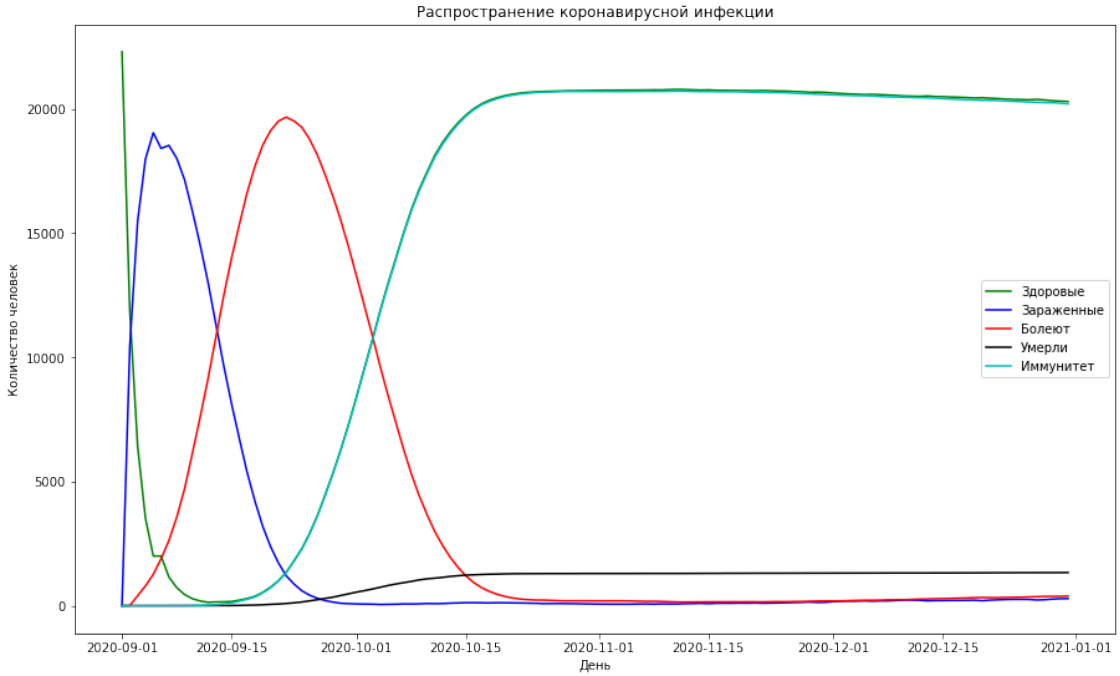
24 1

[29]:
```
res[0].draw()
```



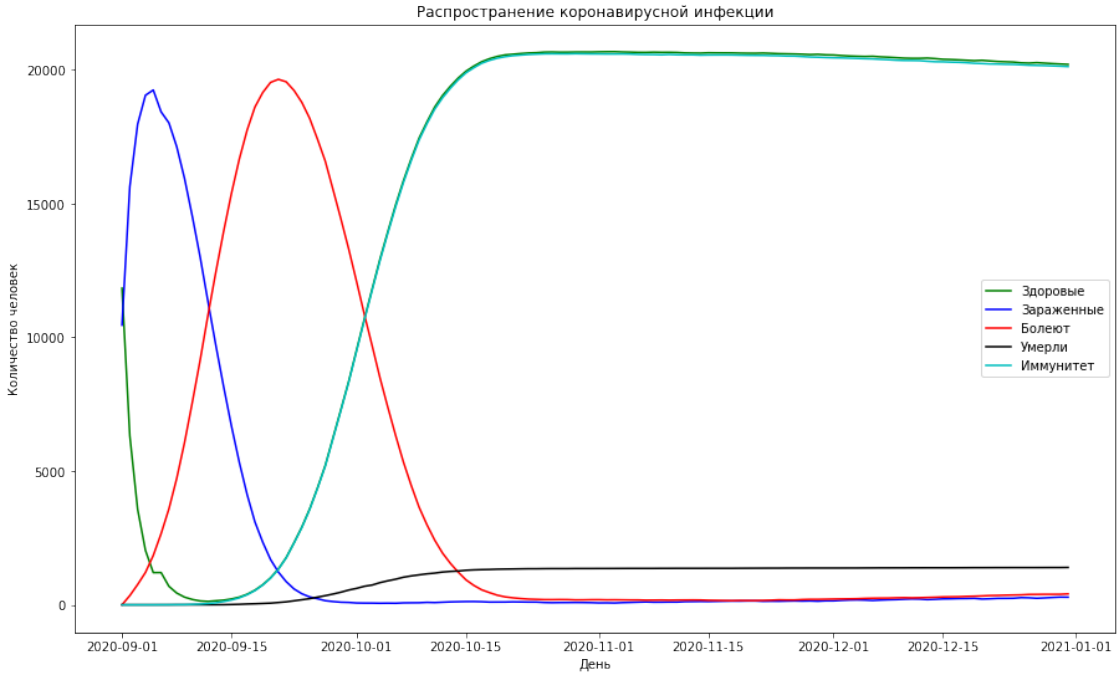
25 2

[30]:
```
res[1].draw()
```



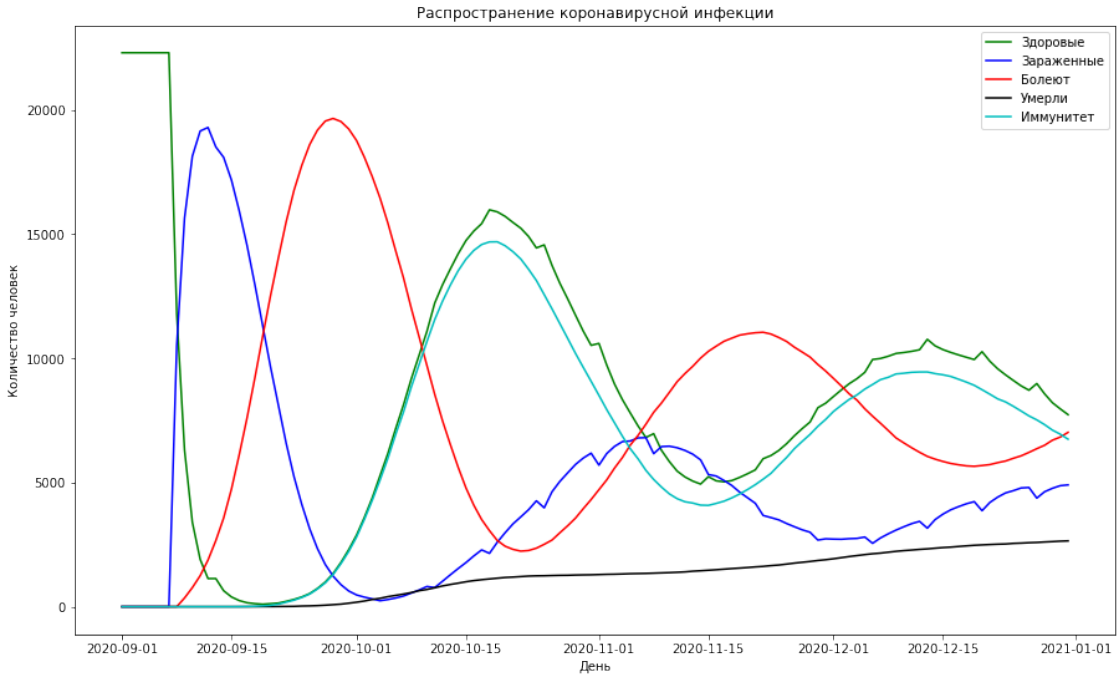
26 3

[31]:
```
res[2].draw()
```



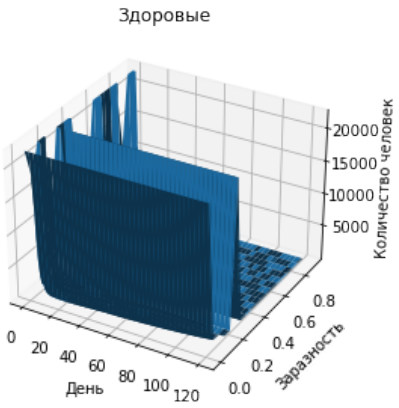
27 4

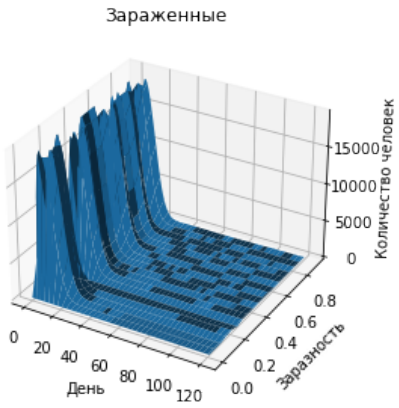
[32]:
```
res[3].draw()
```

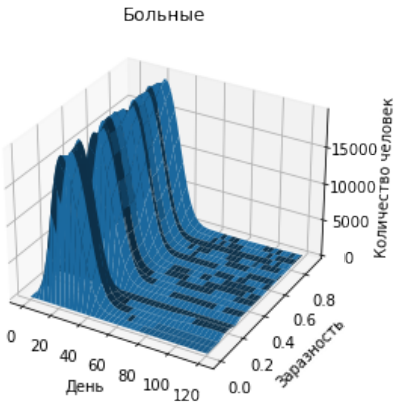


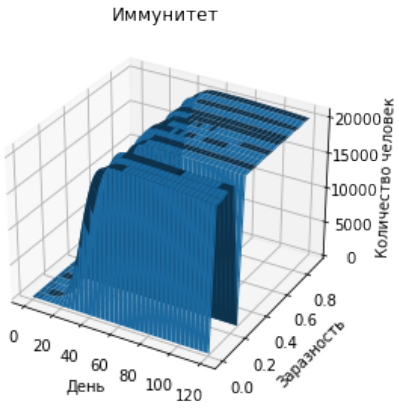
28 **5**

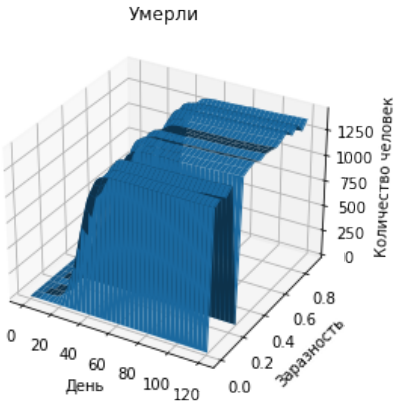
[33]:
```
res[4].draw()
```





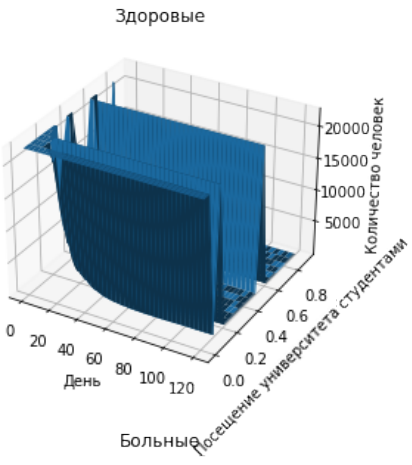


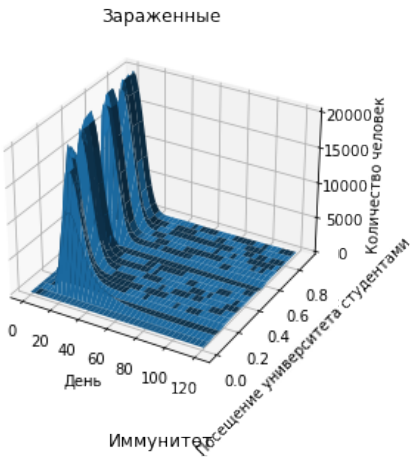


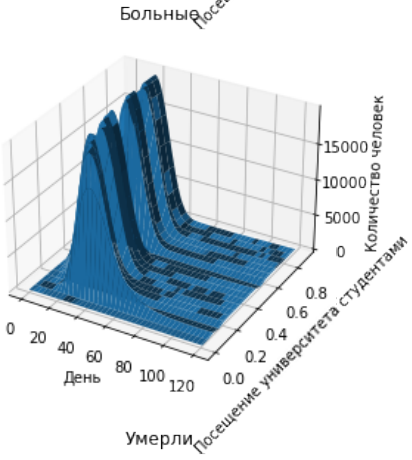


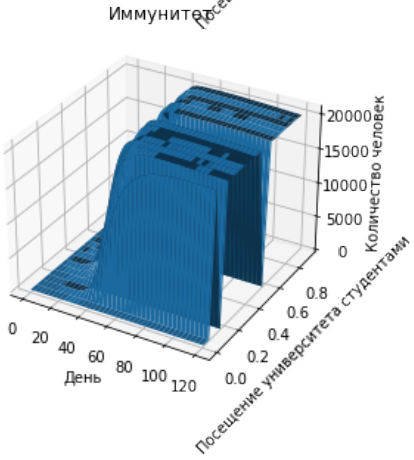
29 6

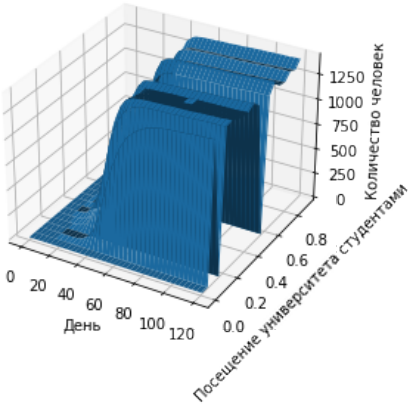
[34]:
```
res[5].draw()
```





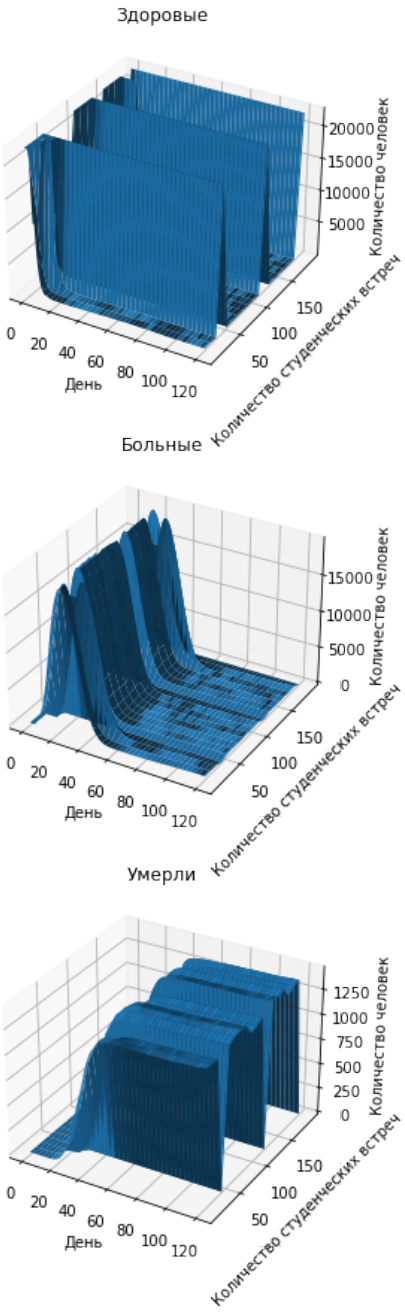






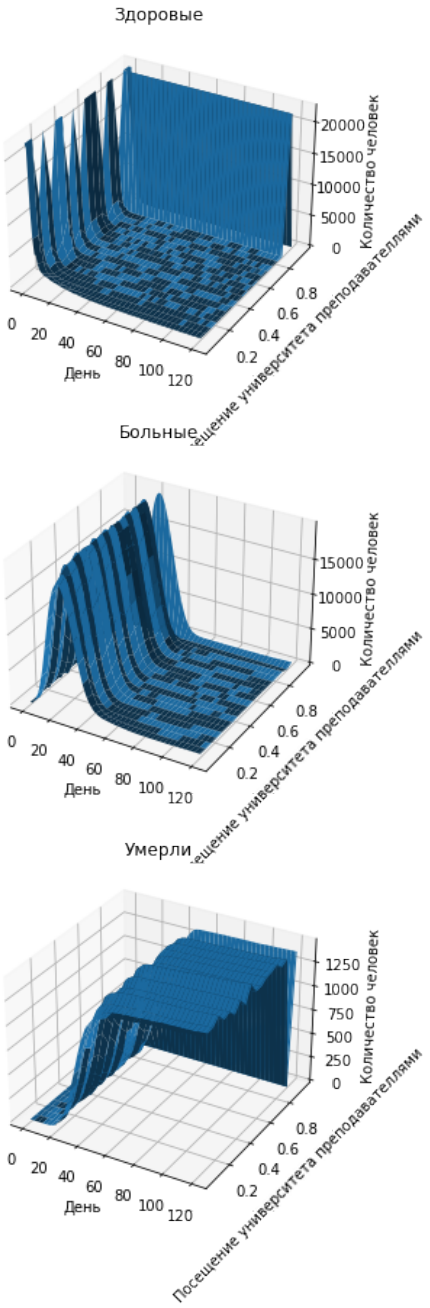
30 **7**

[35]:
```
res[6].draw()
```



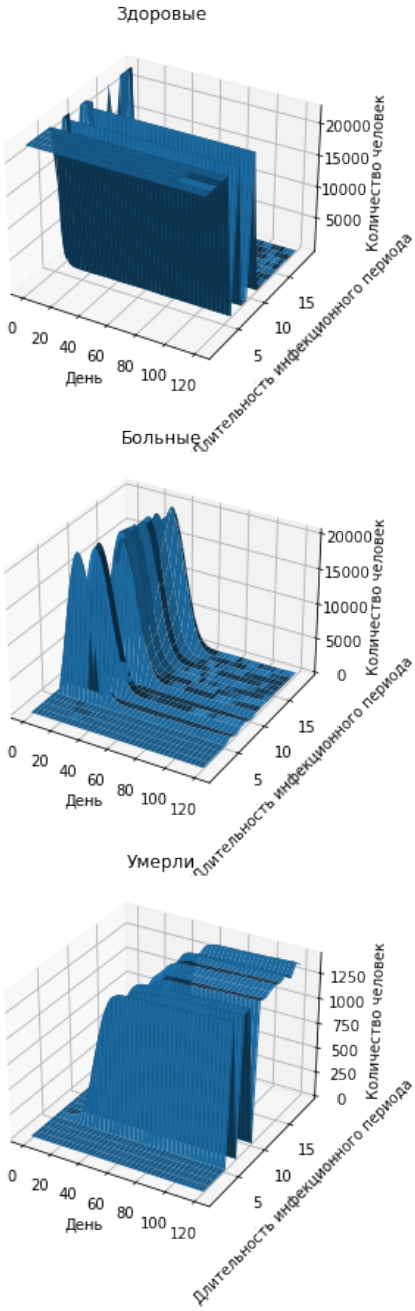
31 **8**

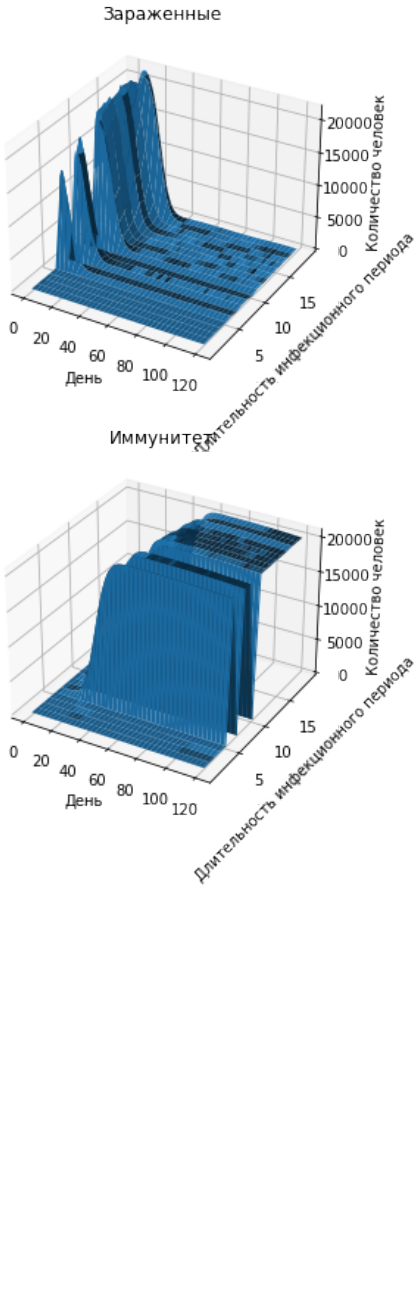
[36]:
```
res[7].draw()
```



32 9

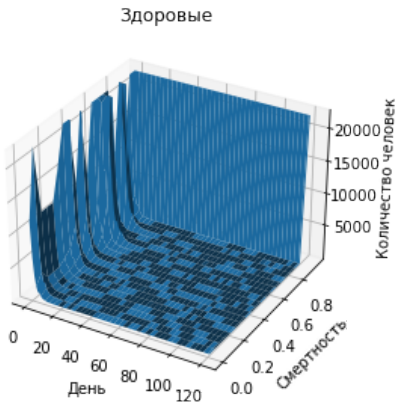
[37]:
```
res[8].draw()
```

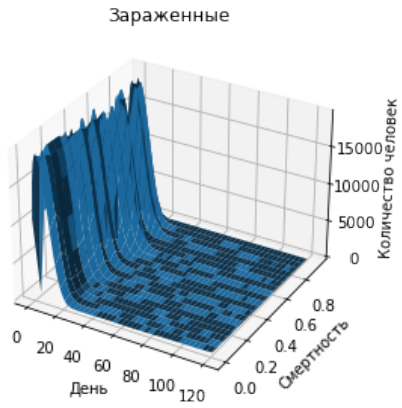


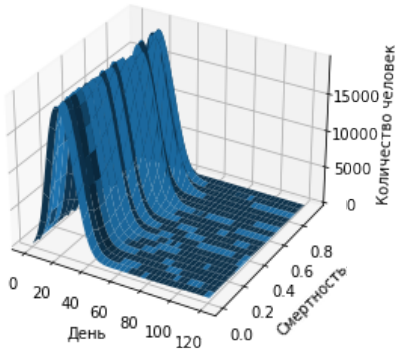


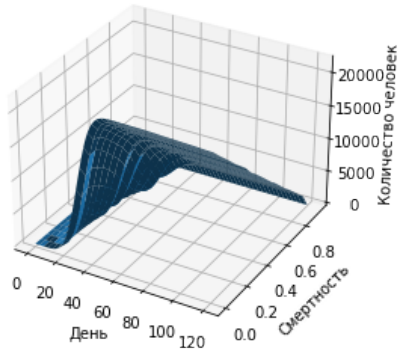
33 10

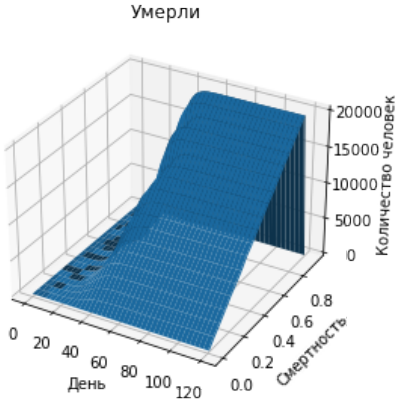
[38]:
```
res[9].draw()
```





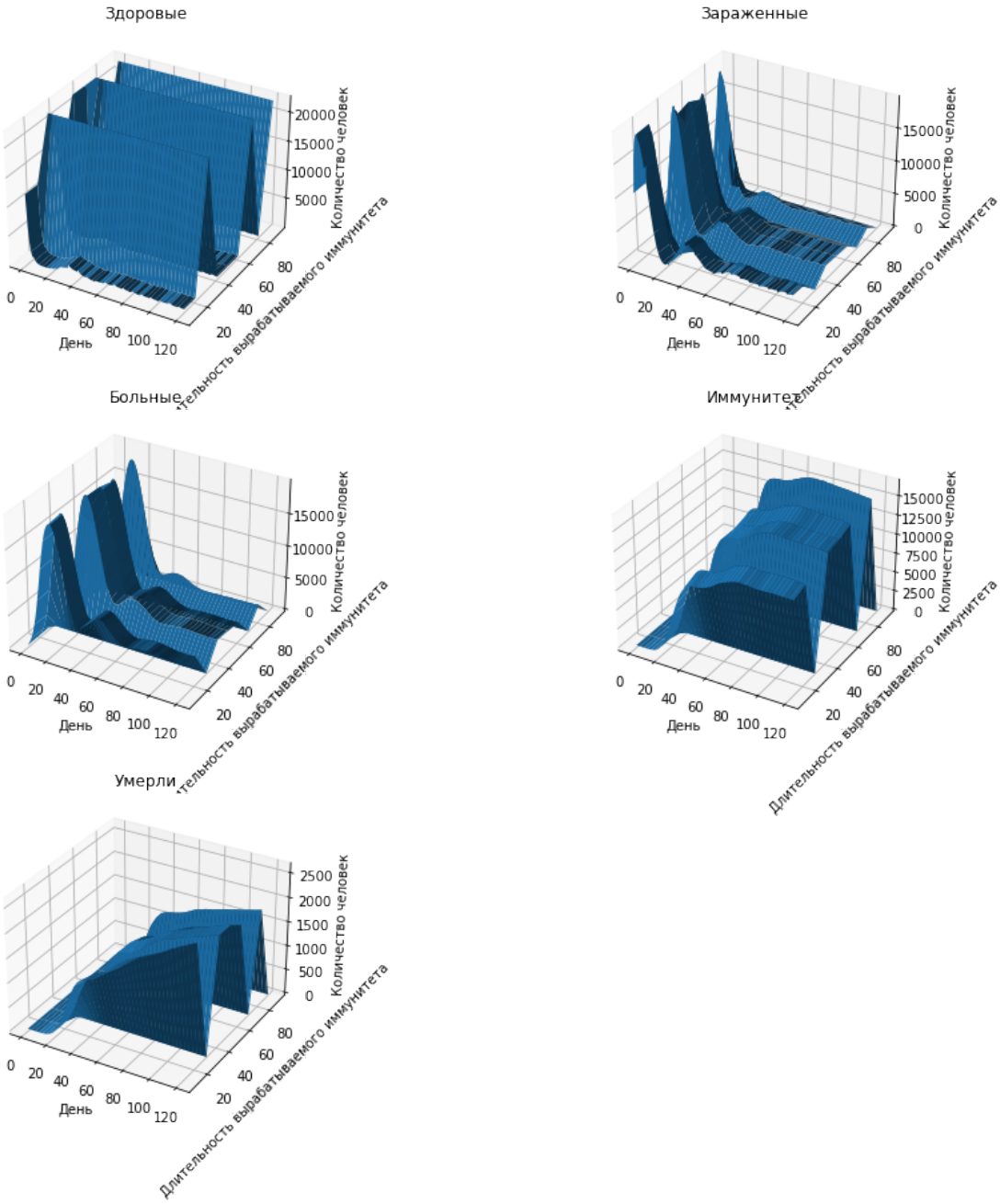






34 **11**

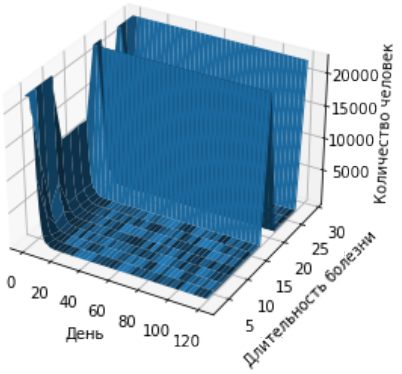
[39]:
```
res[10].draw()
```



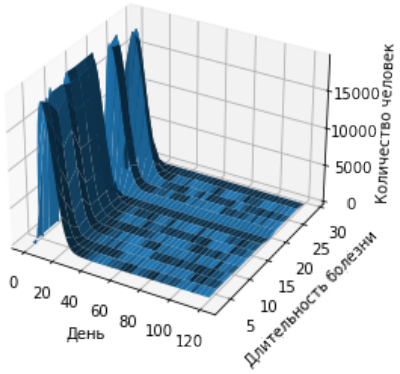
35 **12**

[40]:
```
res[11].draw()
```

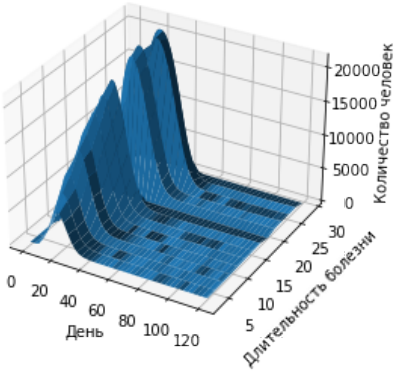
Здоровые



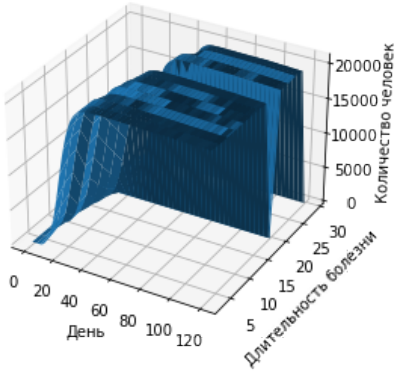
Зараженные



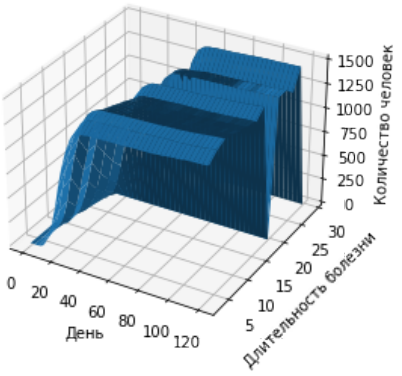
Больные



Иммунитет



Умерли



[ ]:

[ ]:

[ ]: