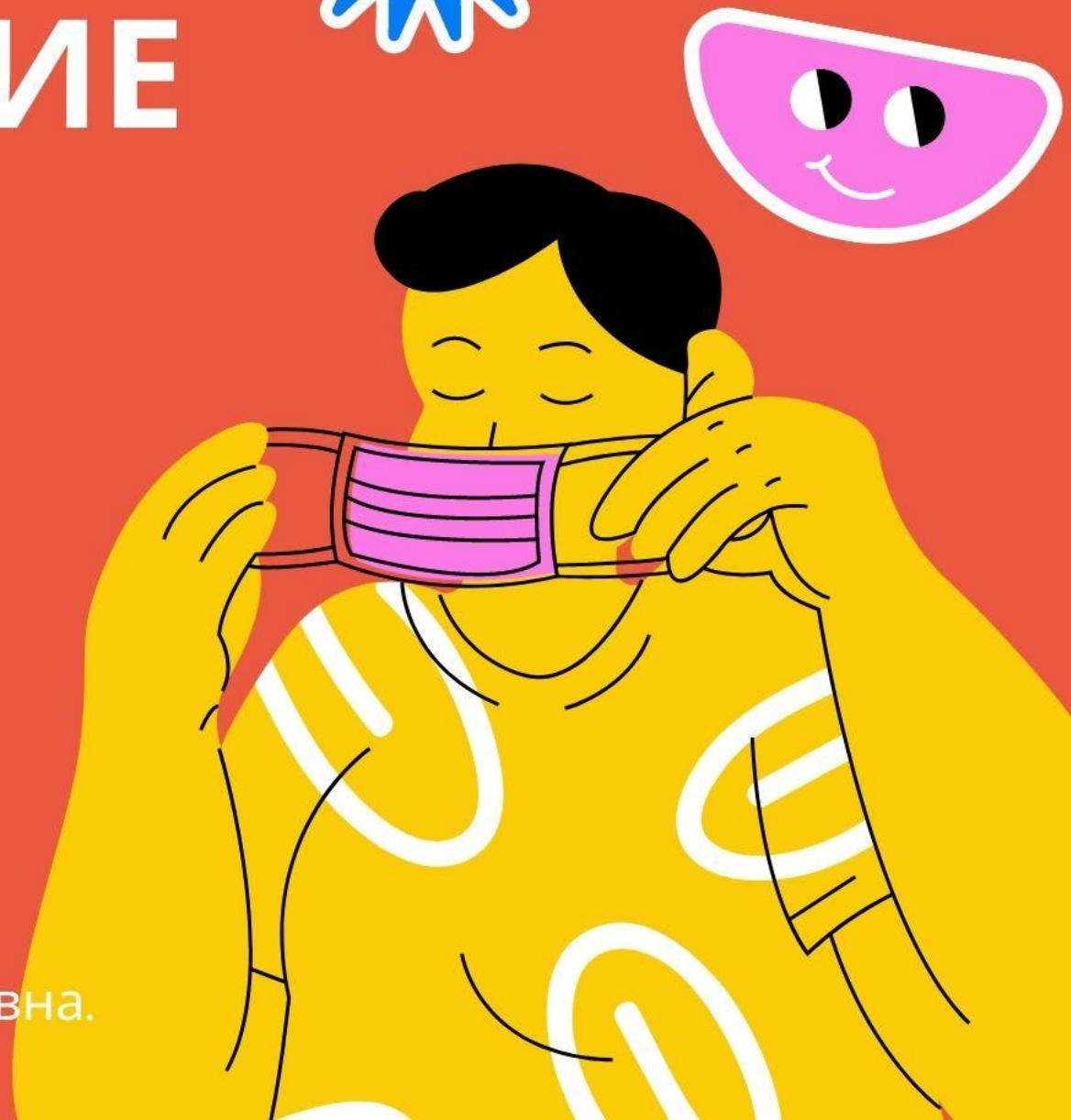
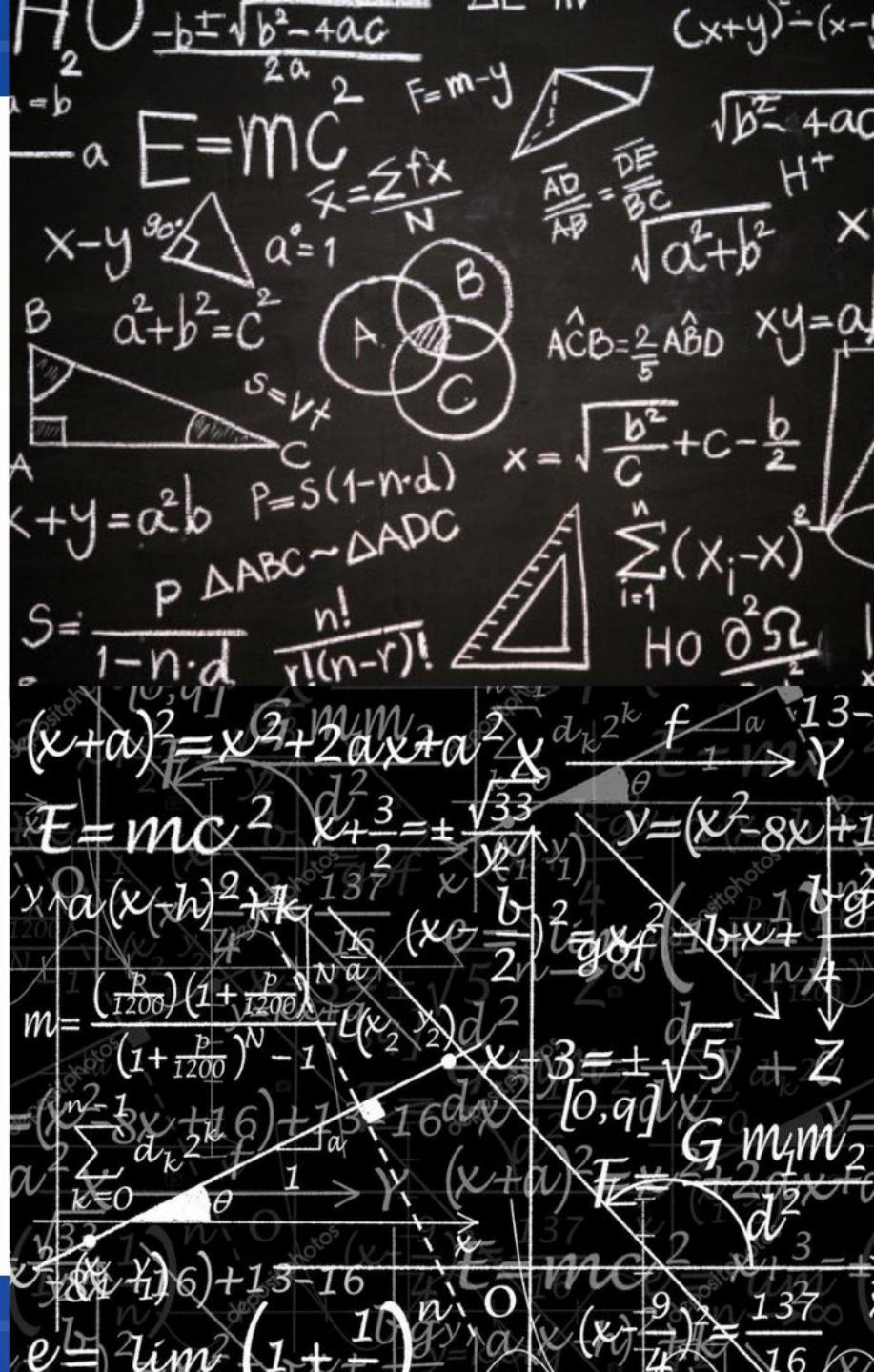
ФГБОУ ВО «Российская академия народного характера и государственной службы при Президенте Российской Федерации»
Поволжский институт управления имени П.А. Столыпина

МОДЕЛИРОВАНИЕ СОЦИАЛЬНЫХ ПРОЦЕССОВ В УСЛОВИЯХ ПАНДЕМИИ



Войтус Антон Михайлович, 1 курс;
Мохан Никита Дмитриевич, 1 курс;
Менькова Анастасия Михайловна, 1 курс;
Научный руководитель: Сытник Наталия Сергеевна.

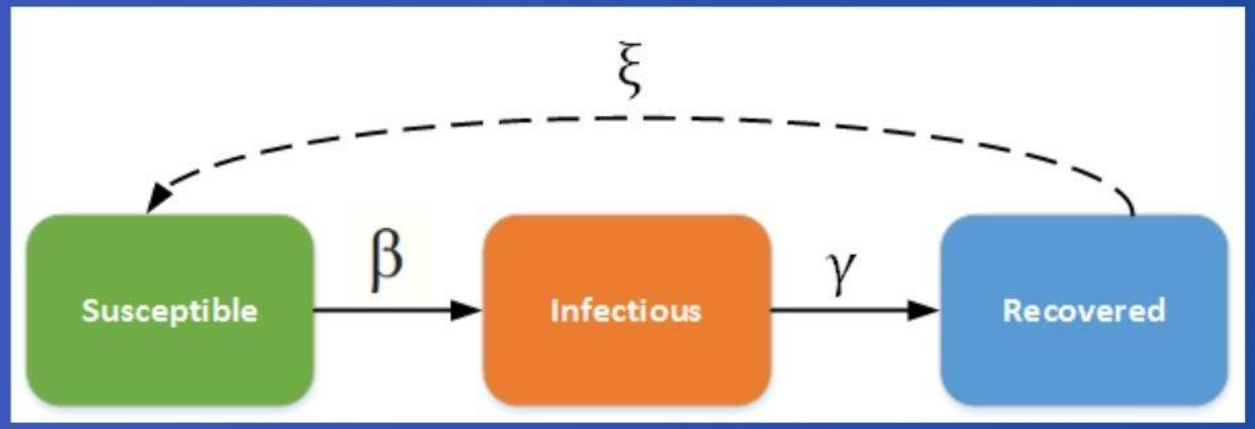
Дифференциальные уравнения и теории вероятности - это два инструмента помогающие моделировать распространения заболеваний



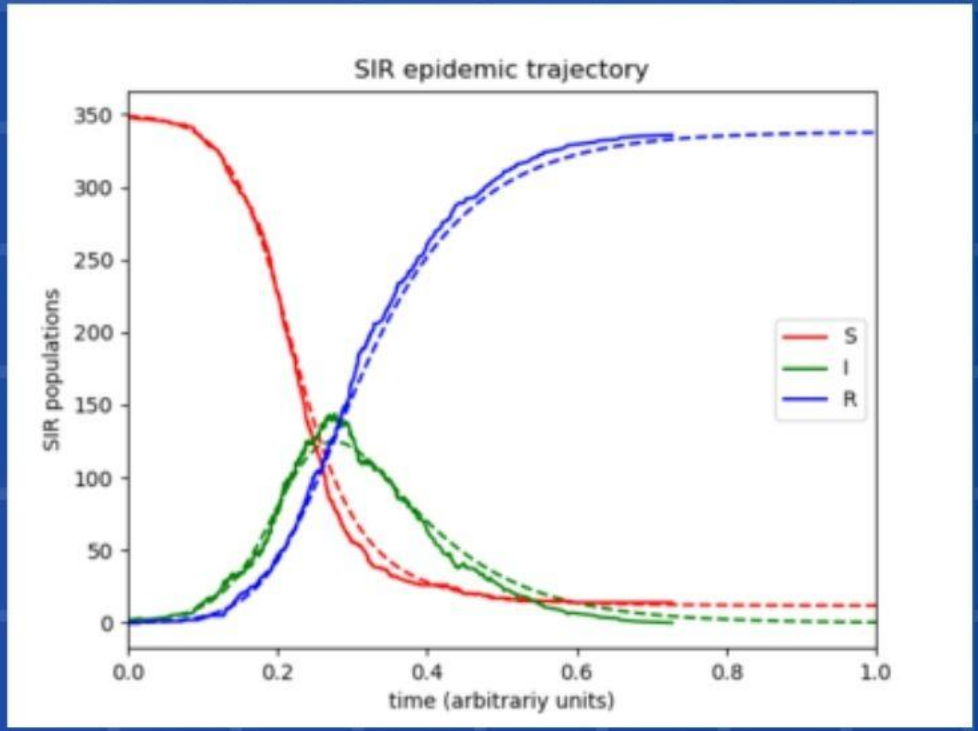
Даниил Бернулли

УИЛЬЯМ ФАРР

Модель SIR

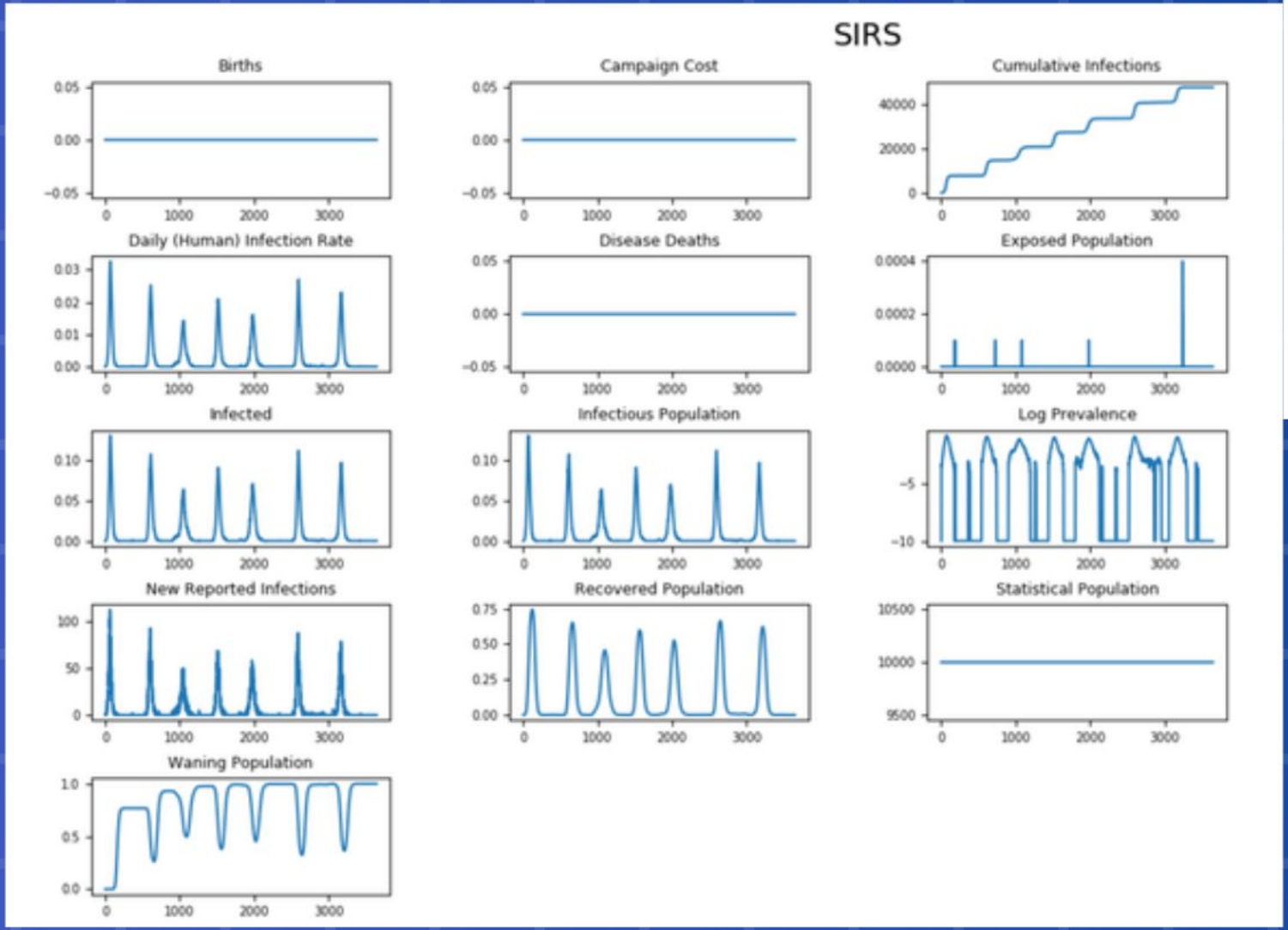


$$\frac{dS}{dt} = -\frac{\beta IS}{N},$$
$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I,$$
$$\frac{dR}{dt} = \gamma I.$$



SUSCEPTIBLE
INFECTED
RECOVERED

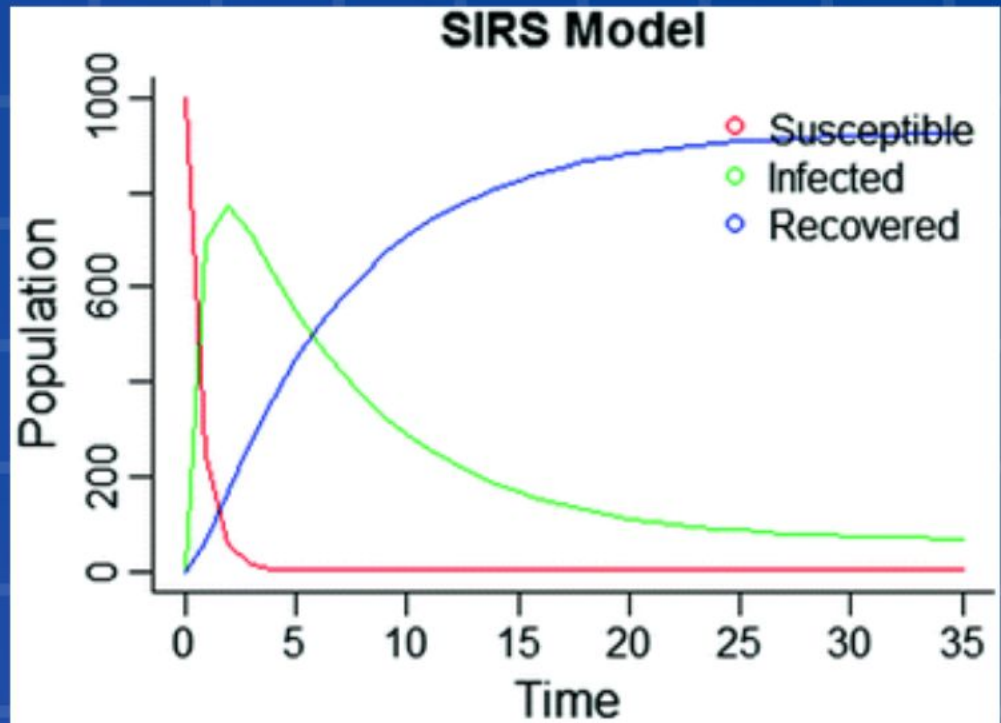
Модель SIRS



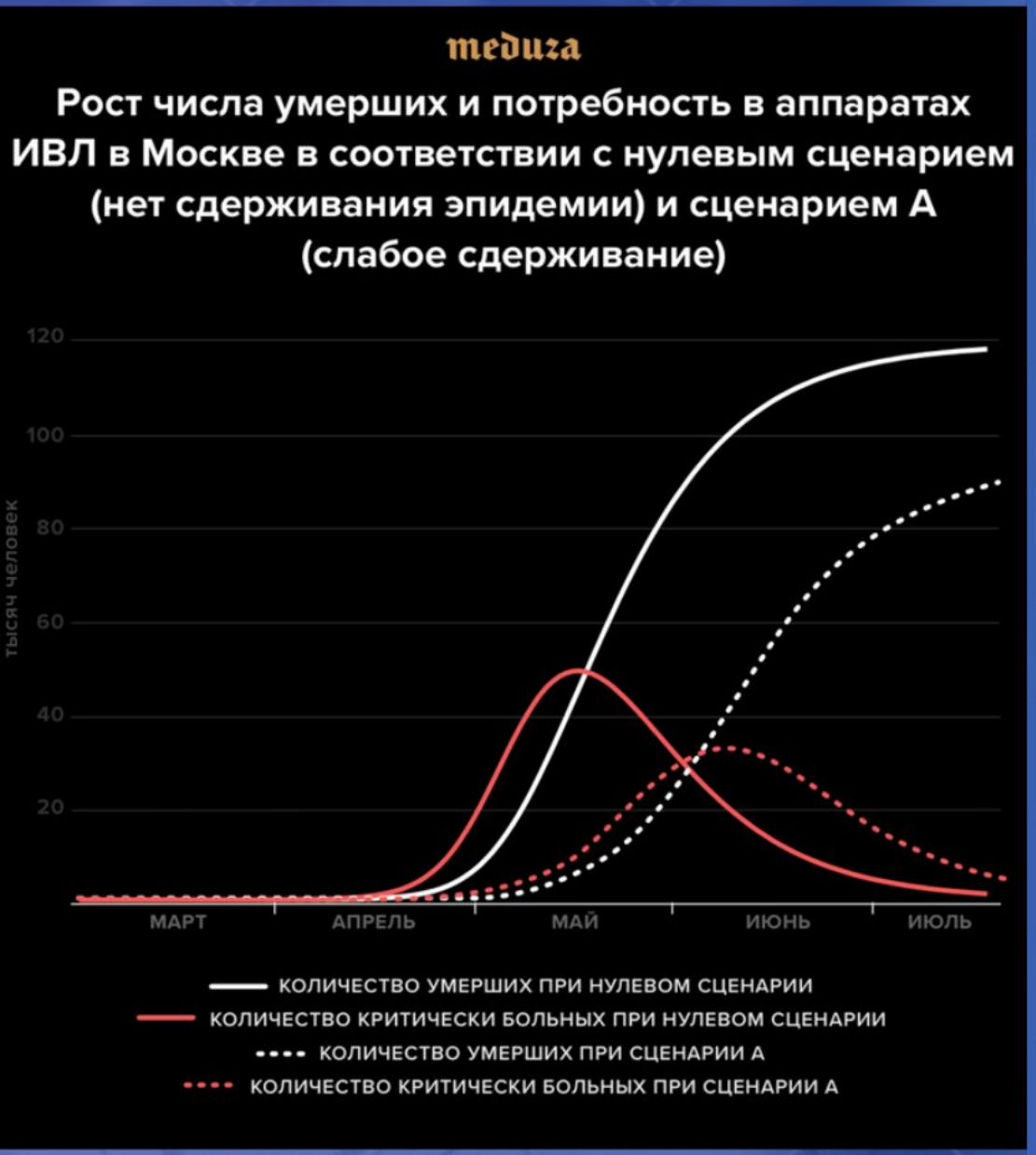
$$\frac{dS}{dt} = -\beta SI + \mu(N - S) + fR,$$

$$\frac{dI}{dt} = \beta SI - \gamma I - \mu I,$$

$$\frac{dR}{dt} = \gamma I - \mu R - fR,$$



Модель SEIR

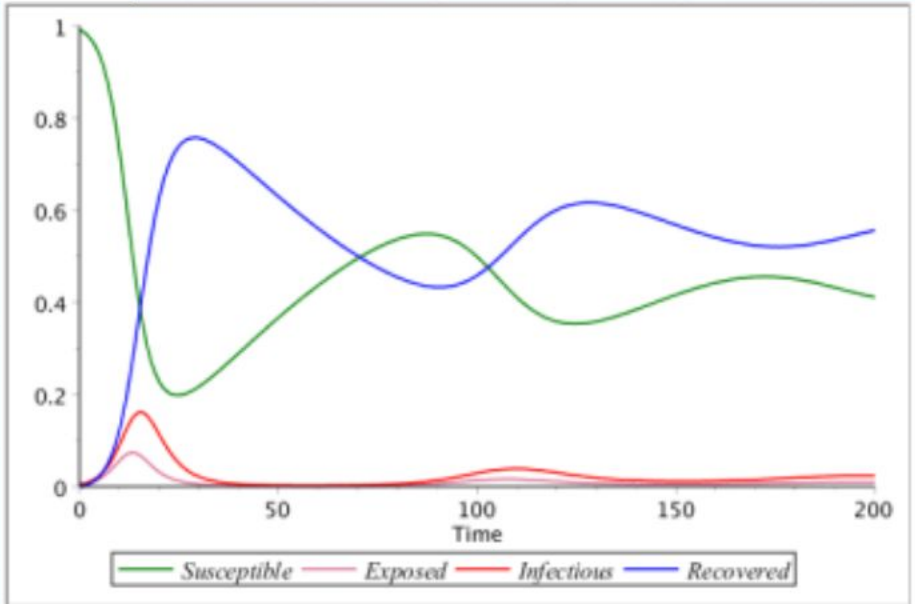


$$\frac{dS}{dt} = \mu N - \mu S - \beta \frac{I}{N} S$$

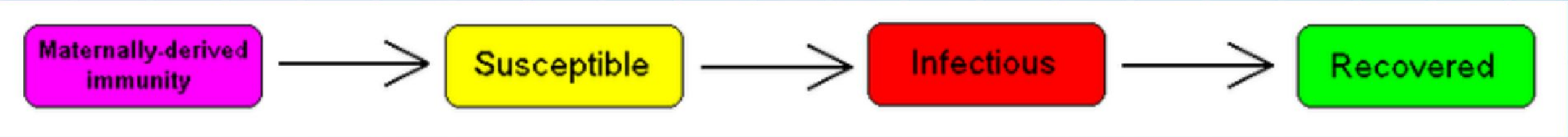
$$\frac{dE}{dt} = \beta \frac{I}{N} S - (\mu + a) E$$

$$\frac{dI}{dt} = aE - (\gamma + \mu) I$$

$$\frac{dR}{dt} = \gamma I - \mu R.$$



Также существует модель MSEIR



$$\frac{dM}{dT} = \Lambda - \delta M - \mu M$$

$$\frac{dS}{dT} = \delta M - \frac{\beta SI}{N} - \mu S$$

$$\frac{dI}{dT} = \frac{\beta SI}{N} - \gamma I - \mu I$$

$$\frac{dR}{dT} = \gamma I - \mu R$$

Применение в жизни

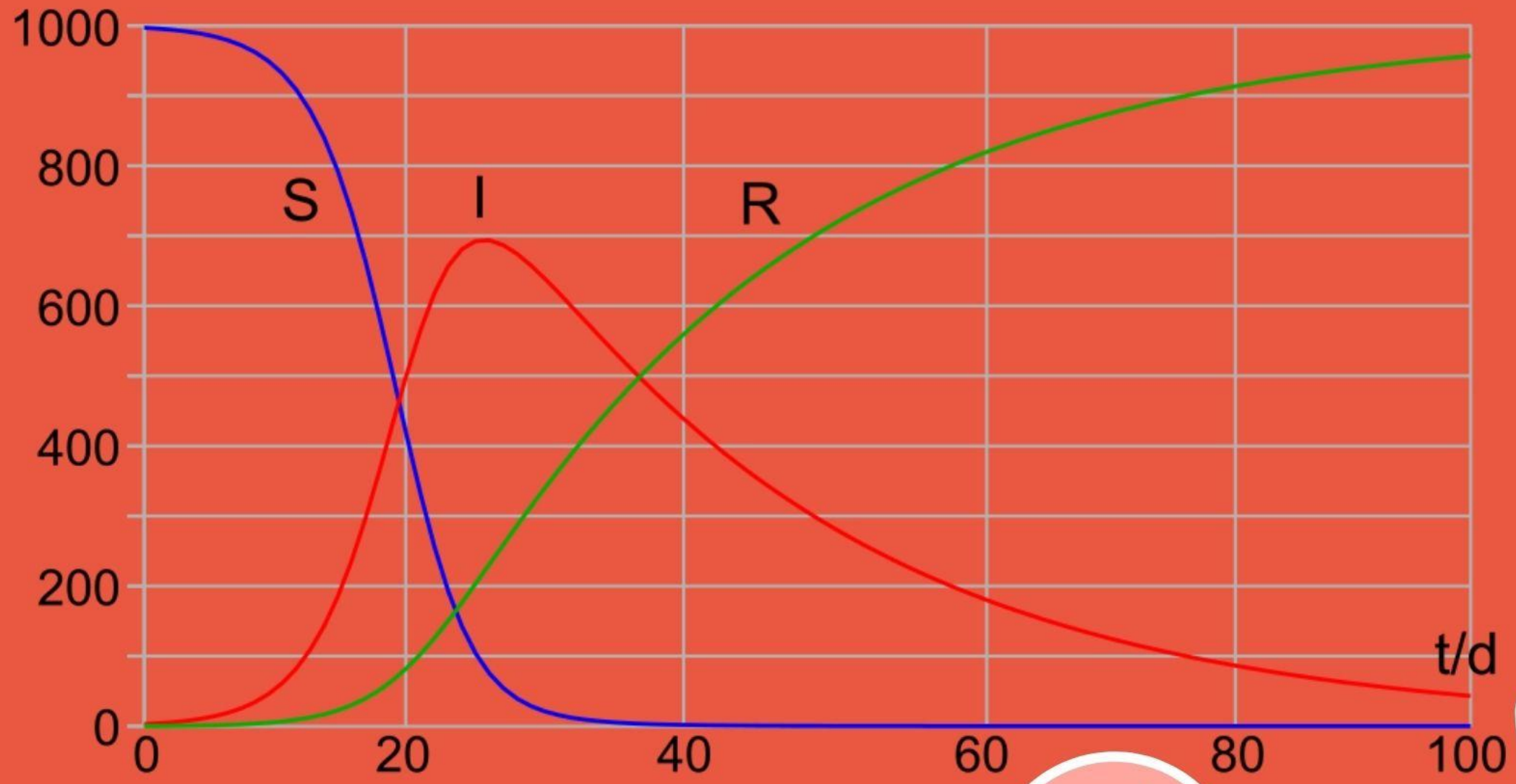
Исследование системы SIR аналитическим методом

$$\begin{cases} S' = -\dfrac{\beta}{N} IS, \\ I' = \dfrac{\beta}{N} IS - \gamma I, \\ R' = \gamma I \end{cases}$$

- $S(t)$ – общее количество восприимчивых (*susceptible*),
- $I(t)$ – общее количество инфицированных (*infectious*),
- $R(t)$ – общее количество выздоровевших (*recovered* or *removed*),
- N – общая численность популяции

$$S(0) = S_0$$
$$I(0) = I_0$$
$$R(0) = 0$$

- β – *интенсивность заражения*,
- γ – *интенсивность выздоравливания*


1000
800
600
400
200
0
S
I
R
t/d
0
20
40
60
80
100

Исследование SIR с помощью языка программирования Python

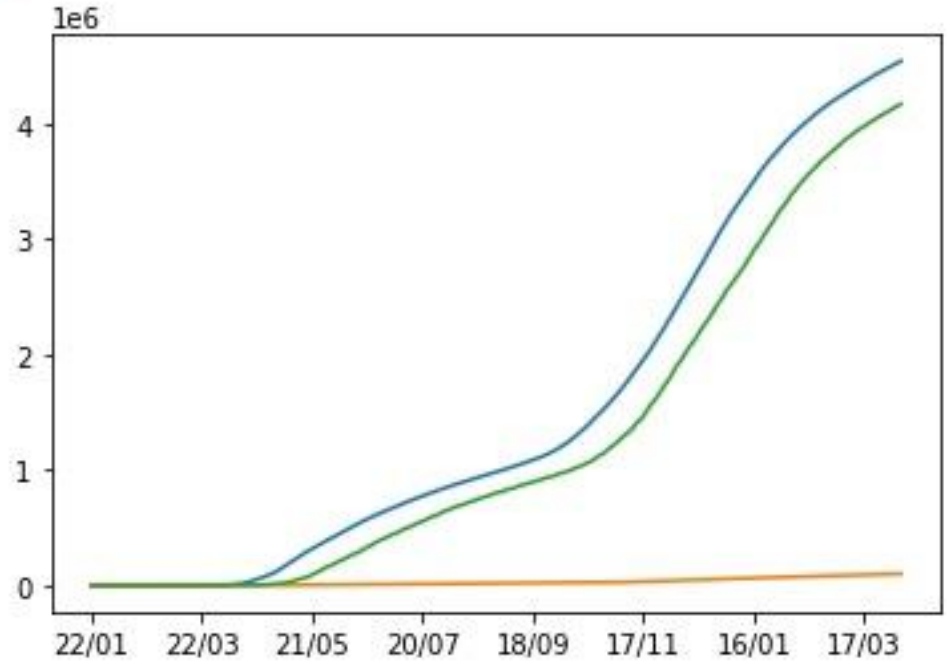
	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/28/21	3/29/21	3/30/21	3/31/21	4/1/21	4/2/21	4/3/21	4/4/21	4/5/21	4/6/21
0	NaN	Afghanistan	33.939110	67.709953	0	0	0	0	0	0	...	56294	56322	56384	56454	56517	56572	56595	56676	56717	56779
1	NaN	Albania	41.153300	20.168300	0	0	0	0	0	0	...	124134	124419	124723	125157	125506	125842	126183	126531	126795	126936
2	NaN	Algeria	28.033900	1.659600	0	0	0	0	0	0	...	116836	116946	117061	117192	117304	117429	117524	117622	117739	117879
3	NaN	Andorra	42.506300	1.521800	0	0	0	0	0	0	...	11850	11888	11944	12010	12053	12115	12174	12231	12286	12328
4	NaN	Angola	-11.202700	17.873900	0	0	0	0	0	0	...	22063	22132	22182	22311	22399	22467	22579	22631	22717	22885
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
269	NaN	Vietnam	14.058324	108.277199	0	2	2	2	2	2	...	2591	2594	2594	2603	2617	2620	2626	2631	2637	2648
270	NaN	West Bank and Gaza	31.952200	35.233200	0	0	0	0	0	0	...	236462	238248	240065	242353	244645	246893	248482	251288	253922	256461
271	NaN	Yemen	15.552727	48.516388	0	0	0	0	0	0	...	4033	4115	4247	4357	4531	4620	4697	4798	4881	4975
272	NaN	Zambia	-13.133897	27.849332	0	0	0	0	0	0	...	87872	88012	88199	88418	88549	88730	88800	88930	89009	89071
273	NaN	Zimbabwe	-19.015438	29.154857	0	0	0	0	0	0	...	36822	36839	36839	36882	36896	36903	36911	36923	36934	36966

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/28/21	3/29/21	3/30/21	3/31/21	4/1/21	4/2/21	4/3/21	4/4/21	4/5/21	4/6/21
213	NaN	Russia	61.52401	105.318756	0	0	0	0	0	0	...	4469327	4477916	4486078	4494234	4503291	4511973	4520879	4529576	4538101	4546307

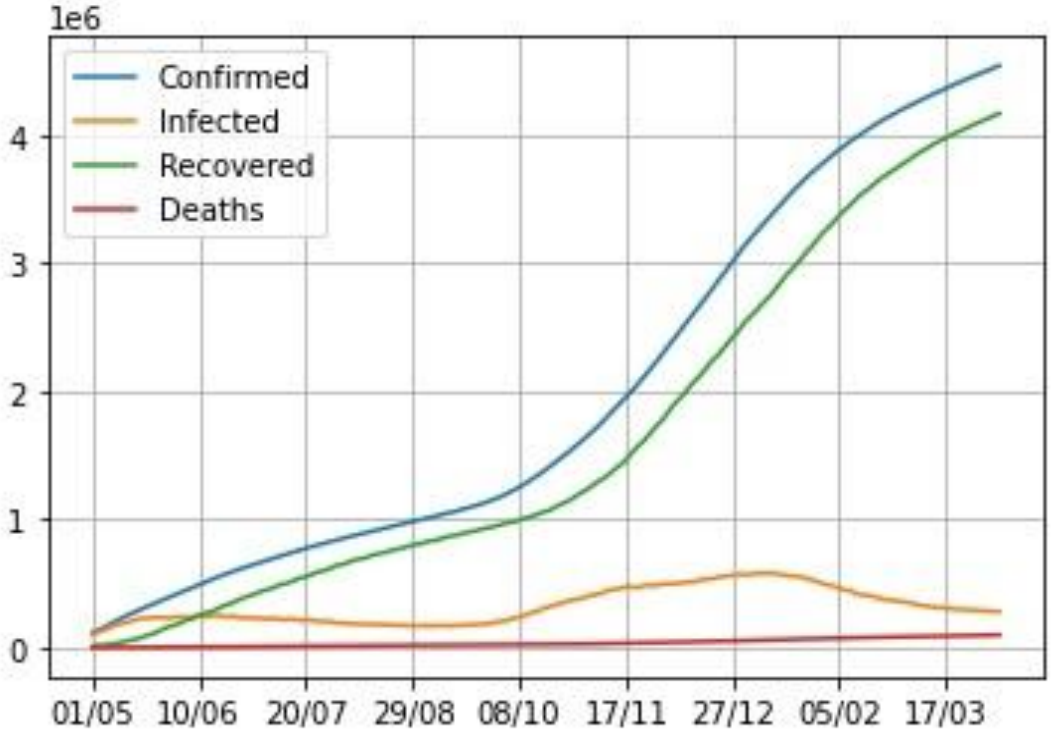
1 rows × 445 columns

```
array(['22/01', '23/01', '24/01', '25/01', '26/01', '27/01', '28/01',
       '29/01', '30/01', '31/01', '01/02', '02/02', '03/02', '04/02',
       '05/02', '06/02', '07/02', '08/02', '09/02', '10/02', '11/02',
       '12/02', '13/02', '14/02', '15/02', '16/02', '17/02', '18/02',
       '19/02', '20/02', '21/02', '22/02', '23/02', '24/02', '25/02',
       '26/02', '27/02', '28/02', '29/02', '01/03', '02/03', '03/03',
       '04/03', '05/03', '06/03', '07/03', '08/03', '09/03', '10/03',
       '11/03', '12/03', '13/03', '14/03', '15/03', '16/03', '17/03',
       '18/03', '19/03', '20/03', '21/03', '22/03', '23/03', '24/03',
       '25/03', '26/03', '27/03', '28/03', '29/03', '30/03', '31/03',
       '01/04', '02/04', '03/04', '04/04', '05/04', '06/04', '07/04',
       '08/04', '09/04', '10/04', '11/04', '12/04', '13/04', '14/04',
       '15/04', '16/04', '17/04', '18/04', '19/04', '20/04', '21/04',
       '22/04', '23/04', '24/04', '25/04', '26/04', '27/04', '28/04',
       '29/04', '30/04', '01/05', '02/05', '03/05', '04/05', '05/05',
       '06/05', '07/05', '08/05', '09/05', '10/05', '11/05', '12/05',
       '13/05', '14/05', '15/05', '16/05', '17/05', '18/05', '19/05',
       '20/05', '21/05', '22/05', '23/05', '24/05', '25/05', '26/05',
       '27/05', '28/05', '29/05', '30/05', '31/05', '01/06', '02/06',
       '03/06', '04/06', '05/06', '06/06', '07/06', '08/06', '09/06',
       '10/06', '11/06', '12/06', '13/06', '14/06', '15/06', '16/06',
       '17/06', '18/06', '19/06', '20/06', '21/06', '22/06', '23/06',
       '24/06', '25/06', '26/06', '27/06', '28/06', '29/06', '30/06',
       '01/07', '02/07', '03/07', '04/07', '05/07', '06/07', '07/07',
       '08/07', '09/07', '10/07', '11/07', '12/07', '13/07', '14/07',
       '15/07', '16/07', '17/07', '18/07', '19/07', '20/07', '21/07',
       '22/07', '23/07', '24/07', '25/07', '26/07', '27/07', '28/07',
       '29/07', '30/07', '31/07', '01/08', '02/08', '03/08', '04/08',
       '05/08', '06/08', '07/08', '08/08', '09/08', '10/08', '11/08',
       '12/08', '13/08', '14/08', '15/08', '16/08', '17/08', '18/08',
       '19/08', '20/08', '21/08', '22/08', '23/08', '24/08', '25/08',
       '26/08', '27/08', '28/08', '29/08', '30/08', '31/08', '01/09',
       '02/09', '03/09', '04/09', '05/09', '06/09', '07/09', '08/09',
       '09/09', '10/09', '11/09', '12/09', '13/09', '14/09', '15/09',
       '16/09', '17/09', '18/09', '19/09', '20/09', '21/09', '22/09',
       '23/09', '24/09', '25/09', '26/09', '27/09', '28/09', '29/09',
       '30/09', '01/10', '02/10', '03/10', '04/10', '05/10', '06/10',
       '07/10', '08/10', '09/10', '10/10', '11/10', '12/10', '13/10',
       '14/10', '15/10', '16/10', '17/10', '18/10', '19/10', '20/10',
       '21/10', '22/10', '23/10', '24/10', '25/10', '26/10', '27/10',
       '28/10', '29/10', '30/10', '31/10', '01/11', '02/11', '03/11',
       '04/11', '05/11', '06/11', '07/11', '08/11', '09/11', '10/11',
       '11/11', '12/11', '13/11', '14/11', '15/11', '16/11', '17/11',
       '18/11', '19/11', '20/11', '21/11', '22/11', '23/11', '24/11',
       '25/11', '26/11', '27/11', '28/11', '29/11', '30/11', '01/12',
       '02/12', '03/12', '04/12', '05/12', '06/12', '07/12', '08/12',
       '09/12', '10/12', '11/12', '12/12', '13/12', '14/12', '15/12',
       '16/12', '17/12', '18/12', '19/12', '20/12', '21/12', '22/12',
       '23/12', '24/12', '25/12', '26/12', '27/12', '28/12', '29/12',
       '30/12', '31/12', '01/01', '02/01', '03/01', '04/01', '05/01',
       '06/01', '07/01', '08/01', '09/01', '10/01', '11/01', '12/01',
       '13/01', '14/01', '15/01', '16/01', '17/01', '18/01', '19/01',
       '20/01', '21/01', '22/01', '23/01', '24/01', '25/01', '26/01',
       '27/01', '28/01', '29/01', '30/01', '31/01', '01/02', '02/02',
       '03/02', '04/02', '05/02', '06/02', '07/02', '08/02', '09/02',
       '10/02', '11/02', '12/02', '13/02', '14/02', '15/02', '16/02',
       '17/02', '18/02', '19/02', '20/02', '21/02', '22/02', '23/02',
       '24/02', '25/02', '26/02', '27/02', '28/02', '01/03', '02/03',
       '03/03', '04/03', '05/03', '06/03', '07/03', '08/03', '09/03',
       '10/03', '11/03', '12/03', '13/03', '14/03', '15/03', '16/03',
       '17/03', '18/03', '19/03', '20/03', '21/03', '22/03', '23/03',
       '24/03', '25/03', '26/03', '27/03', '28/03', '29/03', '30/03',
       '31/03', '01/04', '02/04', '03/04', '04/04', '05/04', '06/04'],
      dtype='<U5')
```

```
T = len(Russia_confirmed)
t = np.arange(T)
plt.plot(Russia_confirmed,)
plt.plot(Russia_deaths,)
plt.plot(Russia_recovered,)
plt.xticks(t[::60],dates[::60])
pass
```



```
T = len(C)
t = np.arange(T)
plt.plot(C, label='Confirmed')
plt.plot(I, label='Infected')
plt.plot(R, label='Recovered')
plt.plot(D, label='Deaths')
plt.legend()
plt.grid()
plt.xticks(t[::40],dates[::40])
pass
```

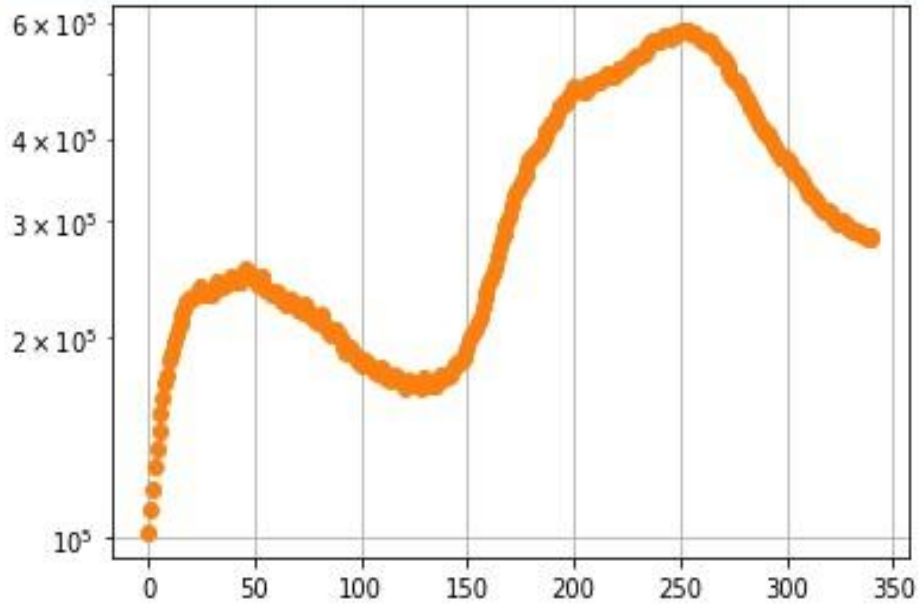


```python
def exponentian_growth(t, I0, alpha):
    return I0*np.exp(alpha*t)
```

```python
res, _ =optimize.curve_fit(exponentian_growth, t, I)
I0, alpha = res
I0, alpha
```

```
(-4.196950660891156e-14, 1.000000000781605)
```

```python
tt = np.array([0, T - 1])
II = I0*np.exp(alpha*tt)
plt.semilogy(tt, II)
plt.semilogy(t, I, 'o')
plt.grid()
pass
```



```
In [24]: np.exp(alpha) - 1
Out[24]: 0.22631035702992786

In [25]: gamma = 1/8

In [27]: R0 = 1 + alpha/gamma
         R0
Out[27]: 2.6320796120451524

In [ ]:
```