Построить SIR-модель развития эпидемии за весь период наблюдений по датасету для выбранной страны. Для этого:

1. Вычислить количество инфицированных I на каждый день наблюдений
2. Выполнить нормировку данных на 100 тыс. населения.
3. На основе данных построить оценку параметра γ – интенсивности выздоровления.
4. Построить оценку параметра SIR-модели β (интенсивность заражения) в предположении, что изначально все население является восприимчивым к заболеванию. Обратить внимание, что значение β может быть разным для разных временных интервалов.
5. Построить график зависимости среднего количества инфицированных от времени. Для сравнения на той же диаграмме построить график для реальных (нормированных) данных.

Справочно о SIR-модели.

Модель эпидемии SIR была предложена в статье Kermack W., McKendrick A. «A contribution to the mathematical theory of epidemics». Proc.R. Soc. 1927. V. A115. P. 700–721.

Данная модель основана на разделении популяции S (susceptible) — восприимчивые, I (infectious) — больные, R (recovered) — выздоровевшие (невосприимчивые), $S + I + R = N$ – численность популяции.

Динамика переходов индивидуумов их одной группы в другую S→ I→ R описывается системой дифференциальных уравнений:

$$\begin{cases} \frac{dI}{dt} = \beta \frac{SI}{N} - \gamma I \\ \frac{dS}{dt} = -\beta \frac{SI}{N} \\ \frac{dR}{dt} = \gamma I \end{cases},$$

где β – интенсивность заражения, т.е. среднее число индивидуумов, которых может заразить один больной в единицу времени, γ – интенсивность выздоровления, величина, обратная средней продолжительности заболевания.

Поскольку в датасетах представлены ежедневные наблюдения, перейдем от дифференциальных уравнений к разностным:

dt=1 день, $n = t$ – номер дня. Тогда

$dI = I(t + dt) - I(t) = I_{n+1} - I_n = \beta \frac{S_n I_n}{N} - \gamma I_n$, аналогично для остальных уравнений.

$$\begin{cases} I_{n+1} - I_n = \beta \frac{S_n I_n}{N} - \gamma I_n \\ S_{n+1} - S_n = -\beta \frac{S_n I_n}{N} \\ R_{n+1} - R_n = \gamma I_n \end{cases}$$ начальные условия I_0, S_0, R_0

Или $$\begin{cases} I_{n+1} = I_n + \beta \frac{S_n I_n}{N} - \gamma I_n \\ S_{n+1} = S_n - \beta \frac{S_n I_n}{N} \\ R_{n+1} = R_n + \gamma I_n \end{cases}$$

$\nu_n = \beta \frac{S_n I_n}{N} -$ расчетное число заражений за n-й день

Выполнение задания.

Загружаем данные:

```
import pandas as pd
import numpy as np
df=pd.read_excel('Россия1.xlsx')
df
```

	Страна	Дата	Заражений	Заражений_за_день	Выздоровлений	Выздоровлений_за_день	Смертей	Смертей_за_день	Население_страны	Тестов	Тестов_за_день
0	Россия	11.04.2020	13584	1667	1045	250	106	12	146880432	1359993	81246
1	Россия	12.04.2020	15770	2186	1291	246	130	24	146880432	1426014	66021
2	Россия	13.04.2020	18328	2558	1470	179	148	18	146880432	1517992	91978
3	Россия	14.04.2020	21102	2774	1694	224	170	22	146880432	1613413	95421
4	Россия	15.04.2020	24490	3388	1986	292	198	28	146880432	1718019	104606
...	...	...	...	...	...	...	...	...	...	...	...
95	Россия	15.07.2020	745197	6410	522375	10417	11753	156	146880432	24364568	311052
96	Россия	16.07.2020	751612	6415	530801	8426	11920	167	146880432	24676930	312362
97	Россия	17.07.2020	758001	6389	538467	7666	12106	186	146880432	24991740	314810
98	Россия	18.07.2020	764215	6214	545909	7442	12228	122	146880432	25251614	259874
99	Россия	19.07.2020	770311	6096	549387	3478	12323	95	146880432	25449167	197553

100 rows × 11 columns

Убираем столбцы, которые не понадобятся для последующих вычислений:

```
[43] df.drop(['Страна'],axis=1,inplace=True)
     df.drop(['Тестов'],axis=1,inplace=True)
     df.drop(['Тестов_за_день'],axis=1,inplace=True)
     df.drop(['Смертей_за_день'],axis=1,inplace=True)
     df
```

	Дата	Заражений	Заражений_за_день	Выздоровлений	Выздоровлений_за_день	Смертей	Население_страны
0	11.04.2020	13584	1667	1045	250	106	146880432
1	12.04.2020	15770	2186	1291	246	130	146880432
2	13.04.2020	18328	2558	1470	179	148	146880432
3	14.04.2020	21102	2774	1694	224	170	146880432
4	15.04.2020	24490	3388	1986	292	198	146880432
...	...	...	...	...	...	...	...
95	15.07.2020	745197	6410	522375	10417	11753	146880432
96	16.07.2020	751612	6415	530801	8426	11920	146880432
97	17.07.2020	758001	6389	538467	7666	12106	146880432
98	18.07.2020	764215	6214	545909	7442	12228	146880432
99	19.07.2020	770311	6096	549387	3478	12323	146880432

100 rows × 7 columns

Вычислим количество инфицированных I и количество восприимчивых S:

```
[44] df['Инфицированные_I']=df['Заражений']-df['Выздоровлений']-df['Смертей']
     df['Восприимчивые_S']=df['Население_страны']-df['Заражений']
     df
```

	Дата	Заражений	Заражений_за_день	Выздоровлений	Выздоровлений_за_день	Смертей	Население_страны	Инфицированные_I	Восприимчивые_S
0	11.04.2020	13584	1667	1045	250	106	146880432	12433	146866848
1	12.04.2020	15770	2186	1291	246	130	146880432	14349	146864662
2	13.04.2020	18328	2558	1470	179	148	146880432	16710	146862104
3	14.04.2020	21102	2774	1694	224	170	146880432	19238	146859330
4	15.04.2020	24490	3388	1986	292	198	146880432	22306	146855942
...	...	...	...	...	...	...	...	...	...
95	15.07.2020	745197	6410	522375	10417	11753	146880432	211069	146135235
96	16.07.2020	751612	6415	530801	8426	11920	146880432	208891	146128820
97	17.07.2020	758001	6389	538467	7666	12106	146880432	207428	146122431
98	18.07.2020	764215	6214	545909	7442	12228	146880432	206078	146116217
99	19.07.2020	770311	6096	549387	3478	12323	146880432	208601	146110121

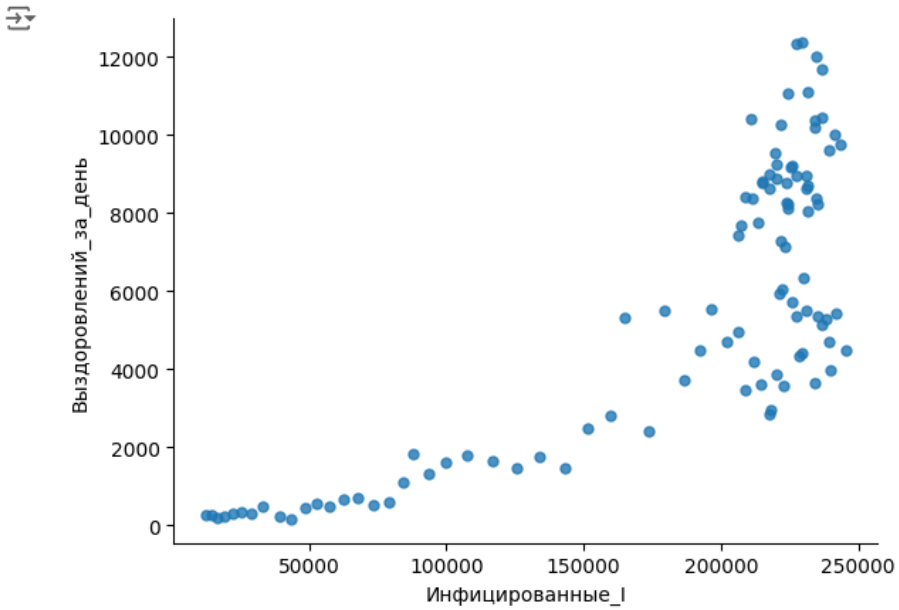
100 rows × 9 columns

<u>Оценка параметра γ с помощью метода наименьших квадратов</u>

Кол-во_инфицированных_I vs Выздоровлений_за_день

```
[45] # @title Кол-во_инфицированных_I vs Выздоровлений_за_день

     from matplotlib import pyplot as plt
     df.plot(kind='scatter', x='Инфицированные_I', y='Выздоровлений_за_день', s=25, alpha=.8)
     plt.gca().spines[['top', 'right',]].set_visible(False)
```



r_i – наблюдаемое количество выздоровлений за i-й день, I_i- количество больных, наблюдаемое в i-й день

Тогда $r_i = \gamma I_i + \varepsilon_i$, ε_i – ошибка модели. γ необходимо выбрать таким образом, чтобы минимизировать квадрат ошибки, т.е.

$$\sum_{i=1}^{n} (r_i - \gamma I_i)^2 \to \min_{\gamma}$$

Для нахождения минимума продифференцируем целевую функцию и

приравняем к нулю (необходимое условие экстремума):

$$2\sum_{i=1}^{n}(r_i - \gamma I_i) * (-I_i)=0$$
$$\sum_{i=1}^{n}(r_i I_i) - \gamma \sum_{i=1}^{n} {I_i}^2 = 0$$

$\hat{\gamma} = \frac{\sum_{i=1}^{n}(r_i I_i)}{\sum_{i=1}^{n} {I_i}^2}$ – оценка для параметра γ

Вычислим оценку параметра γ.

```
[48] gam=sum(df['Выздоровлений_за_день']*df['Инфицированные_I'])/sum(df['Инфицированные_I']*df['Инфицированные_I'])
     print ('Параметр гамма', gam)
     print ('Средняя продолжительность заболевания', 1/gam)
```

```
Параметр гамма 0.03182500033051489
Средняя продолжительность заболевания 31.42183785120549
```

Теперь сделаем то же самое с помощью линейной регрессии.

```
[ ] import statsmodels.formula.api as smf
    lm = smf.ols(formula='Выздоровлений_за_день ~0 + Инфицированные_I', data=df).fit()
    lm.params
    print(lm.summary())
```

```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:     Выздоровлений_за_день   R-squared (uncentered):                   0.881
Model:                               OLS   Adj. R-squared (uncentered):              0.880
Method:                    Least Squares   F-statistic:                              731.9
Date:                   Tue, 12 Nov 2024   Prob (F-statistic):                    1.57e-47
Time:                           20:24:55   Log-Likelihood:                         -914.63
No. Observations:                    100   AIC:                                      1831.
Df Residuals:                         99   BIC:                                      1834.
Df Model:                              1
Covariance Type:               nonrobust
=======================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------
Инфицированные_I     0.0318      0.001     27.054      0.000       0.029       0.034
==============================================================================
Omnibus:                        6.930   Durbin-Watson:                   1.294
Prob(Omnibus):                  0.031   Jarque-Bera (JB):                4.621
Skew:                           0.375   Prob(JB):                       0.0992
Kurtosis:                       2.260   Cond. No.                         1.00
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Построим оценку параметра β.

```
[ ] lm = smf.ols(formula='Заражений_за_день ~ 0 + Расчёт', data=df).fit()
    lm.params
    print(lm.summary())
```

```
                                 OLS Regression Results
================================================================================
Dep. Variable:      Заражений_за_день   R-squared (uncentered):                   0.903
Model:                            OLS   Adj. R-squared (uncentered):              0.902
Method:                 Least Squares   F-statistic:                              922.0
Date:                Tue, 12 Nov 2024   Prob (F-statistic):                    5.79e-52
Time:                        20:27:00   Log-Likelihood:                         -922.14
No. Observations:                 100   AIC:                                      1846.
Df Residuals:                      99   BIC:                                      1849.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Расчёт         0.0386      0.001     30.364      0.000       0.036       0.041
==============================================================================
Omnibus:                       15.601   Durbin-Watson:                   0.044
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                9.614
Skew:                           0.608   Prob(JB):                      0.00817
Kurtosis:                       2.090   Cond. No.                         1.00
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Сравним с оценкой, полученной вручную с помощью метода наименьших квадратов.

Обозначим $A_i = \frac{S_i I_i}{N}, X_i$ – наблюдаемое количество заражений в день.

Тогда нам надо минимизировать суммарную квадратичную ошибку

$$\sum_{i=1}^{n} (X_i - \beta A_i)^2 \to \min_{\beta}$$

Для нахождения минимума продифференцируем целевую функцию и приравняем к нулю (необходимое условие экстремума):

$$2 \sum_{i=1}^{n} (X_i - \beta A_i) * (-A_i) = 0$$
$$\sum_{i=1}^{n} (X_i A_i) - \beta \sum_{i=1}^{n} {A_i}^2 = 0$$

$\hat{\beta} = \frac{\sum_{i=1}^{n} (X_i A_i)}{\sum_{i=1}^{n} {A_i}^2}$ – оценка для параметра β

```
[56] beta = sum(df['Заражений_за_день']*df['Расчёт'])/sum(df['Расчёт']*df['Расчёт'])
     print ('Параметр бета', beta)
```

```
Параметр бета 0.03863430714471173
```

Нормируем данные и строим график:

```
[60] df['Инфицированные_нормир']=df['Инфицированные_I'] * 100000 / df['Население_страны']
     df
```

Нормированные инфицированные

```
# @title Нормированные инфицированные

from matplotlib import pyplot as plt
df.plot(kind='scatter', x='Дата', y='Инфицированные_нормир', s=10, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```