SIR & SEIR MODEL CALIBRATION

DATA INTRO

Data : Influenza infected in Saint Petersburg

Period : 2003-11-15 – 2004-01-04

Count : 51

Library : Scipy & numpy & matplotlib

Function : scipy.optimize.curve_fit & scipy.optimize.minimize (BFGS)

Evaluation metrics : R-Squared (R2)

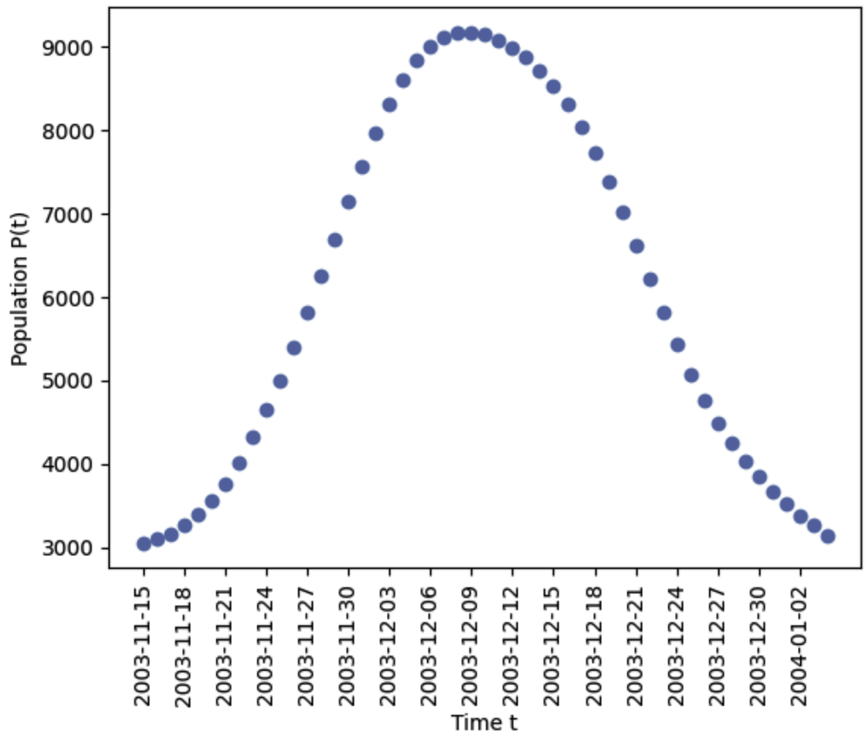

Figure 1 – provided datasets

SIR MODEL

S : susceptible   I : infected R : recovered

R2 : 0.9585

total infected : 317825

total recovered : 458725 (why is it more than recovered??)

Formula :

[46]:
```python
def sir_model(y, t, beta, gamma):
    dS = -beta*y[0]*y[1] /N
    dI =  beta*y[0]*y[1] /N - gamma*y[1]
    dR =  gamma*y[1]
    return [dS, dI, dR]

# Define the parameters
def fit_odeint(x, beta, gamma):
    return integrate.odeint(sir_model, (S0, I0, R0), x, args=(beta, gamma))[:,1]

# Initial values
N = 4681000 # Population of SPB
I0 = pd_datetime_infect['infect'].tolist()[0]
S0 = N - I0
R0 = 0.0
```

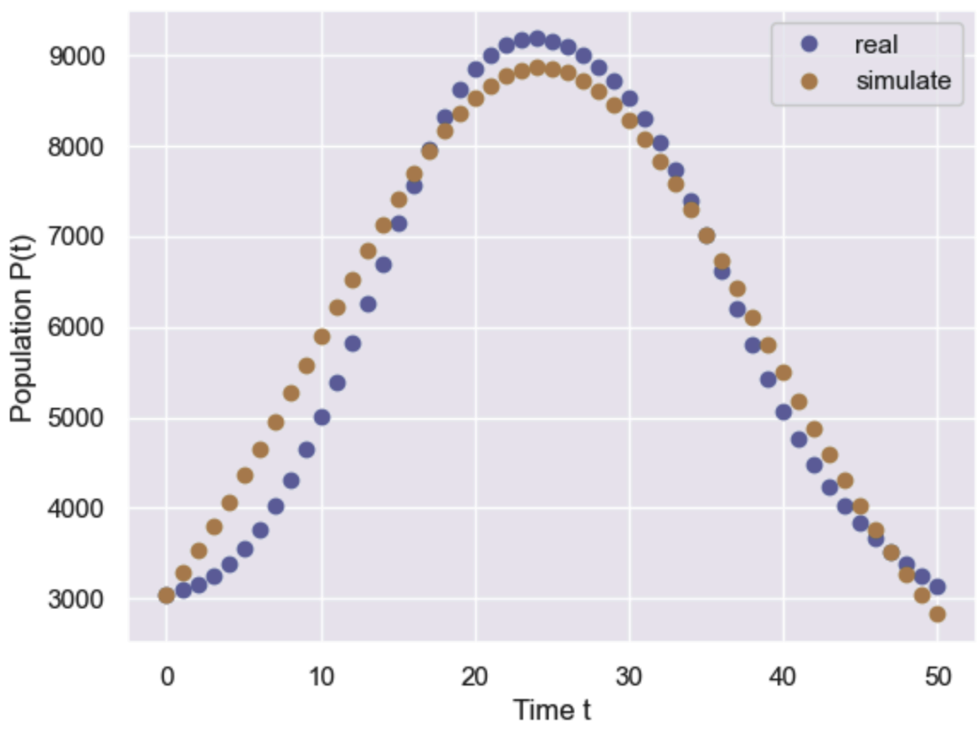

Figure 2 – Infected cases calibration by SIR

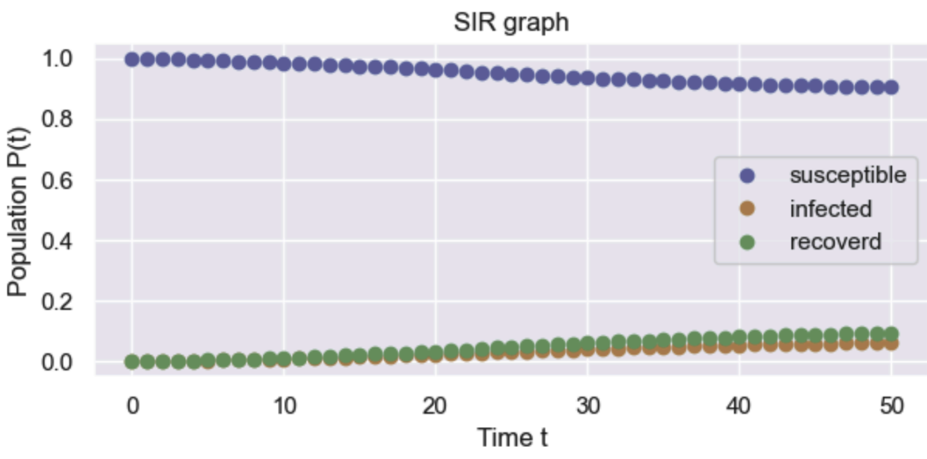

Figure 3 –SIR graph

SEIR MODEL

S : susceptible E : exposed I : infected R : recovered

For better calibration, initial exposed is set as infected * 1.1

#func1 scipy.optimize.curve_fit

R2 : 0.9882

total exposed : 492094

total infected : 309297

total recovered : 770624 (why more than infected ??)

Formula :

```
[38]: # S : susceptible , E : exposed , I : infected , R : recovered

def seir_model(y, t,beta ,gamma, sigma):
    # susceptible - infected
    dS = -(beta * y[2]) * y[0] / N
    # exposed - infected
    dE = (beta * y[2]) * y[0] / N  - gamma * y[1]
    # infected - recovered
    dI = gamma * y[1] - sigma * y[2]
    dR = sigma * y[2]
    return [dS, dE, dI, dR]

def fit_odeint(x,beta ,gamma, sigma):
    return integrate.odeint(seir_model, (S0, E0, I0, R0), x, args=(beta ,gamma, sigma))[:, 2]

# Initial values
N = 4681000 # Population of spb
I0 = pd_datetime_infect['infect'].tolist()[0]
E0 = I0 * 1.1
#E0 = 0
R0 = 0.0
S0 = N - I0 - E0 - R0
```

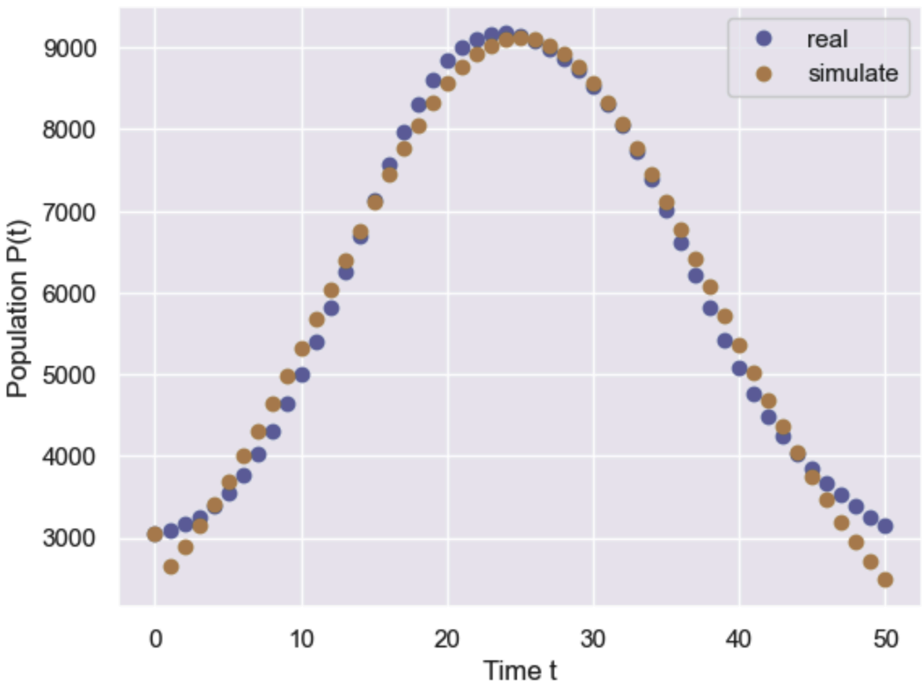

Figure 4 – Infected cases calibration by SEIR

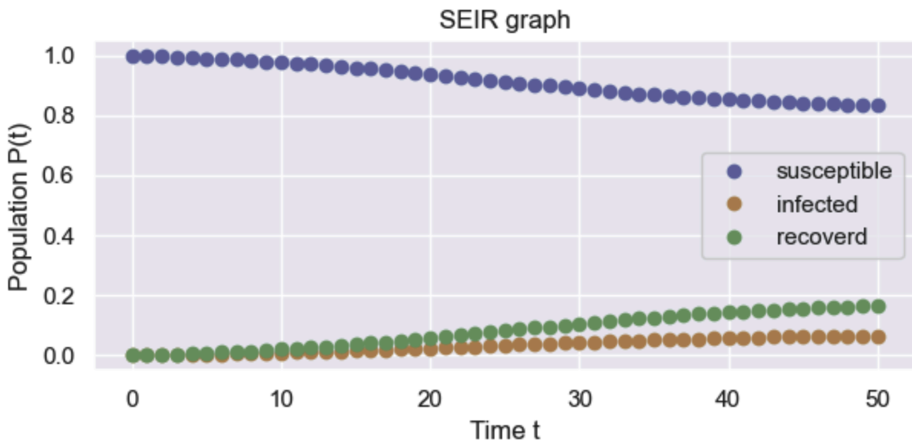

Figure 5 –SEIR graph

#func2 scipy.optimize.minimize (BFGS)

R2 : 0.9783

total exposed : 603465

total infected : 301521

total recovered : 855096 (why more than infected ??)

Formula :

initial value follow last formula

```
•[42]: # change to minus sign to get most minimizing value
def test(params):
    beta ,gamma, sigma = params
    return -r2_score(pd_datetime_infect['infect'],\
                     integrate.odeint(seir_model, (S0, E0, I0, R0), list(range(0,len(pd_datetime_infect['datetime']))),\
                                      args=(beta ,gamma, sigma))[:, 2])

from scipy.optimize import minimize
#bnds = ((0.095,0.75), (0.2,8), (0.08, 0.33))
res = minimize(test, [1,1,1], method='BFGS', bounds=None, tol=1e-6)
print(res.fun)
res.x
```

```
/var/folders/6m/1kyq0wq56bx12x382j8hdw1h0000gq/T/ipykernel_26084/1919019302.py:8: RuntimeWarning: Method BFGS cannot handle bounds.
  res = minimize(test, [1,1,1], method='BFGS', bounds=bnds, tol=1e-6)
-0.978398051170285
[42]: array([3.16604047, 1.42358352, 2.86294065])
```

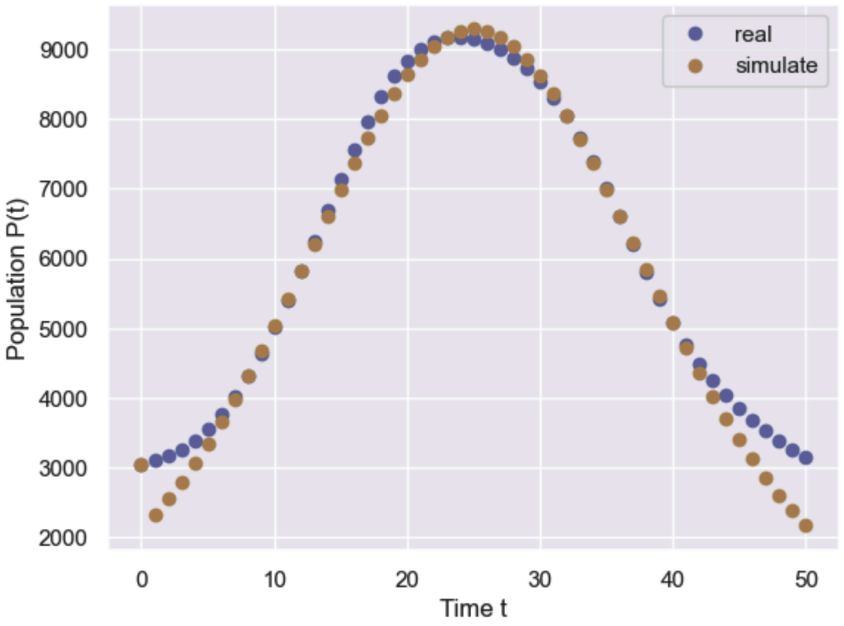

Figure 6 –SEIR graph

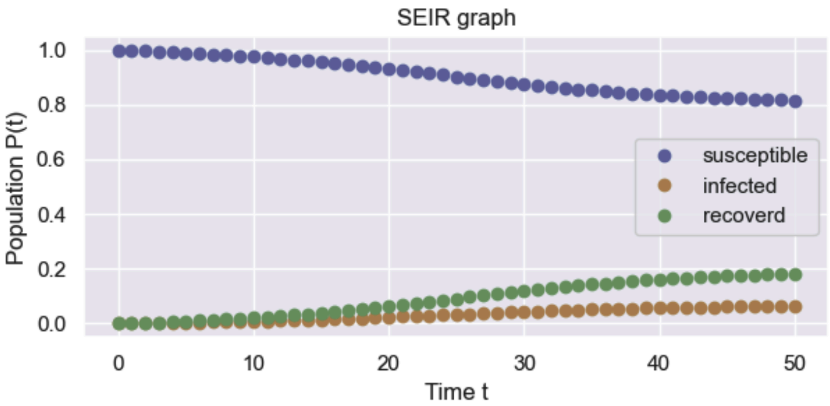

Figure 7 –SEIR graph

Conclusion :

Some easy demonstrations of SIR & SEIR models, from the result of R2, show slightly better performance than SIR. However there is doubt why recovered entities are more than infected entities.

Notebook could be found at :
https://github.com/MaChengYuan/epidemiology/blob/main/SPB_influenza_calibration.ipynb

REVIEW OF PAPER

From my understanding of this paper, experiments 1-3 serve the purpose for calibration of multiple curves derived from datasets of different years in 3 cities in Russia, which had shown the accessibility of narrowing down the searching interval of dependent variables without reducing the performance of the model drastically. Two argument Δ and k_{inc} may lead to 'undesirable freedom' but could help to fit improper model curves. Moreover they allow the estimation of pandemics start and level of non-epidemic.

My understanding for experiment 4 is to find dependent variables γ and σ from uniform distribution of restraint values used in experiment 2. But I could not understand the details. The results shows the performance of models are more likely based on α and β more than γ and σ.