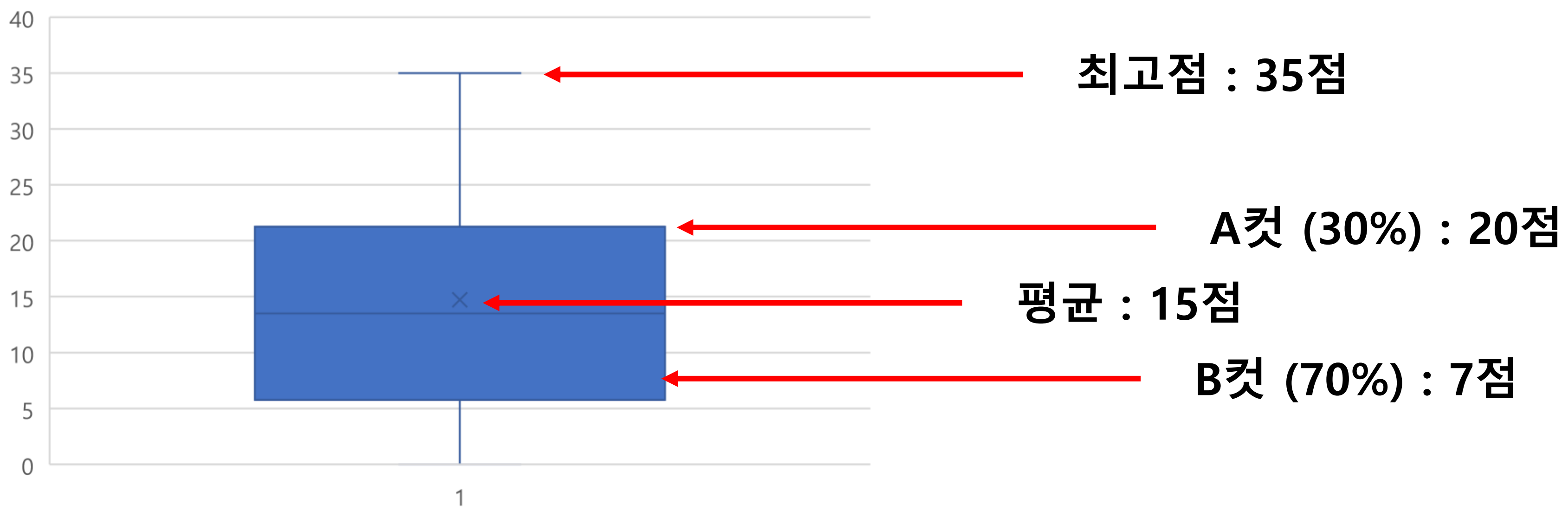
2024학년도 2학기

문제해결프로그래밍 강의 10주차

조기필

2024.11.14

중간고사 성적 분포



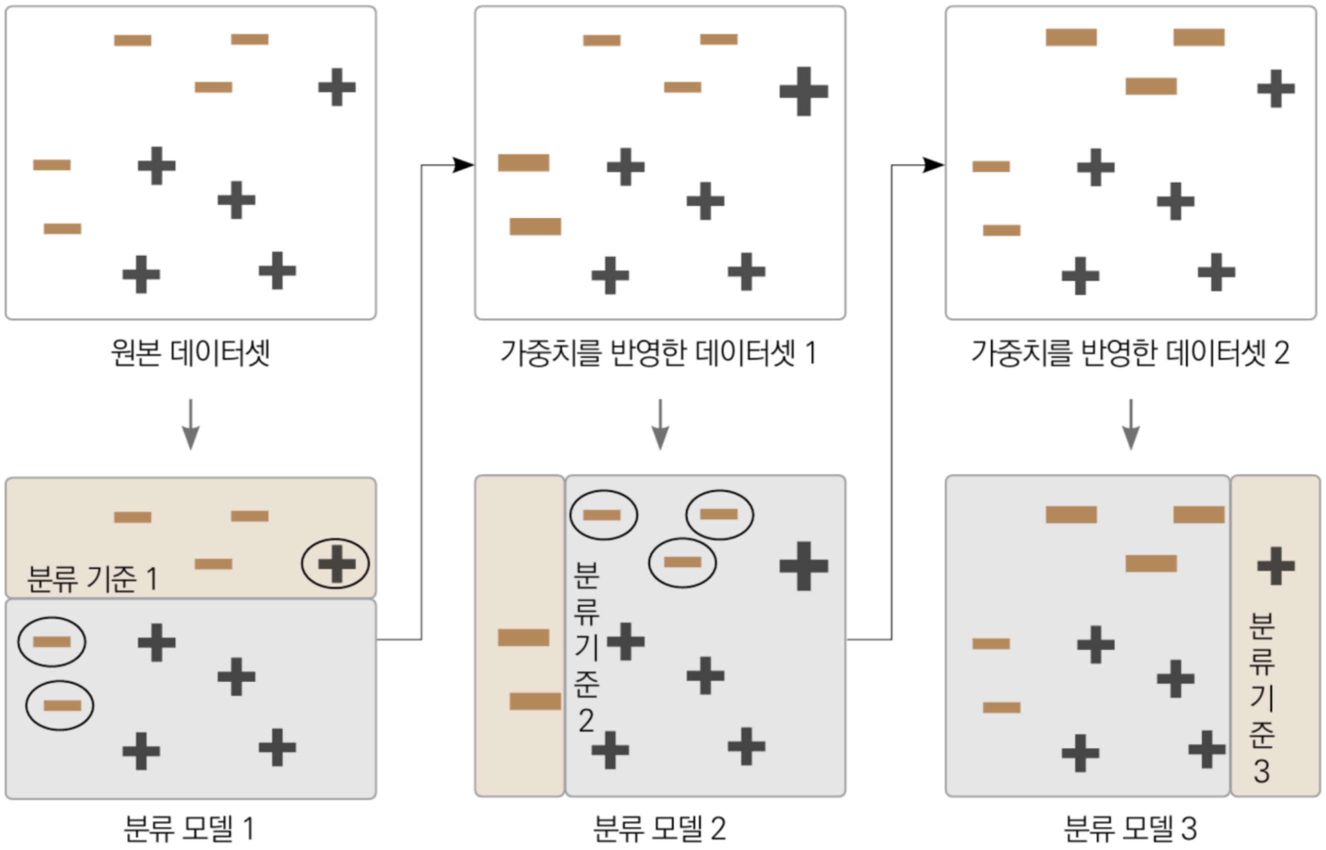
지난시간 복습

1. 안전 운전자 예측
2. 코로나-19 수학 모델 (모델 만들기)

문제 2 : 안전 운전자 예측

Boosting은 북돋우다는 의미

: 가중치를 활용해 분류 성능이 약한 모델을 강하게 만드는 기법



1 '+'와 '-'로 구성된 원본 데이터셋을 준비합니다.

2 처음에는 모든 데이터에 동일한 가중치를 줍니다.

3 분류 모델 1로 '+'와 '-'를 분류합니다.

4 분류 모델 1이 잘못 분류한 데이터(동그라미 표시된 데이터)에 더 높은 가중치를 부여합니다.

5 분류 모델 2는 가중치가 부여된 데이터에 집중해 데이터를 분류합니다. 다른 데이터보다 가중치가 부여된 데이터를 더 제대로 구분하려고 노력한다는 말입니다.

6 분류 모델 2가 잘못 분류한 데이터에 더욱 높은 가중치를 부여합니다.

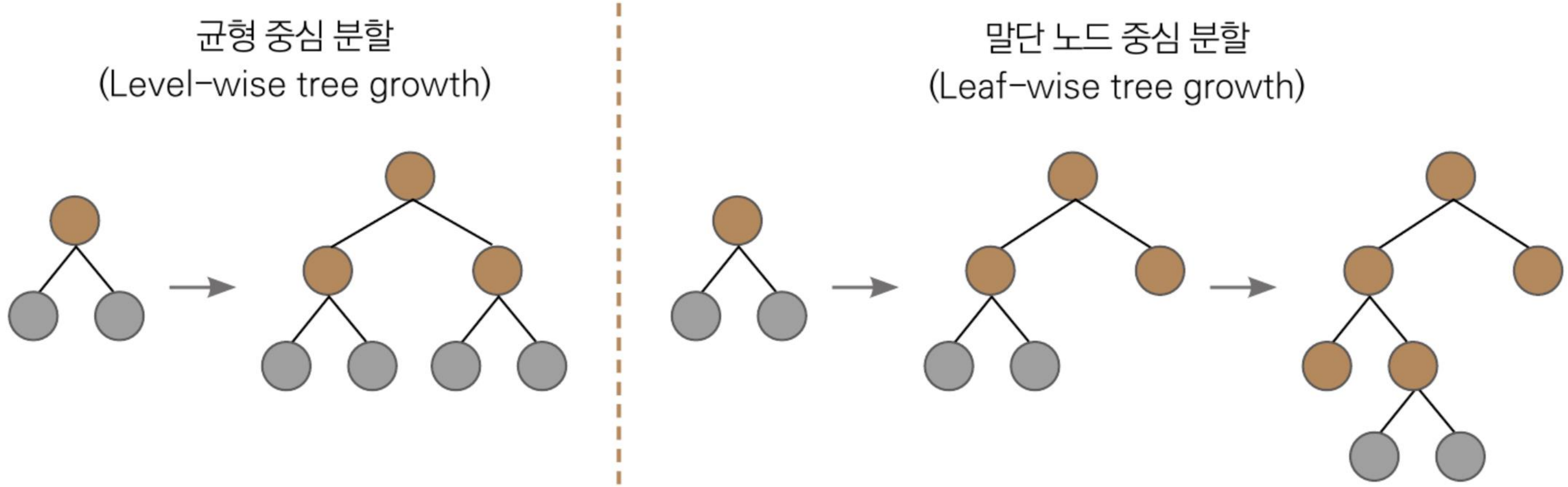
7 분류 모델 3은 이전 단계에서 가중치가 부여된 데이터에 집중해 데이터를 분류합니다.

문제 2 : 안전 운전자 예측

XGBoost (Extreme Gradient Boosting)과 LightGBM

: 성능이 우수한 트리 기반 부스팅 알고리즘

랜덤포레스트의 병렬 배치와 다르게 직렬 배치



XGBoost

LightGBM
(속도가 빠르지만, 데이터 개수가 적을때는 과대적합되기 쉽다.)

문제 2 : 안전 운전자 예측

LightGBM 모델을 적용해보자.

```
from lightgbm import LGBMClassifier
```

LightGBM 라이브러리

```
LGBM_model = LGBMClassifier(force_col_wise=True)
```

LightGBM에서 메모리 부족 문제를 해결하기 위한 옵션, 행단위에서 열단위로 메모리 할당

```
LGBM_model.fit(X_train, y_train)
```

```
[LightGBM] [Info] Number of positive: 17355, number of negative: 458814
[LightGBM] [Info] Total Bins 1096
[LightGBM] [Info] Number of data points in the train set: 476169, number of used
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.036447 -> initscore=-3.274764
[LightGBM] [Info] Start training from score -3.274764
```

```
▾ LGBMClassifier
LGBMClassifier(force_col_wise=True)
```

양성(positive) 클래스의 평균 확률값과 초기 손실값

문제 2 : 안전 운전자 예측

```
pred_train = LGBM_model.predict(X_train)
pred_test = LGBM_model.predict(X_test)
```

오차행렬 (Confusion Matrix)

```
from sklearn.metrics import confusion_matrix, classification_report
```

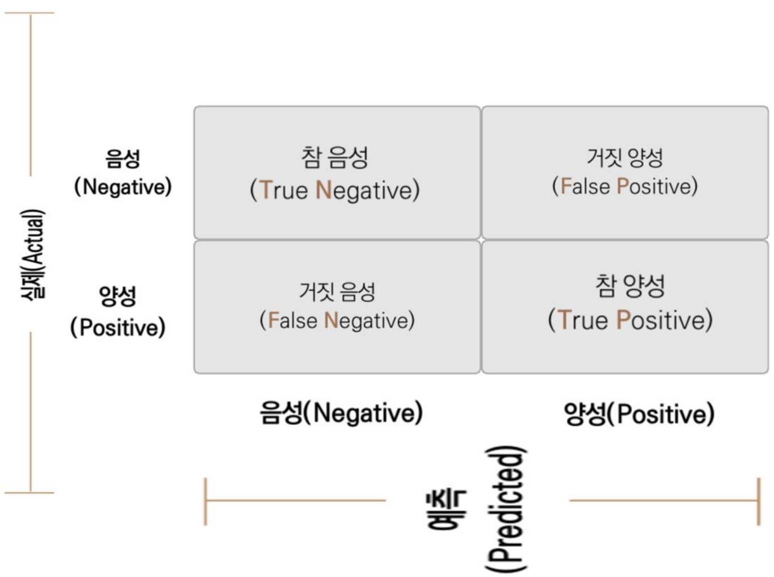
```
confusion_matrix(y_train, pred_train)
```

```
array([[458812,      2],
       [ 17345,     10]])
```

위치에 따른 의미를 잘 비교하자.

```
confusion_matrix(y_test, pred_test)
```

```
array([[114703,      1],
       [  4338,      1]])
```



문제 2 : 안전 운전자 예측

오차행렬을 통한 4가지 평가지표

```
print(classification_report(y_train, pred_train))
```

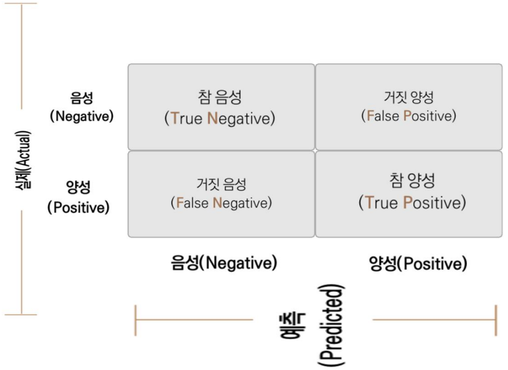
```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98    458814
           1       0.83      0.00      0.00     17355

    accuracy                           0.96    476169
   macro avg       0.90      0.50      0.49    476169
weighted avg       0.96      0.96      0.95    476169
```

샘플수 (support)

양성기준 (0, 1)

단순평균 (macro avg)

샘플에 숫자로 가중치를 준 평균 (weighted avg)




```
array([[458812,      2],
       [ 17345,     10]])
```

문제 2 : 안전 운전자 예측

왜 이런 현상이 생길까?

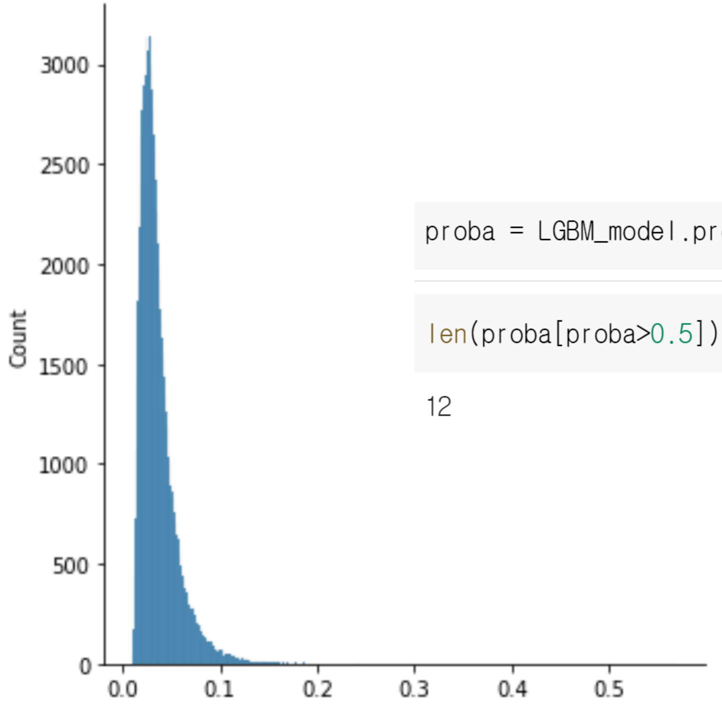
```
array([[458812,      2],
       [ 17345,     10]])
```

전체 정확도를 높이기 위해 양성 예측, 즉 보험금을 받을 사람이라고 예측하는 것을 극도로 꺼렸다.

총 476169명중 단 12명만 보험금을 받을 것이라 예측

```
sns.displot(LGBM_model.predict_proba(X_test)[:,1])
```

<seaborn.axisgrid.FacetGrid at 0x7fdeec9c5790>




```
proba = LGBM_model.predict_proba(X_train)[:,1]
```

```
len(proba[proba>0.5])
```

12

보험금을 탈 확률 분포

보험금을 받을 확률을 어떻게 예측하고, 각 확률에 따라 어떻게 0과 1을 구분 했을까?

```
LGBM_model.predict_proba(X_train)[:,1]
```

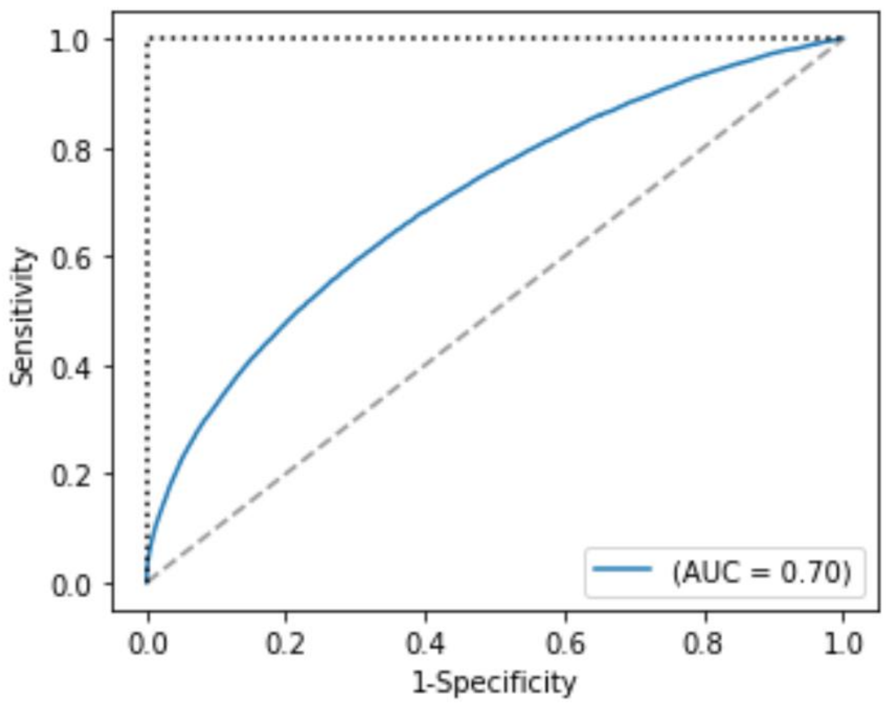
보험금을 탈 확률

문제 2 : 안전 운전자 예측

ROC 곡선의 AUC는 이러한 음성 편향적 결과에 치우치지 않는 평가지표로 사용할 수 있음

```
from sklearn.metrics import roc_curve, auc
```

```
fig = plt.figure(figsize=(5, 4))
fpr, tpr, thresholds = roc_curve(y_train, LGBM_model.predict_proba(X_train)[:,1], pos_label=1)

roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, label='(AUC = {0:0.2f})'.format(roc_auc))

plt.plot([0, 1], [0, 1], linestyle='--', color=(0.6, 0.6, 0.6))
plt.plot([0, 0, 1], [0, 1, 1], linestyle=':', color='black')

plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('1-Specificity')
plt.ylabel('Sensitivity')
plt.legend(loc="lower right")
plt.show()
```



문제 2 : 안전 운전자 예측

로그 손실을 통한 평가지표

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right)\log\left(1 - \hat{y}_i\right)\right)$$

```
def log_loss(y_true,y_proba):
  output = -np.sum(y_true*np.log(y_proba)+(1-y_true)*np.log(1-y_proba))/len(y_true)
  return output
```

```
log_loss(y_train,proba)
```

0.14618002320944476

```
log_loss(y_test,proba_test)
```

0.15267761571032457

문제 2 : 안전 운전자 예측

```
weighted_LGBM_model = LGBMClassifier(force_col_wise=True, scale_pos_weight=20)
weighted_LGBM_model.fit(X_train, y_train)
```

양성 예측에 가중치를 부여함으로써 음성 편향성을 줄일 수 있다.

```
[LightGBM] [Info] Number of positive: 17355, number of negative: 458814
[LightGBM] [Info] Total Bins 1096
[LightGBM] [Info] Number of data points in the train set: 476169, number of used features: 200
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.036447 -> initscore=-3.274764
[LightGBM] [Info] Start training from score -3.274764
```

```
▾ LGBMClassifier
LGBMClassifier(force_col_wise=True, scale_pos_weight=20)
```

```
weighted_pred_train = weighted_LGBM_model.predict(X_train)
weighted_pred_test = weighted_LGBM_model.predict(X_test)
```

```
confusion_matrix(y_train, weighted_pred_train)
```

```
array([[376458,  82356],
       [  9906,   7449]])
```

문제 3 : 코로나-19 수학 모델

문제 3	코로나-19 감염병 확산 과정을 데이터를 통해 관찰하고, 확산 양상을 수학 모델로 나타낸다.

문제 3 : 코로나-19 수학 모델

문제 설정 (해야하는 것, 할 수 있는 것)
코로나-19 감염병 확산 과정을 데이터를 통해 관찰하고, 확산 양상을 수학 모델로 나타낸다.
코로나-19 확산은 어떻게 결정될까? 이 질병은 시간이 지나면 사라질까?
감염병이 발생했을시 확산을 늦추는 방법으로 무엇이 있을까? 그 방법의 효과는?
백신 접종의 효과는 어떻게 될까? 환자를 얼마나 줄일 수 있을까?
백신 접종을 언제 얼마나 맞추어야 가장 효과적일까?
사회적 거리두기 몇 단계가 효과적일까? 앞으로의 감염자 양상을 어떻게 예상할 수 있을까?

문제 3 : 코로나-19 수학 모델

1. 누가, 어떻게 감염되나?

감염병의 특성에 따라 사회구성원을 분류한다면, 어떤 그룹들이 만들어질 수 있는지 토의해 보자.

감염된 사람	**자가 격리된 사람**	**회복된 사람**
감염 될 수 있는 사람	**병원에 입원한 사람**	**잠복기인 사람**

문제 3 : 코로나-19 수학 모델

수학 모델을 완성하려면 이들 사이의 관계를 파악하여야 한다.

어떤 관계가 있을 수 있을까?

문제 3 : 코로나-19 수학 모델

오늘은 5일이다. 밀폐된 방에 감염 될 수 있는 사람 7명, 감염된 사람 3명이 있다. 하루동안 감염된 한 사람이 사람을 마주치는 평균 횟수는 2번이고, 감염시킬 확률을 0.3이라고 하자.

	감염 될 수 있는 사람 수	감염된 사람 수	총 인원
5일	7	3	10

6일에 새로 감염될 것으로 예측되는 사람의 수를 구하여라.

$$1.26 = 2 \times \frac{7}{10} \times 0.3 \times 3$$

감염된 사람 수의 변화 = 감염된 한 사람이 사람을 마주치는 평균 횟수 * (감염될 수 있는 사람 수/총 인구수) * 감염 확률 * 감염된 사람 수

문제 3 : 코로나-19 수학 모델

S	감염될 수 있는 사람(the Susceptible individuals)의 수
I	감염된 사람(the Infected individuals)의 수
R	면역을 가진 사람 (the Recovered individuals)의 수
N	전체 인구 수
α	하루동안 감염된 한 사람이 사람을 마주치는 평균 횟수
p	감염된 사람이 접촉했을 때 감염 확률
ΔI	다음 날 감염된 사람 수의 변화

감염된 사람 수의 변화

= 하루동안 감염된 한 사람이 사람을 마주치는 평균 횟수

* (감염될 수 있는 사람 수/총 인구수)

* 감염 확률

* 감염된 사람 수

$$\Delta I = \alpha\left(\frac{S}{N}\right)pI$$

문제 3 : 코로나-19 수학 모델

감염상수

$$\beta = \alpha p$$

= 하루동안 감염된 한 사람이 사람을 마주치는 평균 횟수 * 감염 확률

감염된 사람 수의 변화

= 하루동안 감염된 한 사람이 사람을 마주치는 평균 횟수 * (감염될 수 있는 사람 수/총 인구수)

* 감염 확률 * 감염된 사람 수

$$\Delta I = \alpha \left(\frac{S}{N}\right) pI = \beta \left(\frac{S}{N}\right) I$$

문제 3 : 코로나-19 수학 모델

S	감염될 수 있는 사람(the Susceptible individuals)의 수
I	감염된 사람(the Infected individuals)의 수
R	회복된 사람 (the Recovered individuals)의 수
N	전체 인구 수
α	하루동안 감염된 한 사람이 사람을 마주치는 평균 횟수
p	감염된 사람이 접촉했을 때 감염 확률
ΔI	다음 날 감염된 사람 수의 변화
β	감염상수

감염된 사람 수의 변화

= 하루동안 감염된 한 사람이 사람을 마주치는 평균 횟수

* (감염될 수 있는 사람 수/총 인구수)

* 감염 확률

* 감염된 사람 수

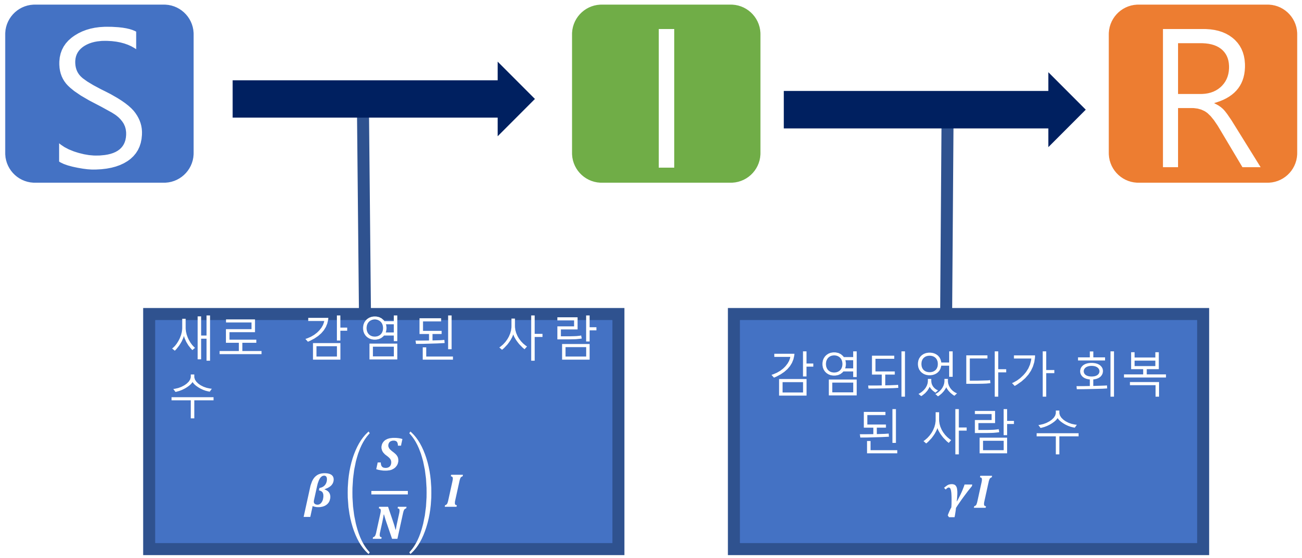
$$\Delta I = \alpha\left(\frac{S}{N}\right)pI = \beta\left(\frac{S}{N}\right)I$$

문제 3 : 코로나-19 수학 모델

사고 순서	I->R 화살표 수식으로 표현하기
필요한 정보	회복률, 감염된 사람의 수
기호로 바꾸기	γ : 회복률, I : 감염된 사람의 수
기호를 사용하여 수식으로 쓰기	γI

문제 3 : 코로나-19 수학 모델

I에서 R로의 화살표의 의미를 설명하고 수식으로 써보자.



문제 3 : 코로나-19 수학 모델

- S_1 은 둘째 날 감염 가능한 사람의 수

  I_1 은 둘째 날 감염된 사람 수

  R_1 은 둘째 날 회복된 사람 수

	감염 가능한 사람 수(S)	감염된 사람 수(I)	회복된 사람(R)
0	S_0	I_0	R_0
1	$S_1 = S_0 - \beta S_0 \frac{I_0}{N}$	$I_1 = I_0 + \beta S_0 \frac{I_0}{N} - \gamma I_0$	$R_1 = R_0 + \gamma I_0$

문제 3 : 코로나-19 수학 모델

SIR COVID-19 수학 모델 (이산모형)

단위 시간당 변화량

$$S_{t+1} = S_t - \beta S_t \frac{I_t}{N}$$

$$I_{t+1} = I_t + \beta S_t \frac{I_t}{N} - \gamma I_t$$

$$R_{t+1} = R_t + \gamma I_t$$

SIR COVID-19 수학 모델
(연속 모형, 상미분방정식)

$$\frac{d}{dt} S(t) = -\beta S(t) \frac{I(t)}{N}$$

$$\frac{d}{dt} I(t) = \beta S(t) \frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt} R(t) = \gamma I(t)$$

오늘의 학습내용

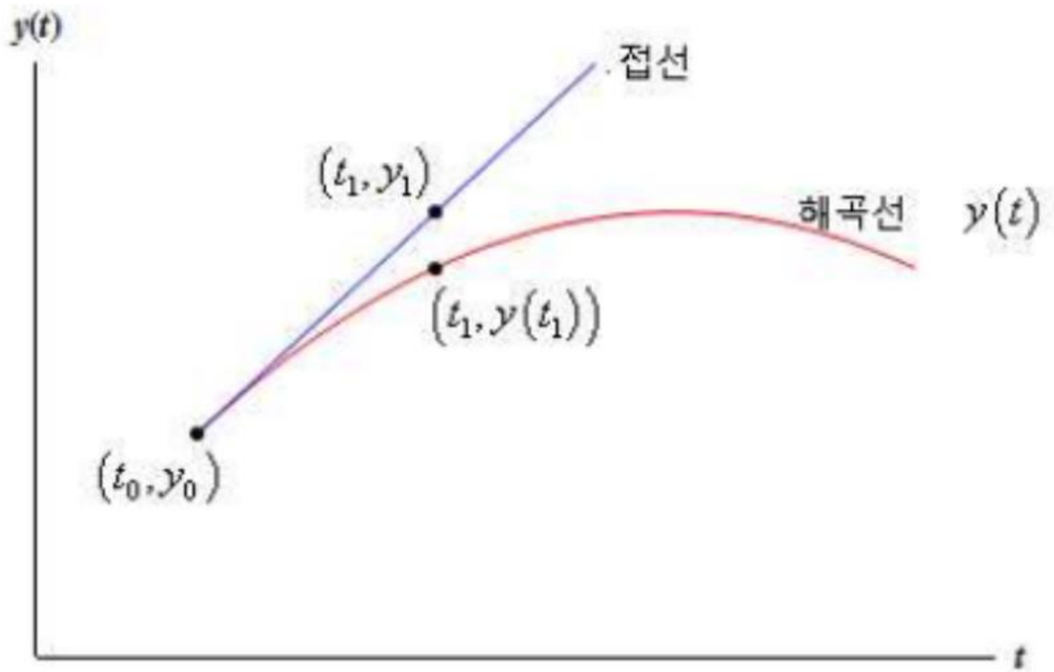
1. 미분방정식 수치해석 기법
2. 고정점과 안정성 검사

문제 3 : COVID-19 수학 모델

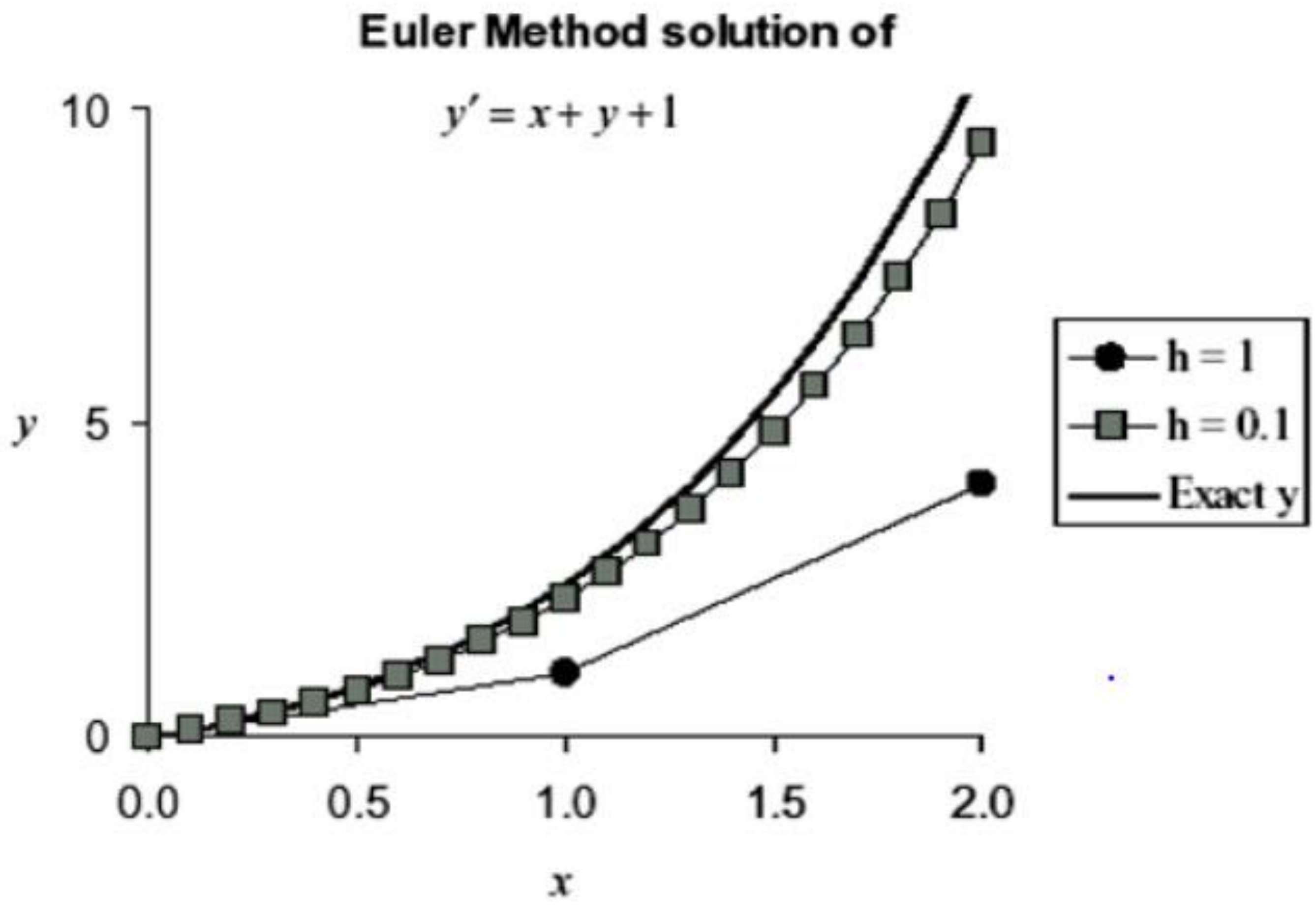
Euler 방법 : 등분된 x값으로부터 근사해를 얻는 방법

초기값 문제 $\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$

$$y_1 = y_0 + f(t_0, y_0)(t_1 - t_0)$$



문제 3 : COVID-19 수학 모델



문제 3 : COVID-19 수학 모델

데이터 분석을 위한 필수 패키지 삼대장

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

시간 변수들을 정의하자.

```
t0 = 0
tf = 300
n = 300
h = (tf-t0)/n
```

시작시간

끝시간

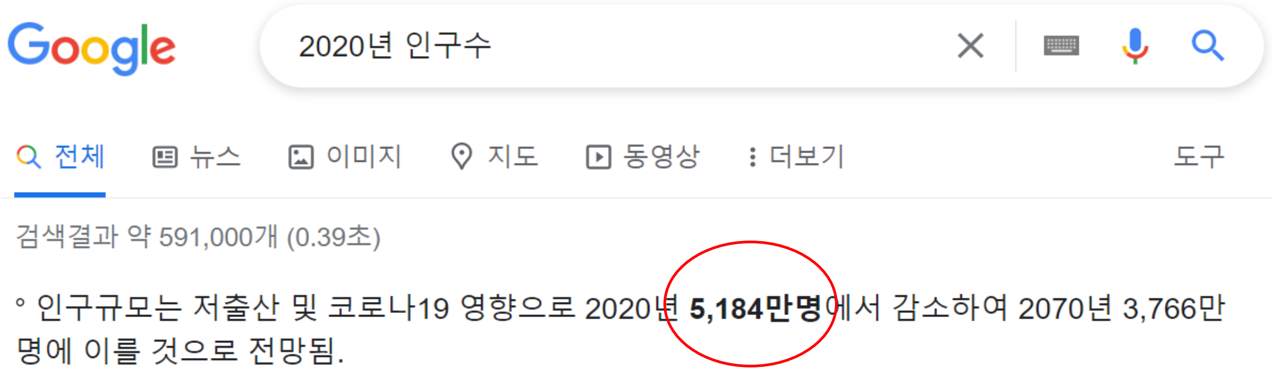
총 시간간격 개수

시간 간격

```
time = np.linspace(t0,tf,n+1)
```

시작점, 끝점, 점의 개수(간격수 +1) 의 3가지 변수로
벡터를 생성할 수 있다.

문제 3 : COVID-19 수학 모델



모든 원소가 0인 행렬 생성 (n+1행, 3열)

```
Euler_y = np.zeros((n+1,3))
initial_value = np.array([51839994.0, 6.0, 0.0])
```

```
Euler_y[0,:] = initial_value
```

$S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,$

6명의 감염자가 처음 발생했다고 가정

문제 3 : COVID-19 수학 모델

수식안에 나머지 변수 정의

```
N = 51840000
beta = 0.4
gamma = 1/14
```

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N}$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt}R(t) = \gamma I(t)$$

$$S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,$$

문제 3 : COVID-19 수학 모델

오일러 방법 (Euler method) 계산

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N}$$

$$y_1 = y_0 + f(t_0, y_0)(t_1 - t_0)$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma(t)I(t)$$

$$\frac{d}{dt}R(t) = \gamma(t)I(t)$$

$$S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,$$

```
S0 = Euler_y[0,0]
I0 = Euler_y[0,1]
R0 = Euler_y[0,2]
```

```
S1 = S0 + (-beta*S0*I0/N)*(time[1]-time[0])
I1 = I0 + (beta*S0*I0/N - gamma*I0)*(time[1]-time[0])
R1 = R0 + (gamma*I0)*(time[1]-time[0])

Euler_y[1,:] = np.array([S1, I1, R1])
```

문제 3 : COVID-19 수학 모델

미분방정식 함수화

```
def f(y_t):
  S, I, R = y_t
  dS = -beta*S*I/N
  dI = beta*S*I/N - gamma*I
  dR = gamma*I
  output = np.array([dS, dI, dR])
  return output
```

Euler 함수화

```
def Euler(f, y_t, h):
  y_t1 = y_t + f(y_t)*h
  return y_t1
```

문제 3 : COVID-19 수학 모델

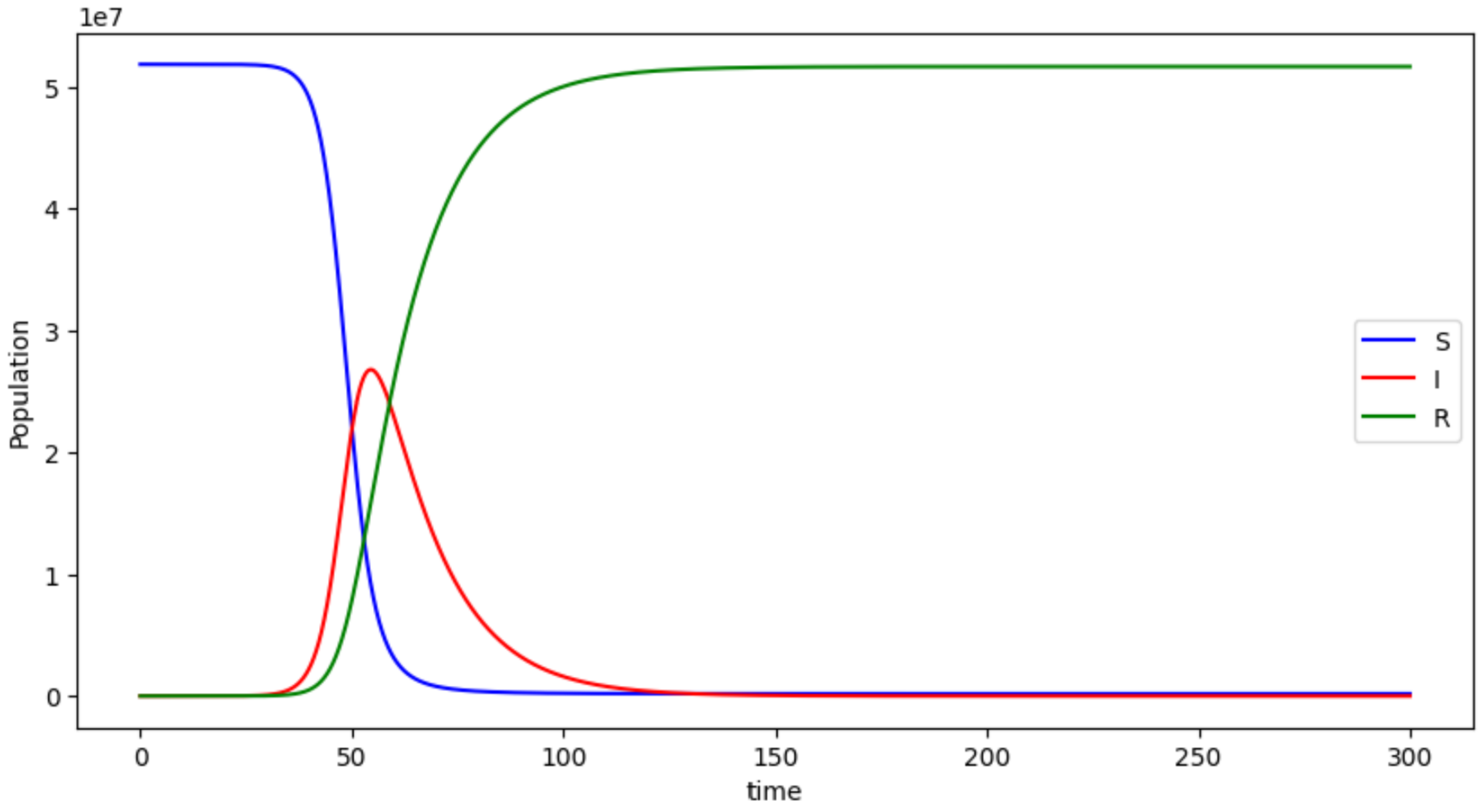
우리의 총 시간 간격은 n개이므로, 이 과정을 n번 반복해야한다.

```
for i in range(n):
  Euler_y[i+1,:] = Euler(f, Euler_y[i,:], h)
```

문제 3 : COVID-19 수학 모델

S, I, R을 그림으로 그려보자.

```
plt.figure(figsize=(10,5))
plt.plot(time,Euler_y[:,0],'b')
plt.plot(time,Euler_y[:,1],'r')
plt.plot(time,Euler_y[:,2],'g')
plt.xlabel('time')
plt.ylabel('Population')
plt.legend(('S','I','R'),loc='best')
plt.show()
```



문제 3 : COVID-19 수학 모델

테일러 급수 (Taylor series)

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots, \quad = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

모든 함수는 그 함수의 미분들의 한 점에서의 값으로 계산된 항의 무한합으로 나타낼 수 있다.

$x = t_{i+1}, a = t_i$,
$f(x) = f(t_{i+1}) = y_{i+1},\ f(a) = f(t_i) = y_i$, 이라고 하면

$$y_{i+1} = y_i + \frac{dy}{dt}(t_{i+1} - t_i) + \frac{1}{2!}\frac{d^2y}{dt^2}(t_{i+1} - t_i)^2$$

$$+ \frac{1}{3!}\frac{d^3y}{dt^3}(t_{i+1} - t_i)^3 + \frac{1}{4!}\frac{d^4y}{dt^4}(t_{i+1} - t_i)^4 + \cdots$$

문제 3 : COVID-19 수학 모델

그렇다면 다음과 같이 어떤 시간 t_i 라는 초기값이 주어진 미분방정식이 있다고 하자.

$$\frac{dy}{dt} = f(t, y), \qquad y(t_i) = y_i$$

우리는 이 y라는 함수의 해를 어떻게 계산할 수 있을까?

어떤 시간 t_i에서 h시간 만큼 지난 시간을 $t_{i+1} = t_i + h$라고 하자.

그때, $y(t_{i+1})$를 $y(t_i)$에 관한 식으로 나타낼 수 있을까?.... 있다!!!

$$y_{i+1} = y_i + \frac{dy}{dt}(t_{i+1} - t_i) + \frac{1}{2!}\frac{d^2y}{dt^2}(t_{i+1} - t_i)^2$$

$$+\frac{1}{3!}\frac{d^3y}{dt^3}(t_{i+1} - t_i)^3 + \frac{1}{4!}\frac{d^4y}{dt^4}(t_{i+1} - t_i)^4 + \quad \ldots$$

문제 3 : COVID-19 수학 모델

만약 두 단위시간의 간격 $t_{i+1} - t_i$가 작은 값이라면 (예를들어 1보다 작은 0에 가까운 수)

$$y_{i+1} = y_i + \frac{dy}{dt}(t_{i+1} - t_i) + \frac{1}{2!}\frac{d^2y}{dt^2}(t_{i+1} - t_i)^2$$

$$+\frac{1}{3!}\frac{d^3y}{dt^3}(t_{i+1} - t_i)^3 + \frac{1}{4!}\frac{d^4y}{dt^4}(t_{i+1} - t_i)^4 + \quad \cdots$$

점점 0에 가까워 진다.

오일러 방법
(Euler method)

$$y_{i+1} \simeq y_i + f(t_i, y_i)h + \frac{1}{2!}f'(t_i, y_i)h^2 + \frac{1}{3!}f''(t_i, y_i)h^3 + \frac{1}{4!}f'''(t_i, y_i)h^4$$

문제 3 : COVID-19 수학 모델

룽게-쿠타 방법 (Runge-Kutta method)

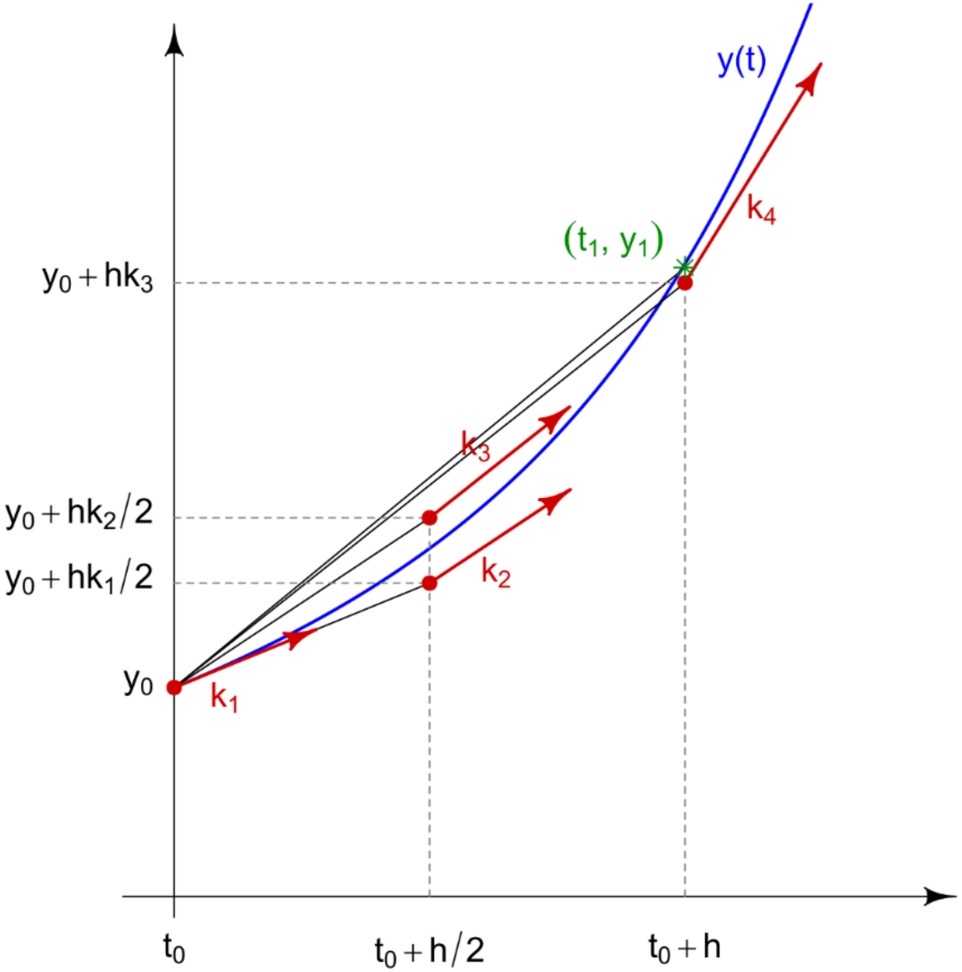
$$y_{i+1} = y_i + (a_1k_1 + a_2k_2 + a_3k_3 + a_4k_4)h$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1h)$$

$$k_3 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2h)$$

$$k_2 = f(t_i + h, y_i + k_3h)$$



문제 3 : COVID-19 수학 모델

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1 h) = f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt} f(t_i, y_i)$$

$$k_3 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2 h) = f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt}\left[f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt} f(t_i, y_i)\right]$$

$$k_4 = f(t_i + h, y_i + k_3 h) = f(t_i, y_i) + h\tfrac{d}{dt}\left[f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt}\left[f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt} f(t_i, y_i)\right]\right]$$

$$\begin{aligned} y_{i+1} = y_i &+ \left[a_1 f(t_i, y_i) + a_2 \left\{f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt} f(t_i, y_i)\right\}\right] h \\ &+ \left[a_3 \left\{f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt}\left[f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt} f(t_i, y_i)\right]\right\}\right] h \\ &+ \left[a_4 \left\{f(t_i, y_i) + h\tfrac{d}{dt}\left[f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt}\left[f(t_i, y_i) + \tfrac{h}{2}\tfrac{d}{dt} f(t_i, y_i)\right]\right]\right\}\right] h \end{aligned}$$

$$= y_i + (a_1 + a_2 + a_3 + a_4) f(t_i, y_i) h + \left(\tfrac{1}{2}a_2 + \tfrac{1}{2}a_3 + a_4\right) f'(t_i, y_i) h^2$$

$$+ \left(\tfrac{1}{4}a_3 + \tfrac{1}{2}a_4\right) f''(t_i, y_i) h^3 + \tfrac{1}{4}a_4 f'''(t_i, y_i) h^4$$

$$\Updownarrow \qquad \Longrightarrow \qquad a_1 = \frac{1}{6}, a_2 = \frac{1}{3}, a_3 = \frac{1}{3}, a_4 = \frac{1}{6}$$

$$y_{i+1} \simeq y_i + f(t_i, y_i) h + \frac{1}{2!} f'(t_i, y_i) h^2 + \frac{1}{3!} f''(t_i, y_i) h^3 + \frac{1}{4!} f'''(t_i, y_i) h^4$$

문제 3 : COVID-19 수학 모델

그럼 어떻게 룽게-쿠타 방법을 적용할까?

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N}$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt}R(t) = \gamma I(t)$$

$S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,$

룽게-쿠타 방법 (Runge-Kutta method)

$$\frac{dy}{dt} = f(t, y), \qquad y(t_i) = y_i$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

이때 여기서,

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1 h)$$

$$k_3 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2 h)$$

$$= f(t_i + h, y_i + k_3 h)$$

문제 3 : COVID-19 수학 모델

그럼 어떻게 룽게-쿠타 방법을 적용할까?

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N}$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt}R(t) = \gamma I(t)$$

$S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,$

룽게-쿠타 방법 (Runge-Kutta method)

$$\frac{dy}{dt} = f(t, y), \qquad y(t_i) = y_i$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

이때 여기서,

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1 h)$$

$$k_3 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2 h)$$

$$= f(t_i + h, y_i + k_3 h)$$

문제 3 : COVID-19 수학 모델

그럼 어떻게 룽게-쿠타 방법을 적용할까?

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N}$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt}R(t) = \gamma I(t)$$

$S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,$

룽게-쿠타 방법 (Runge-Kutta method)

$$\frac{dy}{dt} = \boxed{f(t, y)}, \qquad y(t_i) = y_i$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

이때 여기서,

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1 h)$$

$$k_3 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2 h)$$

$$= f(t_i + h, y_i + k_3 h)$$

문제 3 : COVID-19 수학 모델

그럼 어떻게 룽게-쿠타 방법을 적용할까?

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N}$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt}R(t) = \gamma I(t)$$

$$\boxed{S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,}$$

룽게-쿠타 방법 (Runge-Kutta method)

$$\frac{dy}{dt} = f(t, y), \qquad \boxed{y(t_i) = y_i}$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

이때 여기서,

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1 h)$$

$$k_3 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2 h)$$

$$= f(t_i + h, y_i + k_3 h)$$

문제 3 : COVID-19 수학 모델

룽게-쿠타 방법 (Runge-Kutta method) k1 계산

```
RK4_y = np.zeros((n+1,3))
RK4_y[0,:] = initial_value
```

$$k_1 = f(t_i, y_i)$$

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N}$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma(t)I(t)$$

$$\frac{d}{dt}R(t) = \gamma(t)I(t)$$

$$S(0) = S_0,\ I(0) = I_0,\ R(0) = R_0,$$

```
S0 = RK4_y[0,0]
I0 = RK4_y[0,1]
R0 = RK4_y[0,2]
k1 = f(np.array([S0, I0, R0]))
```

문제 3 : COVID-19 수학 모델

룽게-쿠타 방법 (Runge-Kutta method) 계산

$$k_2 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1 h)$$

```
S0_2 = RK4_y[0,0]+k1[0]*h/2
I0_2 = RK4_y[0,1]+k1[1]*h/2
R0_2 = RK4_y[0,2]+k1[2]*h/2
k2 = f(np.array([S0_2, I0_2, R0_2]))
```

$$k_3 = f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2 h)$$

```
S0_3 = RK4_y[0,0]+k2[0]*h/2
I0_3 = RK4_y[0,1]+k2[1]*h/2
R0_3 = RK4_y[0,2]+k2[2]*h/2
k3 = f(np.array([S0_3, I0_3, R0_3]))
```

$$k_4 = f(t_i + h, y_i + k_3 h)$$

```
S0_4 = RK4_y[0,0]+k3[0]*h
I0_4 = RK4_y[0,1]+k3[1]*h
R0_4 = RK4_y[0,2]+k3[2]*h
k4 = f(np.array([S0_4, I0_4, R0_4]))
```

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

```
RK4_y[1,:]=RK4_y[0,:]+h*(k1+2*k2+2*k3+k4)/6
```

문제 3 : COVID-19 수학 모델

룽게-쿠타 방법 (Runge-Kutta method) 함수화

```
def rk4(f, y_t, h):
  k1 = f(y_t)
  k2 = f(y_t+k1*h/2)
  k3 = f(y_t+k2*h/2)
  k4 = f(y_t+k3*h)
  y_t1 = y_t+h*(k1+2*k2+2*k3+k4)/6
  return y_t1
```

문제 3 : COVID-19 수학 모델

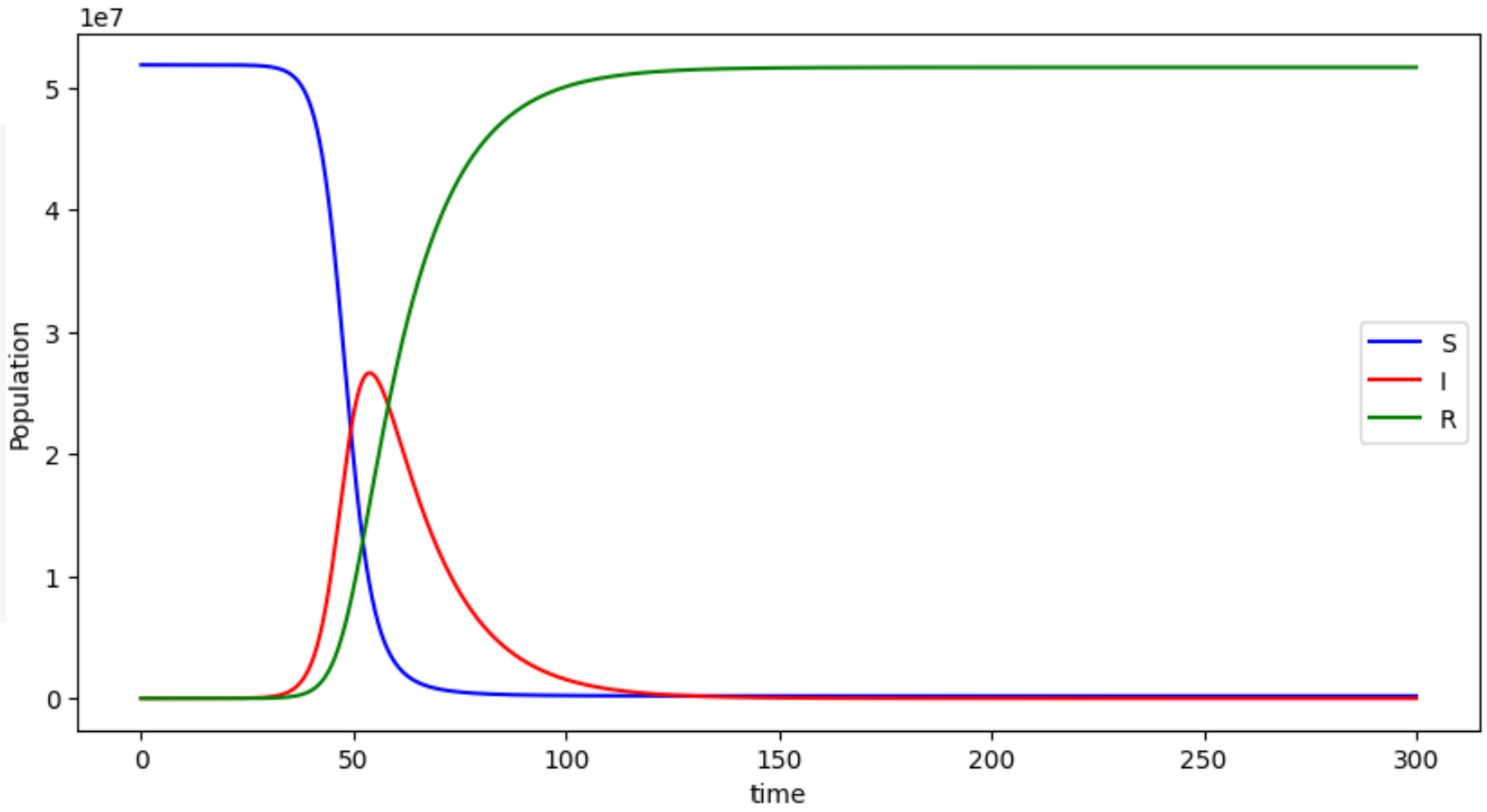
우리의 총 시간 간격은 n개이므로, 이 과정을 n번 반복해야한다.

```
for i in range(n):
  RK4_y[i+1,:] = rk4(f, RK4_y[i,:], h)
```

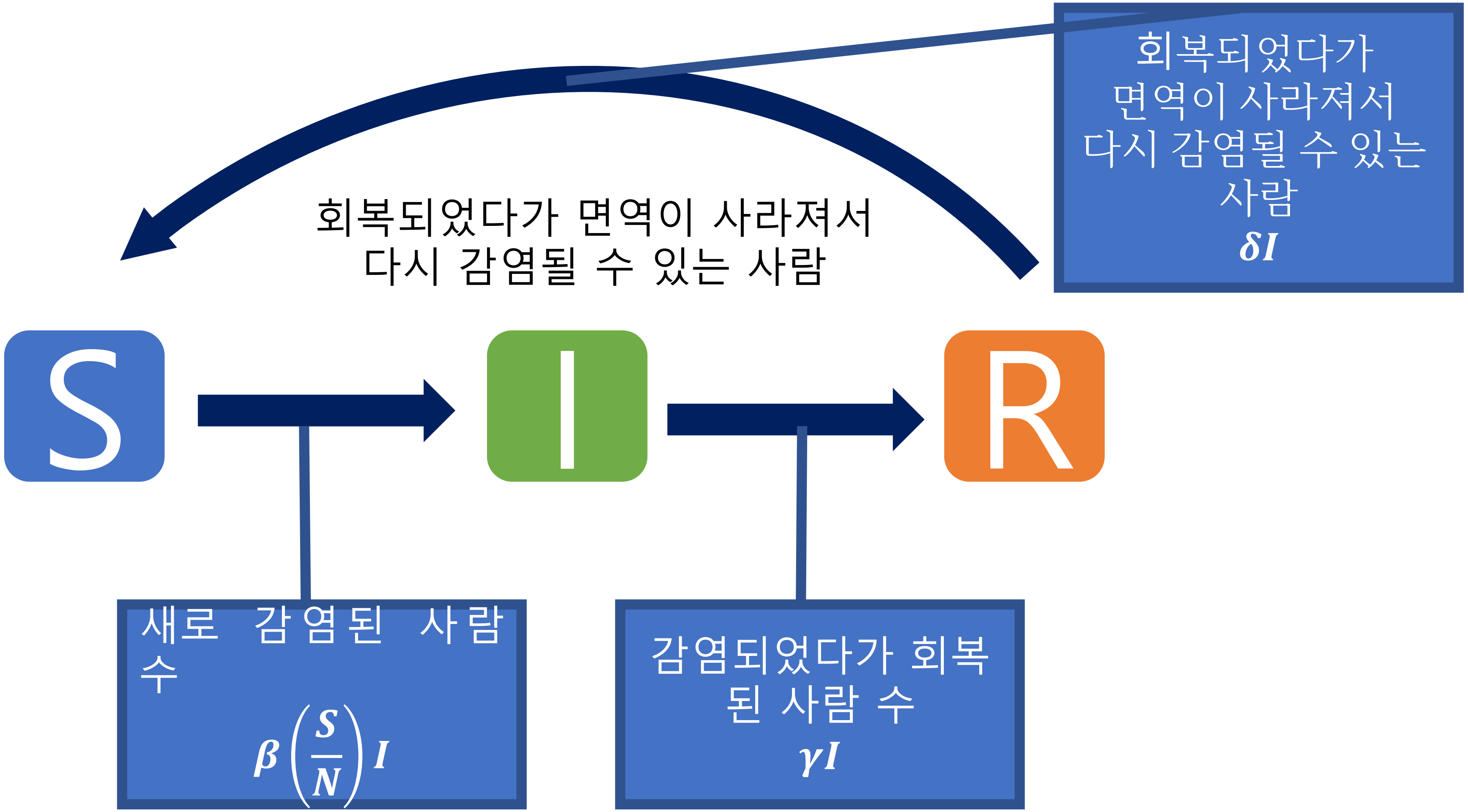
문제 3 : COVID-19 수학 모델

S, I, R을 그림으로 그려보자.

```
plt.figure(figsize=(10,5))
plt.plot(time,RK4_y[:,0],'b')
plt.plot(time,RK4_y[:,1],'r')
plt.plot(time,RK4_y[:,2],'g')
plt.xlabel('time')
plt.ylabel('Population')
plt.legend(('S','I','R'),loc='best')
plt.show()
```



문제 3 : COVID-19 수학 모델



문제 3 : COVID-19 수학 모델

재감염에 대한 보도자료

http://ncov.mohw.go.kr/tcmBoardView.do?brdId=3&brdGubun=31&dataGubun=&ncvContSeq=6812&board_id=312&contSeq=6812

○ 7월간 발생한 2회감염 추정사례의 평균 소요기간은 154~165일(약 5개월)로 '22년 6월까지 발생한 2회감염 추정사례(평균 229일) 보다 약 60여 일 빨라져, 최초감염 후 2회감염이 발생하는 기간이 단축된 것으로 나타났다.

* 2회감염 소요기간 : 2회감염일 – 최초감염일

$$\delta = \frac{1}{229} \text{ 로 가정}$$

문제 3 : COVID-19 수학 모델

SIR COVID-19 수학 모델
(연속 모형, 상미분방정식)

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N} + \delta R(t)$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt}R(t) = \gamma I(t) - \delta R(t)$$

문제 3 : COVID-19 수학 모델

중국 내 신종 코로나 바이러스 감염증 환자 1명이 만들어 내는 최대 감염된 사람 수가 3.6명까지 증가했다는 분석이 나왔다. 보통 감염병 환자 1명이 다른 사람에게 바이러스를 옮길 수 있는 감염력을 '재생산지수'(R)라는 개념으로 추정한다. 이 지수치가 1이면 한 사람이 다른 한사람에게만 바이러스를 감염시킨다는 의미다. 재생산지수가 높아질수록 감염력이 강하다고 볼 수 있다. 메르스(MERS, 중동호흡기증후군)와 사스(SARS, 중증급성호흡기증후군)의 재생산지수는 각각 0.4~0.9명, 4명이었다. 다만, 메르스의 경우 2015년 한국에서 유행할 당시만 보면 재생산지수가 4명에 달했다.

출처: 중국 신종코로나 1명이 최대 3.6명 전파…한국 '메르스' 수준
https://www.yna.co.kr/view/AKR20200205164500017

문제 3 : COVID-19 수학 모델

기초감염재생산수(R_{0}, Basic reproduction number)

최초 감염자가 감염기간 동안에 감염시킬 수 있는 2차 감염자의 수

R_0 =	감염 확률	x	접촉빈도	x	전파기간
	transmissibility		number of contact per day		Duration of infectious period

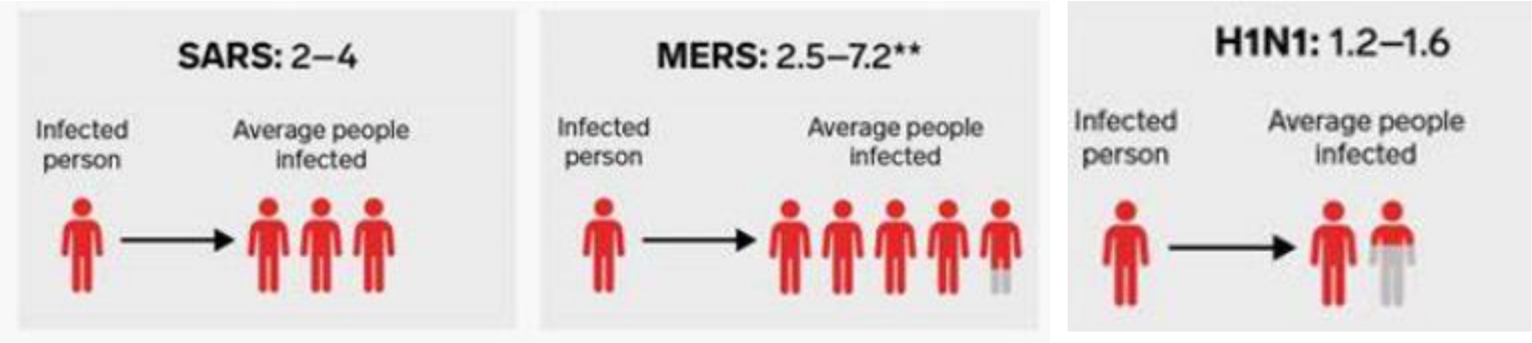
감염 확산 방지 정책

문제 3 : COVID-19 수학 모델

기초감염재생산수(R_0, Basic reproduction number)

- 감염병 확산 규정의 중요한 지표

- **홍역: 12-18**
- **인플루엔자: 1-3**



문제 3 : COVID-19 수학 모델

단위 시간당 접촉

환자들의 감염 전파기간

$$\mathcal{R}_0 = \boldsymbol{\alpha} \times \boldsymbol{p} \times \frac{\mathbf{1}}{\boldsymbol{\gamma}} = \frac{\boldsymbol{\beta}}{\boldsymbol{\gamma}}$$

접촉 당 감염될 확률

문제 3 : COVID-19 수학 모델

질병이 사라지는 조건 ⟺ $R_0 < 1$

질병이 사라지지 않고 계속 남아 있을 조건 ⟺ $R_0 > 1$

문제 3 : COVID-19 수학 모델

고정점과 안정성 검사

문제 3 : COVID-19 수학 모델

기존 수식에서 재감염을 적용해보자.

```
delta = 1/229
```

```
def f2(y_t):
  S, I, R = y_t
  dS = -beta*S*I/N + delta*R
  dI = beta*S*I/N - gamma*I
  dR = gamma*I - delta*R
  k = np.array([dS, dI, dR])
  return k
```

$$\frac{d}{dt}S(t) = -\beta S(t)\frac{I(t)}{N} + \delta R(t)$$

$$\frac{d}{dt}I(t) = \beta S(t)\frac{I(t)}{N} - \gamma I(t)$$

$$\frac{d}{dt}R(t) = \gamma I(t) - \delta R(t)$$

문제 3 : COVID-19 수학 모델

y를 y2로 새로 정의하고, 변경된 함수로 룽게-쿠타를 다시 적용하여 시뮬레이션 해보자.

```
y2 = np.zeros((n+1,3))
y2[0,:] = initial_value
```

```
for i in range(n):
  y2[i+1,:] = rk4(f2, y2[i,:], h)
```

문제 3 : COVID-19 수학 모델

그럼 현재 코드에서 기초감염재생산수는??

```
R0 =beta/gamma
```

```
R0
```

5.6000000000000005 **>1**

질병이 사라지지 않고
계속 남아 있을 조건 ⟷ $R_0 > 1$

단위 시간당 접촉

환자들의 감염 전파기간

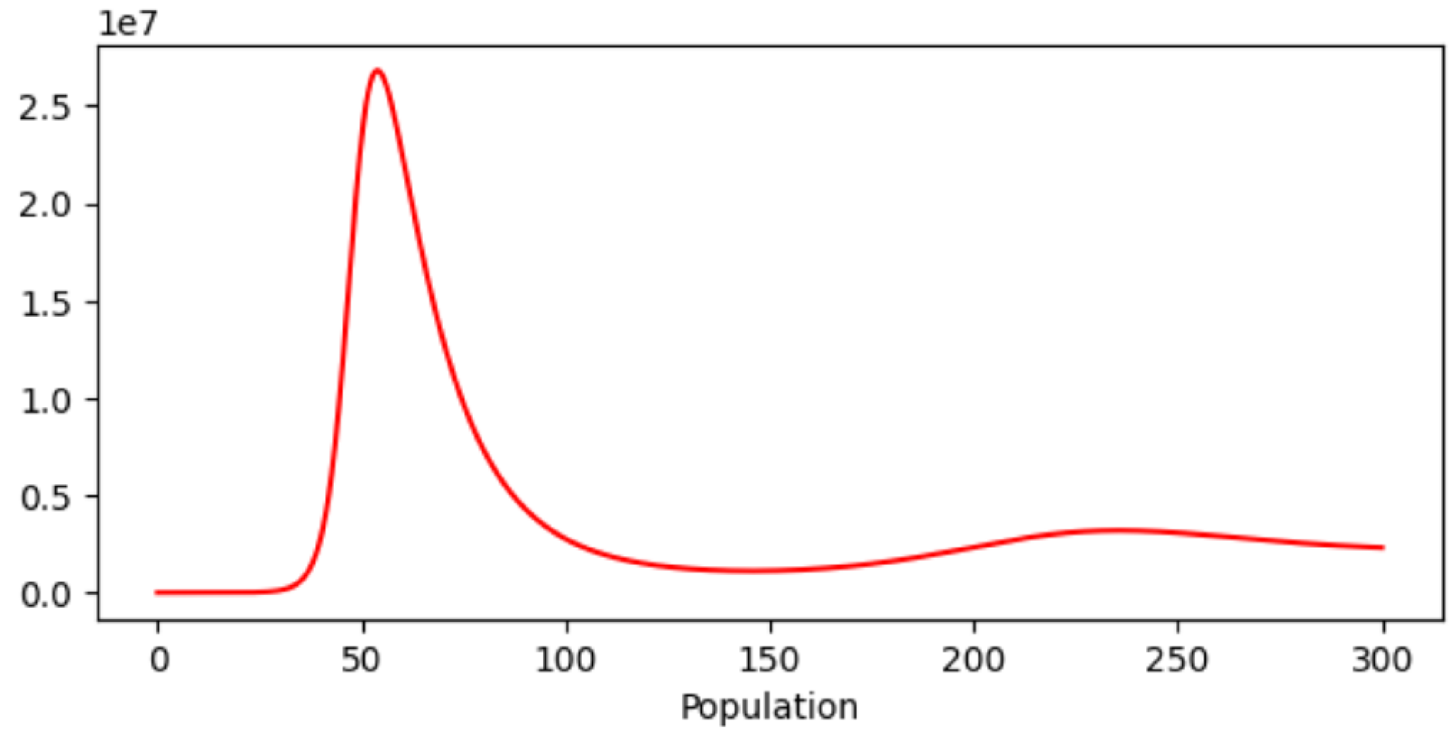
$$\mathcal{R}_0 = \alpha \times p \times \frac{1}{\gamma} = \frac{\beta}{\gamma}$$

접촉 당 감염될 확률

문제 3 : COVID-19 수학 모델

실제 감염자인 I가 0이 되지 않는지 그림으로 그려보자.

```
plt.figure(figsize=(7,3))
plt.plot(time,y2[:,1],'r')
plt.xlabel('time')
plt.xlabel('Population')
plt.show()
```



문제 3 : COVID-19 수학 모델

그럼 기초감염재생산수(R0)를 1보다 작게 할려면 어떻게 해야할까??

beta를 낮추거나 gamma를 높여야 한다.

beta = 0.05로 변경하여 다시 시뮬레이션 해보자.

문제 3 : COVID-19 수학 모델

```
beta = 0.05
```

```
y2 = np.zeros((n+1,3))
y2[0,:] = initial_value
```

```
for i in range(n):
  y2[i+1,:] = rk4(f2, y2[i,:], h)
```

```
R0 =beta/gamma
```

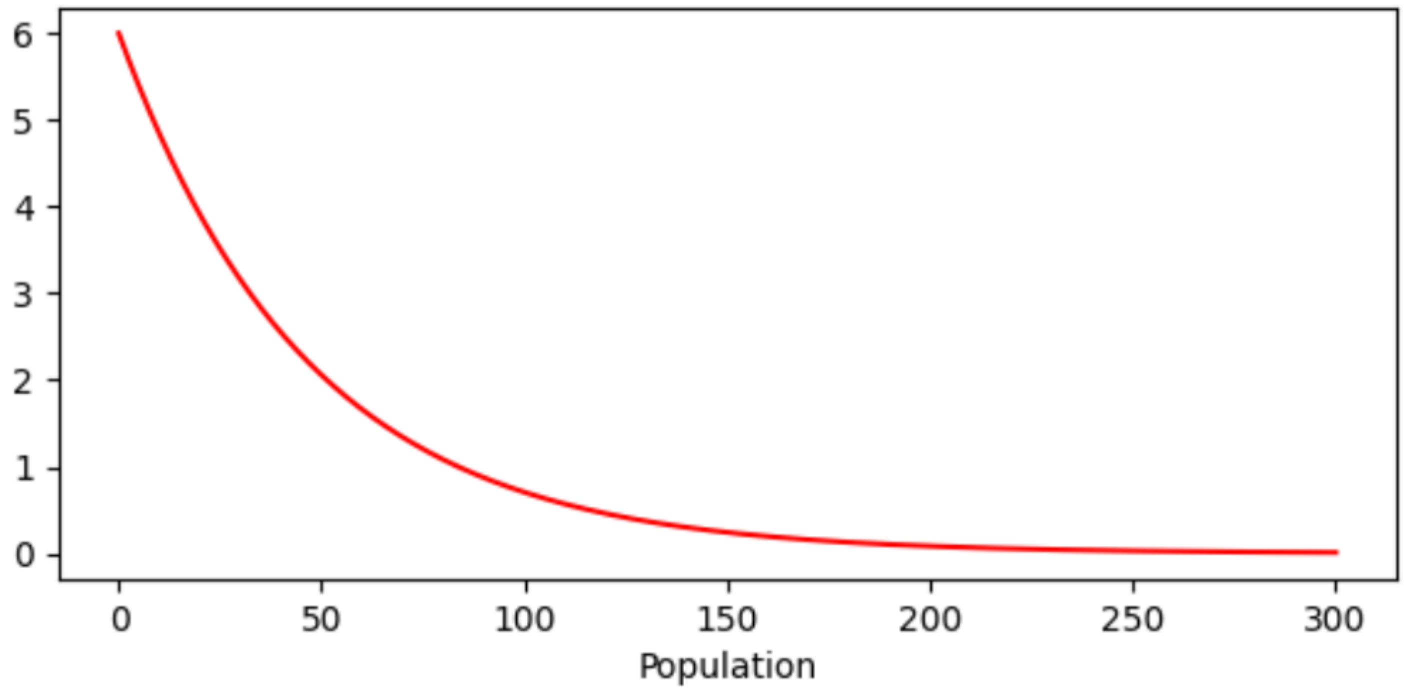
```
R0
```

0.7000000000000001 **<1** → 질병이 사라지는 조건 ⟷ $R_0 < 1$

문제 3 : COVID-19 수학 모델

실제 감염자인 I가 0이 되는지 그림으로 그려보자.

```
plt.figure(figsize=(7,3))
plt.plot(time,y2[:,1],'r')
plt.xlabel('time')
plt.xlabel('Population')
plt.show()
```



Thank you