회귀분석 및 실습 Homework 6

서울대학교 통계학과 2017-11362 박건도

2021년 04월 28일

Get COVID-19 data

```
covid <- read.csv('./global_confirmed_cases_210420.csv')
str(covid)
```

```
## 'data.frame':    75148 obs. of  6 variables:
##  $ CountryName: Factor w/ 183 levels "Afghanistan",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ CountryCode: Factor w/ 183 levels "ABW","AFG","AGO",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date       : Factor w/ 455 levels "2020.1.22","2020.1.23",..: 136 137 138 139 140 141 142 144 145
##  $ Cases      : int  2 2 2 2 3 4 4 5 5 9 ...
##  $ Difference : int  2 0 0 0 1 1 0 1 0 4 ...
##  $ Days       : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
ISR <- covid %>%
  filter(CountryCode == 'ISR') %>%
  select(Days, Cases, Difference)
str(ISR)
```

```
## 'data.frame':    425 obs. of  3 variables:
##  $ Days      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Cases     : int  1 1 1 1 1 2 3 6 6 7 ...
##  $ Difference: int  1 0 0 0 0 1 1 3 0 1 ...
```

우리의 회귀분석에 사용될 나라는 이스라엘이고, 모형에 있어 시간과 확진자 수가 인수로 주어지기 때문에, 나머지 불필요한 데이터들을 제외한 데이터 프레임 `ISR`을 만들었다.

Logistic Model

우선, 위에서 얻은 데이터로 로지스틱 모델에 대해 비선형 회귀분석을 실시해보자. 로지스틱 모델의 식은 다음과 같다.

$$y = \frac{A}{1 + e^{\beta_0 - \beta_1 x}}$$

여기서 y는 Cases가 되고, x는 Days가 된다. 위 식과 nls2 함수를 이용하여 brute-force 방식으로 대략적 해를 구한 뒤, Gauss-Newton 알고리즘을 사용하여 국소해를 찾을 것이다.

```
form_logit <- Cases ~ A / (1 + exp(beta0 - beta1 * Days))
grid_logit <- data.frame(A = c(0, max(ISR$Cases)), beta0 = c(0, 100), beta1 = c(0, 1))
rough_fit_logit <- nls2(form_logit, data = ISR, start = grid_logit, algorithm = "brute-force")
summary(rough_fit_logit)
```

```
##
## Formula: Cases ~ A/(1 + exp(beta0 - beta1 * Days))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## A     5.580e+05  1.166e+04  47.844  < 2e-16 ***
## beta0 6.667e+01  2.561e+01   2.603  0.00958 **
## beta1 3.333e-01  1.281e-01   2.602  0.00959 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 170800 on 422 degrees of freedom
##
## Number of iterations to convergence: 64
## Achieved convergence tolerance: NA
```

```
gn_fit_logit <- nls2(form_logit, data = ISR, start = rough_fit_logit)
summary(gn_fit_logit)
```

```
##
## Formula: Cases ~ A/(1 + exp(beta0 - beta1 * Days))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## A     1.109e+06  2.431e+04   45.60   <2e-16 ***
## beta0 4.968e+00  6.779e-02   73.28   <2e-16 ***
## beta1 1.509e-02  3.336e-04   45.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34960 on 422 degrees of freedom
```

```
##
## Number of iterations to convergence: 10
## Achieved convergence tolerance: 3.647e-06
```

```
coef(gn_fit_logit) # coefficients
```

```
##            A        beta0        beta1
## 1.108564e+06 4.967760e+00 1.509331e-02
```

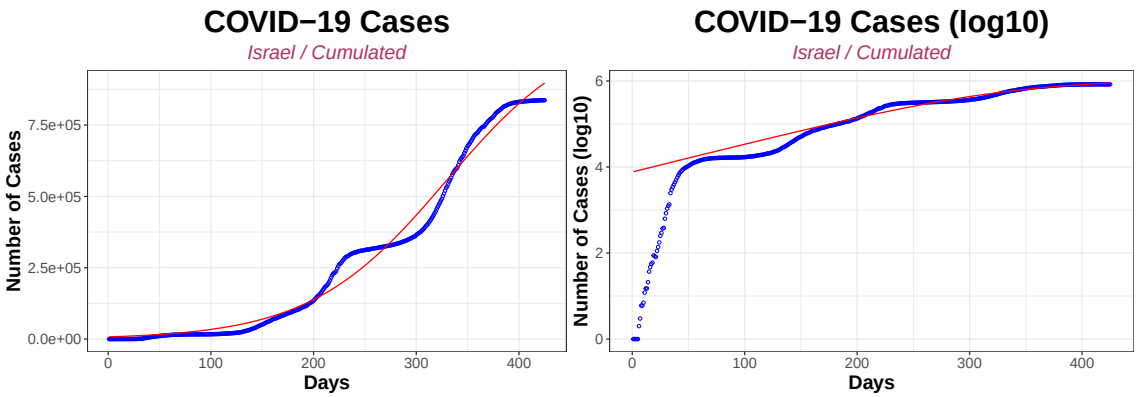
```
deviance(gn_fit_logit) # SSE
```

```
## [1] 515901948727
```

```
getMSE(ISR$Cases, predict(gn_fit_logit, ISR$Cases)) # MSE
```

```
## [1] 1213886938
```

fitting의 결과를 그림으로 표현하면 아래와 같다.



위 그림에서 푸른 점은 실제 데이터를 나타내고, 빨간 선은 로지스틱 모델로 fitting된 값을 의미한다. 오른쪽의 그림은 왼쪽의 그림에서 y축을 log scale로 바꾼 것이다.

Bertalanffy Model

버탈란피 모델은 아래와 같은 식으로 회귀분석을 하면 된다. 그런데 `Days`가 0일때 확진자 수가 0이 되도록 데이터를 조정했으므로, 다음과 같이 $y(0) = 0$을 위한 x에 대한 조정 항 x_0는 없어도 무방하다.

$$y = A\left(1 - e^{-\beta_1 x}\right)^{\beta_0}$$

```
form_bert <- Cases ~ A * (1 - exp(- beta1 * Days)) ^ beta0
grid_bert <- data.frame(A = c(0, max(ISR$Cases)*3), beta0 = c(1, 10), beta1 = c(0, 0.01))
rough_fit_bert <- nls2(form_bert, data = ISR, start = grid_bert, algorithm = "brute-force")
summary(rough_fit_bert)
```

```
##
## Formula: Cases ~ A * (1 - exp(-beta1 * Days))^beta0
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## A     2.511e+06  5.776e+05   4.348 1.72e-05 ***
## beta0 4.000e+00  4.593e-01   8.710  < 2e-16 ***
## beta1 3.333e-03  6.353e-04   5.247 2.45e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53070 on 422 degrees of freedom
##
## Number of iterations to convergence: 64
## Achieved convergence tolerance: NA
```

```
gn_fit_bert <- nls2(form_bert, data = ISR, start = rough_fit_bert,
                    control = nls.control(warnOnly = TRUE))
coef(gn_fit_bert) # coefficients
```

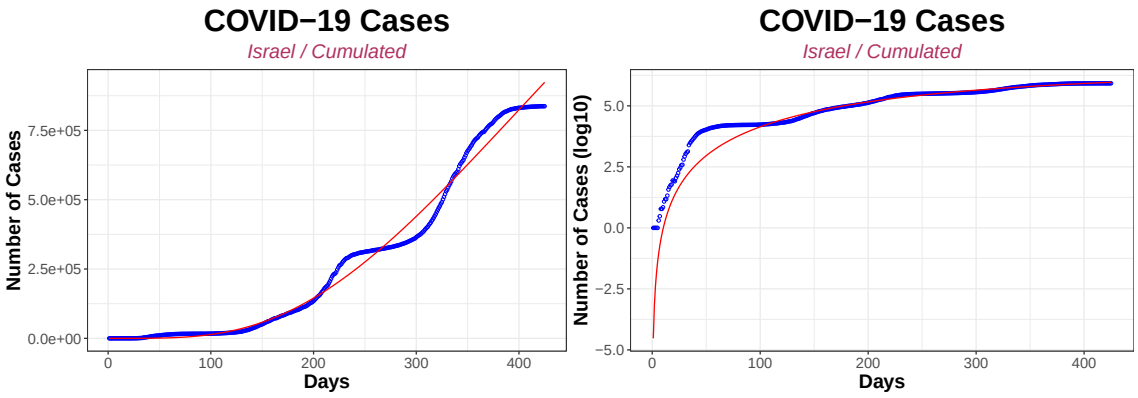
```
##            A        beta0        beta1
## 2.535472e+06 4.510317e+00 3.779195e-03
```

```
deviance(gn_fit_bert) # SSE
```

```
## [1] 519161061214
```

```
getMSE(ISR$Cases, predict(gn_fit_bert, ISR$Cases)) # MSE
```

```
## [1] 1221555438
```



로지스틱 모형에 비해 0 근방에서 비교적 fitting이 잘 되나, 해당 근방의 값을 완벽하게 설명하고 있지는 않다.

Gompertz Model

곰파츠 모델은 다음과 같다.

$$y = Ae^{-e^{\beta_0 - \beta_1 x}}$$

```
form_gomp <- Cases ~ A * exp(-exp(beta0 - beta1 * Days))
grid_gomp <- data.frame(A = c(0, max(ISR$Cases)), beta0 = c(0,1), beta1 = c(0, 0.5))
rough_fit_gomp <- nls2(form_gomp, data = ISR, start = grid_gomp, algorithm = "brute-force")
summary(rough_fit_gomp)
```

```
##
## Formula: Cases ~ A * exp(-exp(beta0 - beta1 * Days))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## A     2.790e+05  1.443e+04  19.333   <2e-16 ***
## beta0 1.000e+00  1.721e+00   0.581    0.561
## beta1 1.667e-01  1.888e-01   0.883    0.378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 287200 on 422 degrees of freedom
##
## Number of iterations to convergence: 64
## Achieved convergence tolerance: NA
```

```
gn_fit_gomp <- nls2(form_gomp, data = ISR, start = rough_fit_gomp)
summary(gn_fit_gomp)
```

```
##
## Formula: Cases ~ A * exp(-exp(beta0 - beta1 * Days))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## A     1.847e+06  1.052e+05   17.55   <2e-16 ***
## beta0 2.091e+00  3.432e-02   60.92   <2e-16 ***
## beta1 5.764e-03  2.480e-04   23.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 34860 on 422 degrees of freedom
##
## Number of iterations to convergence: 23
## Achieved convergence tolerance: 2.66e-06
```

```
coef(gn_fit_gomp) # coefficients
```

```
##            A        beta0        beta1
## 1.847075e+06 2.091123e+00 5.764306e-03
```

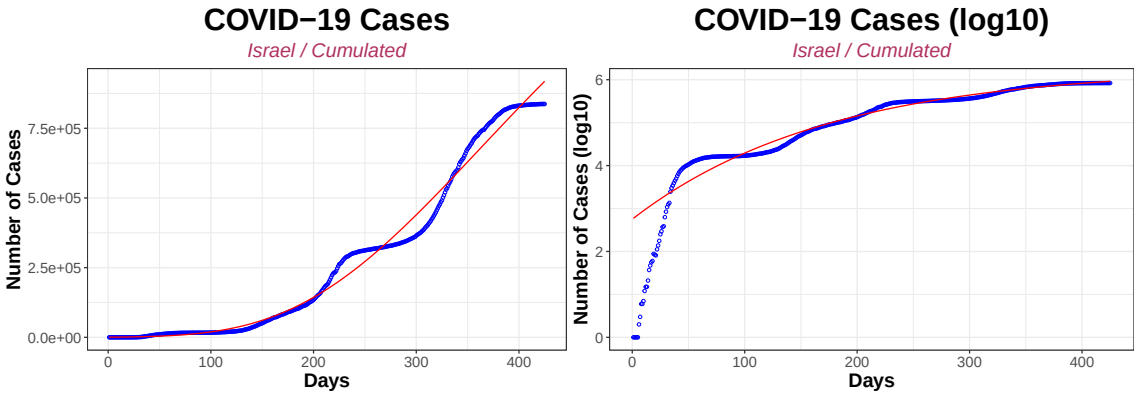
```
deviance(gn_fit_gomp) # SSE
```

```
## [1] 512747252090
```

```
getMSE(ISR$Cases, predict(gn_fit_gomp, ISR$Cases)) # MSE
```

```
## [1] 1206464123
```

fitting의 결과를 그림으로 표현하면 아래와 같다.



곰파츠 모델은 로지스틱 모델과 비슷한 경향성을 보이는 것을 알 수 있다.

Comparison

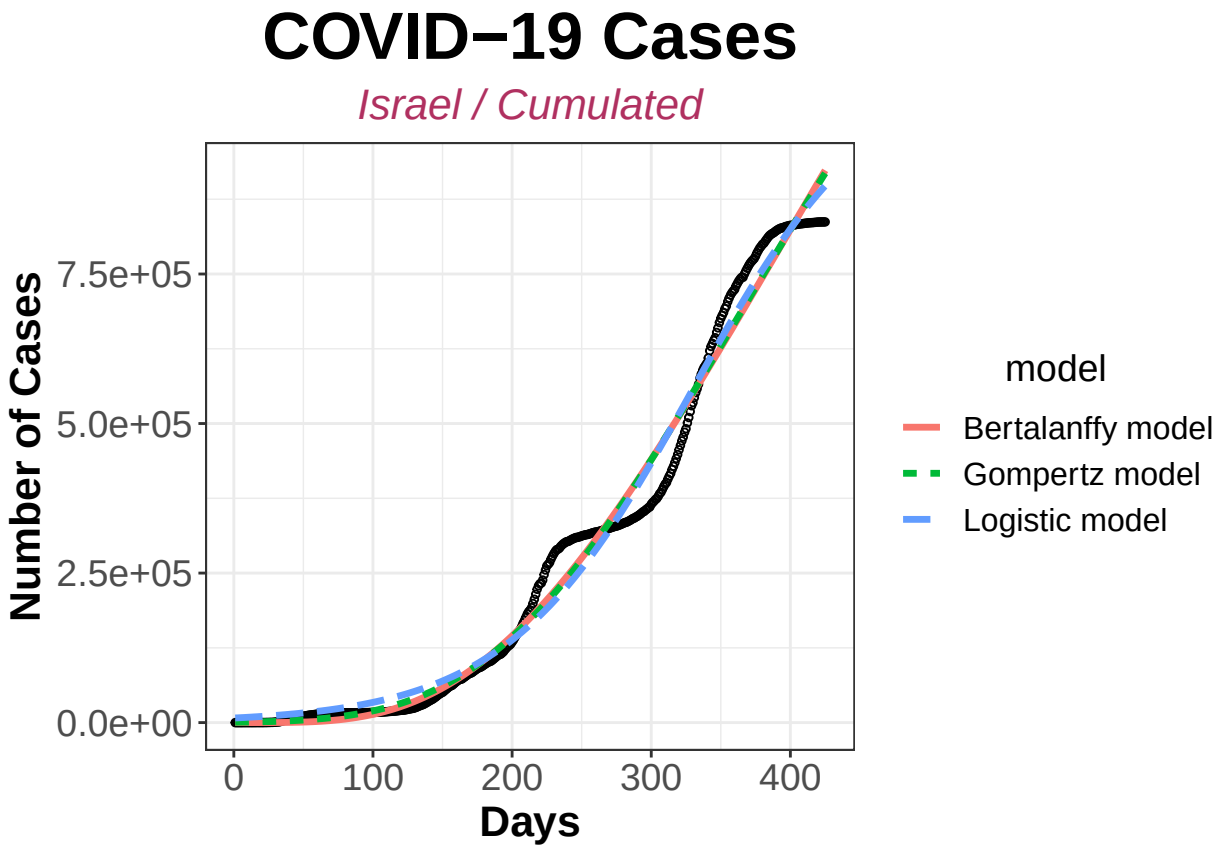
마지막으로, 세 모델을 한 그래프에 나타내고, 일별 확진자 수도 비교해보자.

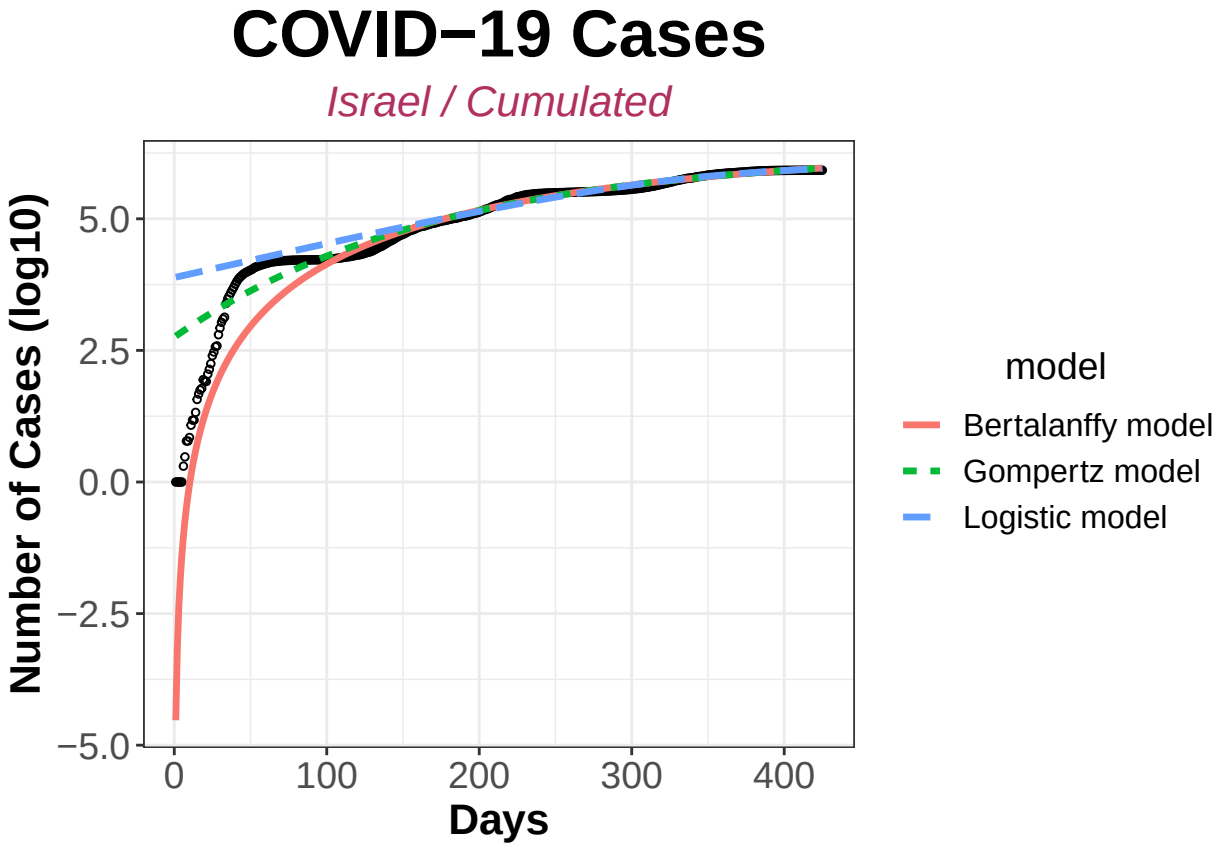
```
ISR_predict <- ISR %>%
  select(Days, y_logit, y_bert, y_gomp) %>%
  mutate(Diff_logit = c(0, y_logit[-1] - y_logit[-length(y_logit)])) %>%
  mutate(Diff_bert = c(0, y_bert[-1] - y_bert[-length(y_bert)])) %>%
  mutate(Diff_gomp = c(0, y_gomp[-1] - y_gomp[-length(y_gomp)])) %>%
  gather(model, Cases, y_logit, y_bert, y_gomp) %>%
  mutate(model = ifelse(model == "y_logit", "Logistic model",
                        ifelse(model == "y_bert", "Bertalanffy model", "Gompertz model"))) %>%
  mutate(Difference = ifelse(model == "Logistic model", Diff_logit,
```

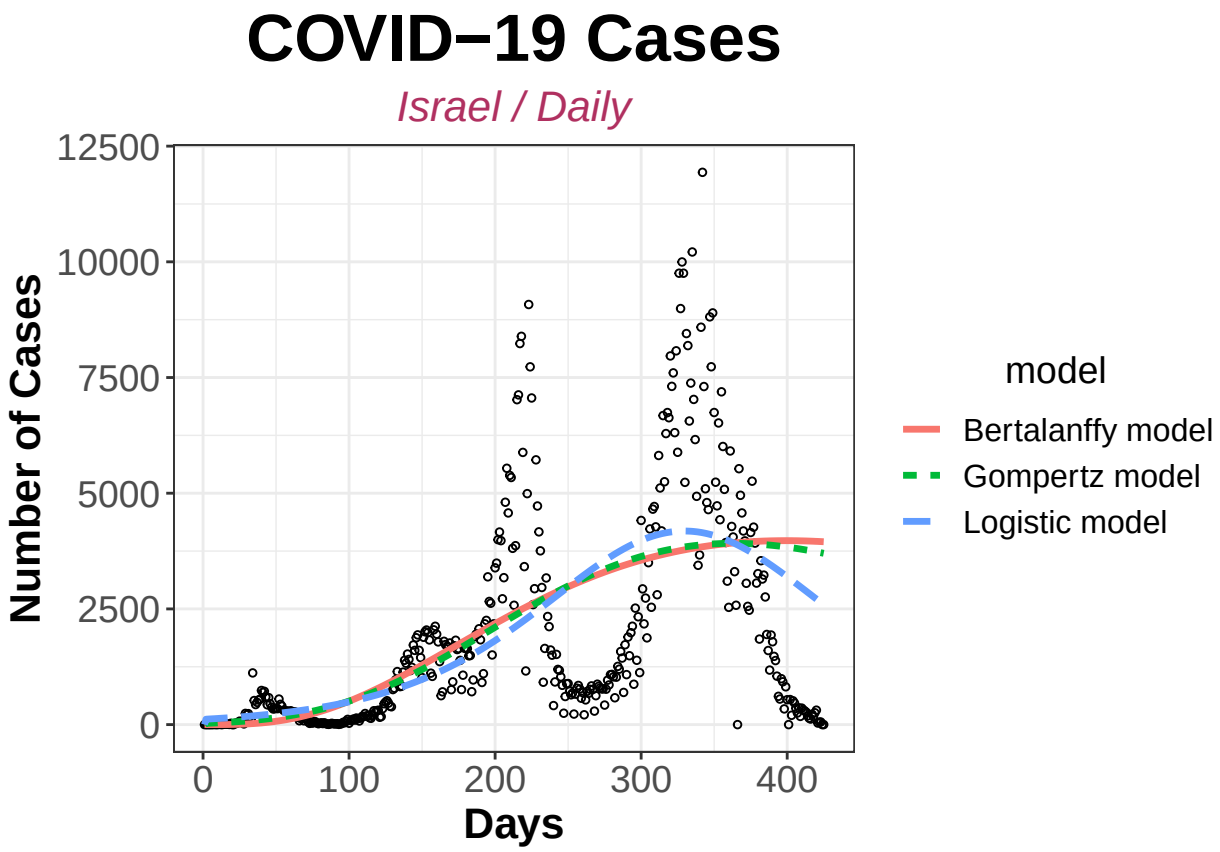
```
                              ifelse(model == "Gompertz model", Diff_gomp, Diff_bert))) %>%
  select(-Diff_logit, -Diff_bert, -Diff_gomp)

MSE_vec <- ISR_predict %>%
  select(-Difference) %>%
  spread(model, Cases) %>%
  sapply(function(yhat) getMSE(ISR$Cases, yhat))

Rsq_vec <- ISR_predict %>%
  select(-Difference) %>%
  spread(model, Cases) %>%
  sapply(function(yhat) getRsq(ISR$Cases, yhat))

result <- rbind(coef(gn_fit_bert), coef(gn_fit_gomp), coef(gn_fit_logit)) %>%
  cbind(MSE_vec[2:4], Rsq_vec[2:4])

colnames(result)[4:5] <- c("MSE", "R^2")
as.data.frame(result)
```

	A	beta0	beta1	MSE	R^2
Bertalanffy model	2535472	4.510317	0.0037792	1221555438	0.9855089
Gompertz model	1847075	2.091123	0.0057643	1206464123	0.9856880
Logistic model	1108564	4.967760	0.0150933	1213886938	0.9855999

참고로 위 표에 나타난 계수들, `beta0`, `beta1`은 각각이 설명하는 변수가 다르므로 위 식을 참고하도록 하고, `A`는 최대 확진자수를 의미한다. R^2 값은 곰파츠 모델이 제일 크고, 로지스틱 모델이 그 다음, 마지막이 버탈란피 모델이지만 세 값이 매우 유사함을 알 수 있다.


COVID−19 Cases
Israel / Cumulated
Number of Cases
7.5e+05
5.0e+05
2.5e+05
0.0e+00
0
100
200
300
400
Days
model
Bertalanffy model
Gompertz model
Logistic model


COVID−19 Cases
Israel / Cumulated
Number of Cases (log10)
5.0
2.5
0.0
−2.5
−5.0
0
100
200
300
400
Days
model
Bertalanffy model
Gompertz model
Logistic model


COVID−19 Cases
Israel / Daily
Number of Cases
12500
10000
7500
5000
2500
0
0
100
200
300
400
Days
model
Bertalanffy model
Gompertz model
Logistic model