회귀분석 및 실습 Homework 6

서울대학교 통계학과 2017-11362 박건도

2021년 04월 25일

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 Days$가 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
```

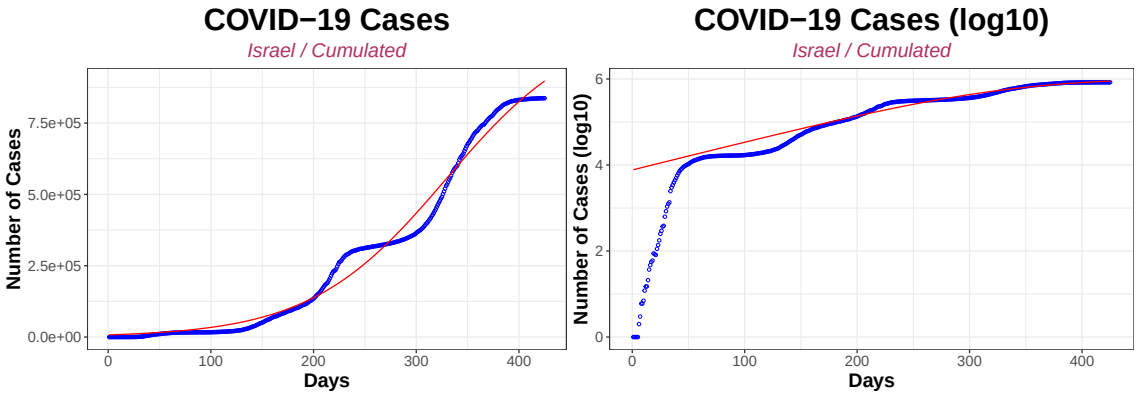
```
coef(gn_fit_logit) # coefficients
```

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

```
deviance(gn_fit_logit) # SSR
```

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

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

베르탈란피 모델은 아래와 같은 식으로 회귀분석을 하면 된다.

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

```
form_bert <- Cases ~ A * (1 - exp(beta0 - beta1 * Days))
grid_bert <- data.frame(A = c(0, max(ISR$Cases)), beta0 = c(0, 1), beta1 = c(0, 0.1))
rough_fit_bert <- nls2(form_bert, data = ISR, start = grid_bert, algorithm = "brute-force")
summary(rough_fit_bert)
```

```
##
## Formula: Cases ~ A * (1 - exp(beta0 - beta1 * Days))
##
## Parameters:
```

```
##         Estimate Std. Error t value Pr(>|t|)
## A     2.790e+05  1.560e+04  17.885  < 2e-16 ***
## beta0 6.667e-01  1.980e-01   3.367 0.000829 ***
## beta1 3.333e-02  9.601e-03   3.472 0.000571 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 272400 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) # Error
```

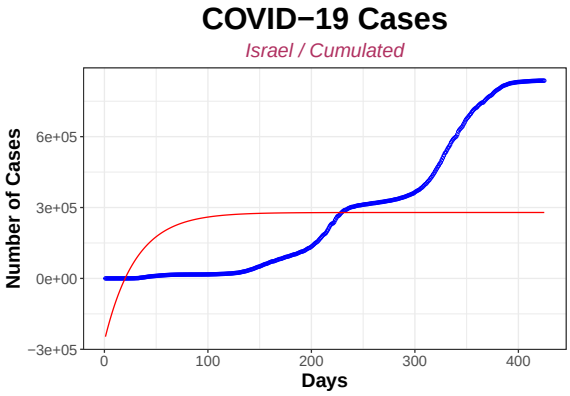
```
## Error in (function (formula, data = parent.frame(), start, control = nls.control(), : singular gradi
```

```
coef(rough_fit_logit) # coefficients
```

```
##            A        beta0        beta1
## 5.580313e+05 6.666667e+01 3.333333e-01
```

```
deviance(rough_fit_logit) # SSR
```

```
## [1] 1.231473e+13
```



베르탈란피 모델을 가지고 fitting을 했을 때, 확진자 수를 전혀 예측하지 못하고 있다. 이는 베르탈란피 모델의 개형과 확진자 수의 개형이 완전히 달라서 생길 수 밖에 없다고 추측한다. 총 확진자 수의 증가량이 점점 증가하는 모습을 보이고 있지만, 베르탈란피 모델은 인구의 증가량은 최대 인구와 현재 인구의 차이에 비례하므로, 전혀 다른 결과를 나타냄을 알 수 있다.

Gompertz Model

곰파츠 모델은 다음과 같다. a * exp(b * (-1) * exp((-1)cDays_after_Start))

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

```
form_gomp <- Cases ~ A * exp(-exp(beta0 - beta1 * Days/10))
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/10))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## A     2.790e+05  2.149e+04  12.987   <2e-16 ***
## beta0 1.000e+00  4.977e-01   2.009   0.0452 *
## beta1 1.667e-01  6.574e-02   2.535   0.0116 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 253300 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/10))
##
## 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-02  2.480e-03   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: 14
## Achieved convergence tolerance: 3.039e-06
```

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

```
##            A        beta0        beta1
## 1.847074e+06 2.091123e+00 5.764307e-02
```

```
deviance(gn_fit_gomp) # SSR
```

```
## [1] 512747252091
```

fitting의 결과를 그림으로 표현하면 아래와 같다.