회귀분석을 통한 COVID-19 Confirmed Cases의 분석 및 예측

러시아의 Confirmed Cases에 대하여

서울대학교 통계학과 2017-11362 박건도

2021년 04월 29일

0. 라이브러리 설정 및 기초 함수 정의

라이브러리

```
library(nls2)
library(segmented)
library(dplyr)
library(tidyr)
library(ggplot2)
set.seed(0)
```

기초 함수

SSE는 `deviance` 함수를 통해 구할 수 있으나 MSE는 따로 제공되지 않기 때문에, 아래의 식으로 알아낸다. 결정계수 R^2 또한 제공되지 않는 값이므로 다음의 식을 이용하여 구한다.

```
getMSE <- function(y, yhat, df){ # MSE = SSE / df
  sum((y - yhat)^2) / df
}
getRsq <- function(y, model){ # R^2 = SSR / SST = 1 - SSE / SST
  1 - deviance(model) / sum((y - mean(y))^2)
}
```

1. 누적 확진자 수 예측

데이터 전처리

주어진 `.csv` 파일을 불러와 구조를 살펴보면 아래와 같다. 이중, 분석해야 하는 러시아의 자료만 필터링하여 `RUS`를 만들었다.

```
covid <- read.csv('./global_confirmed_cases_210420.csv')
RUS <- covid %>%
  filter(CountryCode == 'RUS', Cases > 0) %>%
  mutate(Date = as.character(Date)) %>%
  select(Date, Cases, Days)
str(RUS)
```

```
'data.frame':   446 obs. of  3 variables:
 $ Date : chr  "2020.1.31" "2020.2.1" "2020.2.2" "2020.2.3" ...
 $ Cases: int  2 2 2 2 2 2 2 2 2 2 ...
 $ Days : int  1 2 3 4 5 6 7 8 9 10 ...
```

Date의 column에 들어있는 값들에 일관성을 주기 위해, "yyyy-mm-dd" 의 형식으로 바꿔주었다. 앞의 6개 데이터를 보면, 데이터의 전처리가 잘 이뤄진 것을 볼 수 있다.

```
RUS$Date <- RUS$Date %>%
  sapply(function(d) {
    vec <- strsplit(d, split = "\\.") %>% unlist
    vec_ch <- sapply(vec, function(i) {
      ifelse(nchar(i) == 1, paste0("0", i), i)
    })
    paste(vec_ch, collapse = "-")
  })
start_date <- RUS$Date[1]
head(RUS)
```

Date	Cases	Days
2020-01-31	2	1
2020-02-01	2	2
2020-02-02	2	3
2020-02-03	2	4
2020-02-04	2	5
2020-02-05	2	6

우리는 3/20일 까지의 데이터로 trainning을 한 후, 3/21 ~ 4/20일 까지의 데이터로 예측한 값을 비교할 예정이므로 훈련 데이터를 모아 놓은 RUS_train과 예측할 데이터를 모아 놓은 RUS_predict를 만들었다.

```
RUS_train <- RUS %>% filter(as.Date(Date) <= as.Date("2021-03-20"))
RUS_predict <- RUS %>% filter(as.Date(Date) > as.Date("2021-03-20"))
tail(RUS_train)
```

	Date	Cases	Days
410	2021-03-15	4350728	410
411	2021-03-16	4360033	411
412	2021-03-17	4368943	412
413	2021-03-18	4378656	413
414	2021-03-19	4388268	414
415	2021-03-20	4397816	415

```
head(RUS_predict)
```

Date	Cases	Days
2021-03-21	4407031	416
2021-03-22	4416226	417
2021-03-23	4424595	418
2021-03-24	4433364	419
2021-03-25	4442492	420
2021-03-26	4451565	421

RUS_train의 마지막 부분 행과 RUS_predict의 처음 부분 행을 보면, 잘 분리가 된 것을 알 수 있다.

(1) 모델 fitting 및 누적 확진자 수 예측

(i) Linear regression

위에서 처리한 데이터를 가지고 아래의 식과 같이 선형 회귀 분석을 실시해 보자.

$$\mathrm{y} = \beta_0 + \beta_1 \mathrm{x}$$

여기서 y는 Cases가 되고, x는 Days가 된다. 위 식과 lm 함수를 이용하여 LSE를 구한다.

```
fit_lin <- lm(Cases ~ Days, data = RUS_train)
summary(fit_lin)
```

```
Call:
lm(formula = Cases ~ Days, data = RUS_train)

Residuals:
    Min      1Q  Median      3Q     Max
-686315 -383734  -56692  431045  858427
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -869572.4    44815.5   -19.4   <2e-16 ***
Days          11147.3      186.7    59.7   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 455700 on 413 degrees of freedom
Multiple R-squared:  0.8962,    Adjusted R-squared:  0.8959
F-statistic:  3565 on 1 and 413 DF,  p-value: < 2.2e-16
```

```
Rsq_lin <- getRsq(RUS_train$Cases, fit_lin)
Rsq_lin
```

```
[1] 0.896172
```

선형 회귀 모델 적합 결과, y절편과 기울기 모두 p-value가 2e-16 이하로 유의함을 알 수 있고, R^2 의 값은 0.8962가 나왔다. 계수는 뒤에서 한 번에 정리하였다.

(ii) Logistic model

Logistic 모델은 뒤에서 나올 Gompertz 모델과 마찬가지로 `nls` 함수를 사용하여 비선형 회귀분석을 실시한다. 적합할 Logistic 모델의 식은 아래와 같다.

$$y = \frac{a}{1 + e^{b - cx}}$$

```
form_logit <- Cases ~ a / (1 + exp(b - c * Days))
grid_logit <- data.frame(a = c(0, max(RUS_train$Cases)),
                         b = c(0, 5), c = c(0, 1))
rough_fit_logit <- nls2(form_logit, data = RUS_train,
                        start = grid_logit, alg = "brute-force")
gn_fit_logit <- nls2(form_logit, data = RUS_train, start = rough_fit_logit)
summary(gn_fit_logit)
```

```
Formula: Cases ~ a/(1 + exp(b - c * Days))

Parameters:
   Estimate Std. Error t value Pr(>|t|)
a 6.550e+06  1.610e+05   40.69   <2e-16 ***
b 4.610e+00  3.735e-02  123.44   <2e-16 ***
c 1.322e-02  2.293e-04   57.64   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 131200 on 412 degrees of freedom

Number of iterations to convergence: 15
Achieved convergence tolerance: 4.041e-06
```

```
Rsq_logit <- getRsq(RUS_train$Cases, gn_fit_logit)
Rsq_logit
```

```
[1] 0.9914153
```

적합 결과 세 parameters에 대해 모두 유의하게 (p-value < 2e-16) 나왔고, R^2 의 값은 0.9914가 나왔다.

(iii) Gompertz Model

Gompertz 모델은 다음과 같다.

$$y = ae^{-be^{-cx}}$$

이 모델도 위의 모델과 마찬가지로 LSE를 구한다.

```
form_gomp <- Cases ~ a * exp(-b * exp(- c * Days))
grid_gomp <- data.frame(a = c(0, max(RUS_train$Cases)),
                        b = c(1, 10),
                        c = c(0, 0.01))
rough_fit_gomp <- nls2(form_gomp, data = RUS_train,
                       start = grid_gomp, algorithm = "brute-force")
gn_fit_gomp <- nls2(form_gomp, data = RUS_train, start = rough_fit_gomp)
summary(gn_fit_gomp)
```

```
Formula: Cases ~ a * exp(-b * exp(-c * Days))

Parameters:
   Estimate Std. Error t value Pr(>|t|)
a 1.952e+07  2.002e+06   9.751   <2e-16 ***
b 6.557e+00  6.277e-02 104.449   <2e-16 ***
c 3.697e-03  1.775e-04  20.832   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 142900 on 412 degrees of freedom
```

```
Number of iterations to convergence: 14
Achieved convergence tolerance: 5.271e-06
```

```
Rsq_gomp <- getRsq(RUS_train$Cases, gn_fit_gomp)
Rsq_gomp
```

```
[1] 0.9898103
```

Gompertz 모델 또한 계수가 모두 유의하게 나왔고, R^2 값은 0.9898이 나왔다.

(iv) 누적 확진자 수의 예측

누적 확진자 수에 대한 예측은 `predict` 함수를 사용하여 예측할 것이다.

```
df <- data.frame(Days = RUS_predict$Days)

RUS_predict <- RUS_predict %>%
  mutate(fit_lin = predict(fit_lin, df),
         fit_logit = predict(gn_fit_logit, df),
         fit_gomp = predict(gn_fit_gomp, df))
head(RUS_predict)
```

Date	Cases	Days	fit_lin	fit_logit	fit_gomp
2021-03-21	4407031	416	3767714	4640206	4773748
2021-03-22	4416226	417	3778862	4658037	4798624
2021-03-23	4424595	418	3790009	4675769	4823536
2021-03-24	4433364	419	3801156	4693400	4848486
2021-03-25	4442492	420	3812304	4710931	4873472
2021-03-26	4451565	421	3823451	4728361	4898493

이 자료를 가지고 (4) 번에서 testMSE를 구할 예정이다.

(2) 각 모델의 coefficients

각 모델의 계수를 표로 나타내면 아래와 같다.

```
model1 <- data.frame(beta0 = coef(fit_lin)[1],
                     beta1 = coef(fit_lin)[2],
                     R.square = Rsq_lin)
rownames(model1) <- "Linear regression"
model1
```

```
model2 <- rbind(coef(gn_fit_logit), coef(gn_fit_gomp)) %>%
  cbind(c(Rsq_logit, Rsq_gomp))
rownames(model2) <- c("Logistic model", "Gompertz model")
colnames(model2)[4] <- "R.square"
as.data.frame(model2)
```

	beta0	beta1	R.square
Linear regression	-869572.4	11147.32	0.896172

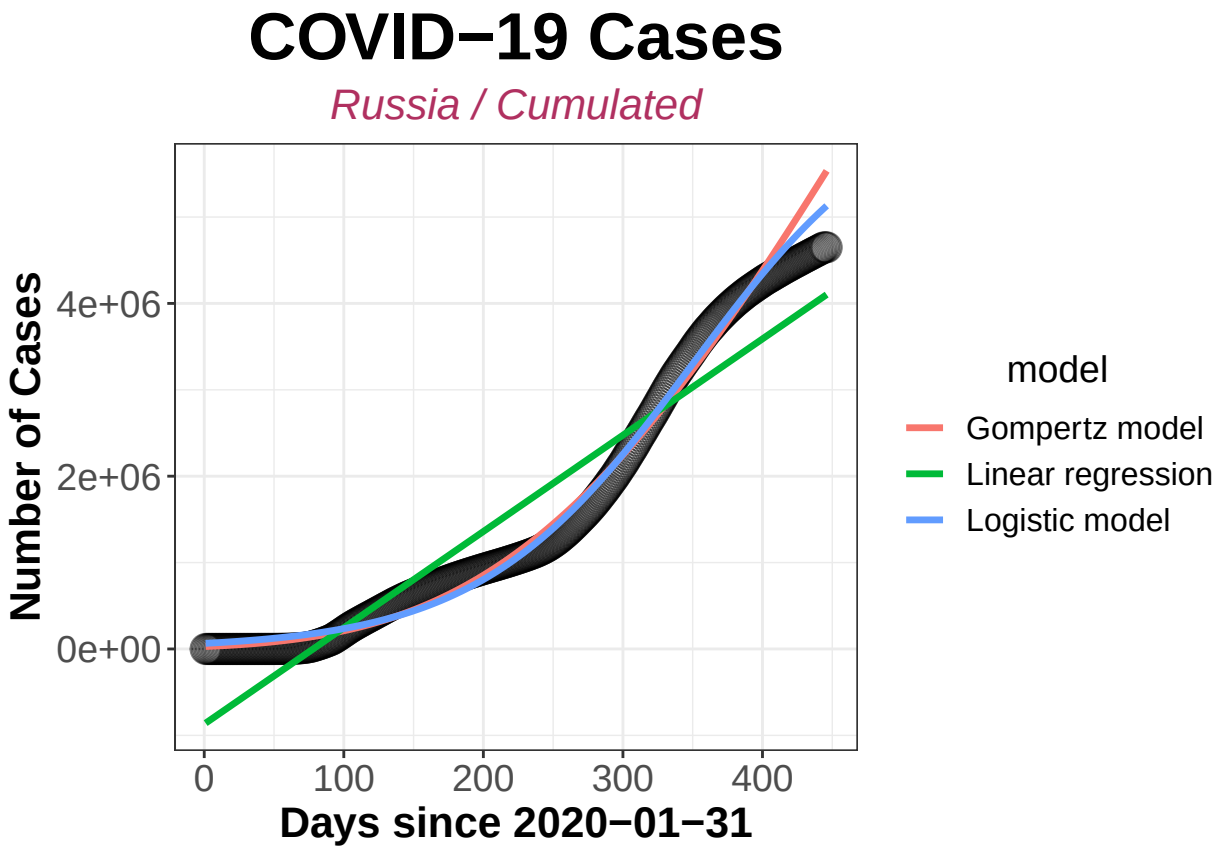
	a	b	c	R.square
Logistic model	6549664	4.610479	0.0132174	0.9914153
Gompertz model	19521207	6.556554	0.0036972	0.9898103

(3) 예측값과 실측값 그래프

여기서는 RUS_train으로 예측된 계수들로 전체 기간에 대해 그래프를 그려볼 것이다.

```
timeline <- data.frame(Days = RUS$Days)

RUS_predict_overall <- RUS %>%
  mutate(y_lin = predict(fit_lin, timeline),
         y_logit = predict(gn_fit_logit, timeline),
         y_gomp = predict(gn_fit_gomp, timeline)) %>%
  gather(model, Cases, y_lin, y_logit, y_gomp) %>%
  mutate(model = ifelse(model == "y_lin", "Linear regression",
                        ifelse(model == "y_logit",
                               "Logistic model",
                               "Gompertz model")))

p1 <- ggplot(data = RUS, aes(x = Days, y = Cases)) +
  geom_point(color = 'black', shape = 1, size = 5, alpha = 0.5) +
  geom_line(data = RUS_predict_overall,
            aes(x = Days, y = Cases, color = model),
            size = 1.2) +
  labs(title = "COVID-19 Cases",
       subtitle = paste("Russia", "/", "Cumulated"),
       x = paste('Days since', start_date),
```

```
       y = 'Number of Cases') +
  theme_bw() +
  theme(plot.title = element_text(size = 25, hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(size = 16, hjust = 0.5,
                                     face = "italic", color = "maroon"),
        axis.title = element_text(size = 16, face = "bold"),
        axis.text = element_text(size = 14),
        legend.title = element_text(size = 14, hjust = 0.5),
        legend.text = element_text(size = 12)) +
  scale_y_continuous(labels = scales::label_scientific())

p1
```



(4) test MSE

이제 적합된 모형이 얼마나 예측을 잘 하는지 알아보자.

```
testMSE_lin <- getMSE(RUS_predict$Cases, RUS_predict$fit_lin, 31 - 2)
testMSE_logit <- getMSE(RUS_predict$Cases, RUS_predict$fit_logit, 31 - 3)
testMSE_gomp <- getMSE(RUS_predict$Cases, RUS_predict$fit_gomp, 31 - 3)
```

```
testMSE <- data.frame(test.MSE = c(testMSE_lin, testMSE_logit, testMSE_gomp))
rownames(testMSE) <- c("Linear regression", "Logistic model", "Gompertz model")
testMSE
```

	test.MSE
Linear regression	387451257081
Logistic model	145992589188
Gompertz model	444601150031

(5) 적합성 판단

위 표를 보면, Logistic 모델의 경우 test MSE의 값이 가장 작다. 또한, (2) 번의 표에서 결정계수 R^2 의 값 또한 Logistic 모델이 가장 1에 가깝다. 따라서, test MSE의 값이 가장 작은 Logistic 모델이 러시아의 COVID-19 확진자 수를 잘 적합한다고 할 수 있다.

2. segmented 일별 확진자 수 예측

이제는 백신 접종관 관련된 데이터를 가지고 분석해보자. 우선, 2번과 3번 문제를 위한 dataset인 `RUS_vac`을 만든다.

데이터 전처리

```
vaccine <- read.csv('covid_vaccine.csv')

RUS_vac <- vaccine %>%
  filter(CountryCode == 'RUS', Cases > 0) %>%
  mutate(Date = as.character(Date)) %>%
  select(Date, Difference, people_vaccinated)

RUS_vac$Date <- RUS_vac$Date %>%
  sapply(function(d) {
    vec <- strsplit(d, split = "\\.") %>% unlist
    vec_ch <- sapply(vec, function(i) {
      ifelse(nchar(i) == 1, paste0("0", i), i)
    })
    paste(vec_ch, collapse = "-")
  })
start_date <- RUS_vac$Date[1]
RUS_vac <- RUS_vac %>%
```

```
  mutate(Days_after_Start = as.integer(as.Date(Date) - as.Date(start_date)))
RUS_vac_train <- RUS_vac %>%
  filter(as.Date(Date) <= as.Date("2021-03-20"))
head(RUS_vac)
```

Date	Difference	people_vaccinated	Days_after_Start
2020-12-15	26265	28500	0
2020-12-16	26074	28500	1
2020-12-17	27787	28500	2
2020-12-18	28116	28500	3
2020-12-19	27772	28500	4
2020-12-20	28510	28500	5

```
tail(RUS_vac_train)
```

	Date	Difference	people_vaccinated	Days_after_Start
91	2021-03-15	9347	5366009	90
92	2021-03-16	9305	5366009	91
93	2021-03-17	8910	5379359	92
94	2021-03-18	9713	5431048	93
95	2021-03-19	9612	5544777	94
96	2021-03-20	9548	5595893	95

(1) 예측 및 시각화

```
timeline_vac <- data.frame(Days_after_Start = RUS_vac$Days_after_Start,
                           people_vaccinated = RUS_vac$people_vaccinated)
fit_1 <- glm(Difference ~ log(Days_after_Start+1) + Days_after_Start,
             data = RUS_vac_train,
             family = poisson)
seg_fit_1 <- segmented(fit_1, seg.Z = ~ Days_after_Start, npsi = 2,
                       control = seg.control(it.max = 10000, n.boot = 50))
```

```
Warning: max number of iterations (1) attained
```

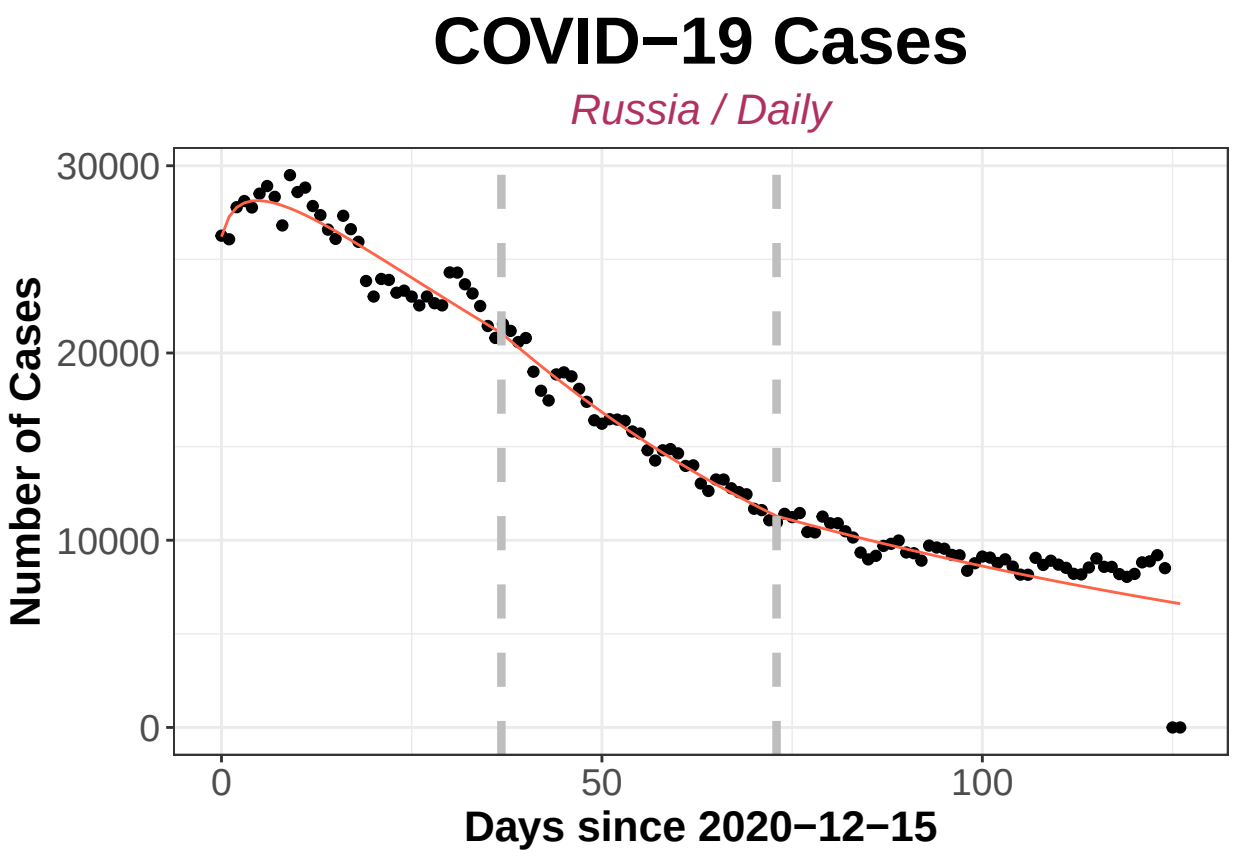
```
summary(seg_fit_1)
```

```
    ***Regression Model with Segmented Relationship(s)***
```

```
Call:
segmented.glm(obj = fit_1, seg.Z = ~Days_after_Start, npsi = 2,
    control = seg.control(it.max = 10000, n.boot = 50))

Estimated Break-Point(s):
                         Est. St.Err
psi1.Days_after_Start 36.787  0.686
psi2.Days_after_Start 72.952  0.656

Meaningful coefficients of the linear terms:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)               10.1738349  0.0044593 2281.47   <2e-16 ***
log(Days_after_Start + 1)  0.0776040  0.0029640   26.18   <2e-16 ***
Days_after_Start          -0.0135988  0.0002421  -56.18   <2e-16 ***
U1.Days_after_Start       -0.0050923  0.0002296  -22.18       NA
U2.Days_after_Start        0.0078140  0.0003579   21.84       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null     deviance: 231522.5  on 95  degrees of freedom
Residual deviance:   2629.6  on 89  degrees of freedom
AIC: 3754.7

Convergence attained in 6 iter. (rel. change 1.7562e-06)
```

```r
df_predict_1 <- data.frame(Date = RUS_vac$Date,
                           Days = RUS_vac$Days_after_Start,
                           Cases = RUS_vac$Difference,
                           Predict = exp(predict(seg_fit_1, timeline_vac)))
psi_1 <- seg_fit_1$psi[1, 2]
psi_2 <- seg_fit_1$psi[2, 2]
p_2 <- ggplot(df_predict_1, aes(x = Days, y = Cases)) +
    geom_point() +
    geom_line(mapping = aes(x = Days, y = Predict),
              data = df_predict_1, color = "tomato") +
    geom_vline(xintercept = psi_1, color = "grey",
               size = 1.5, linetype = "dashed") +
    geom_vline(xintercept = psi_2, color = "grey",
```

```
              size = 1.5, linetype = "dashed") +
  labs(title = "COVID-19 Cases",
       subtitle = paste("Russia", "/", "Daily"),
       x = paste('Days since', start_date),
       y = 'Number of Cases') +
  theme_bw() +
  theme(plot.title = element_text(size = 25, hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(size = 16, hjust = 0.5,
                                     face = "italic", color = "maroon"),
        axis.title = element_text(size = 16, face = "bold"),
        axis.text = element_text(size = 14))

p_2
```



(2) test MSE

위에서 구한 segmented poisson regression 모델을 통해 구한 3/21 ~ 4/20일 까지의 일별 예측값을 이용하여 test MSE를 구해보자.

```
RUS_predict_1 <- df_predict_1 %>%
  filter(as.Date(Date) >= as.Date("2021-03-21"))
testMSE_seg_1 <- getMSE(RUS_predict_1$Cases, RUS_predict_1$Predict, 31 - 6)
testMSE_seg_1 # test MSE
```

```
[1] 4851931
```

3. 백신 접종 변수를 포함한 segmented 분석

(1). 예측 및 시각화

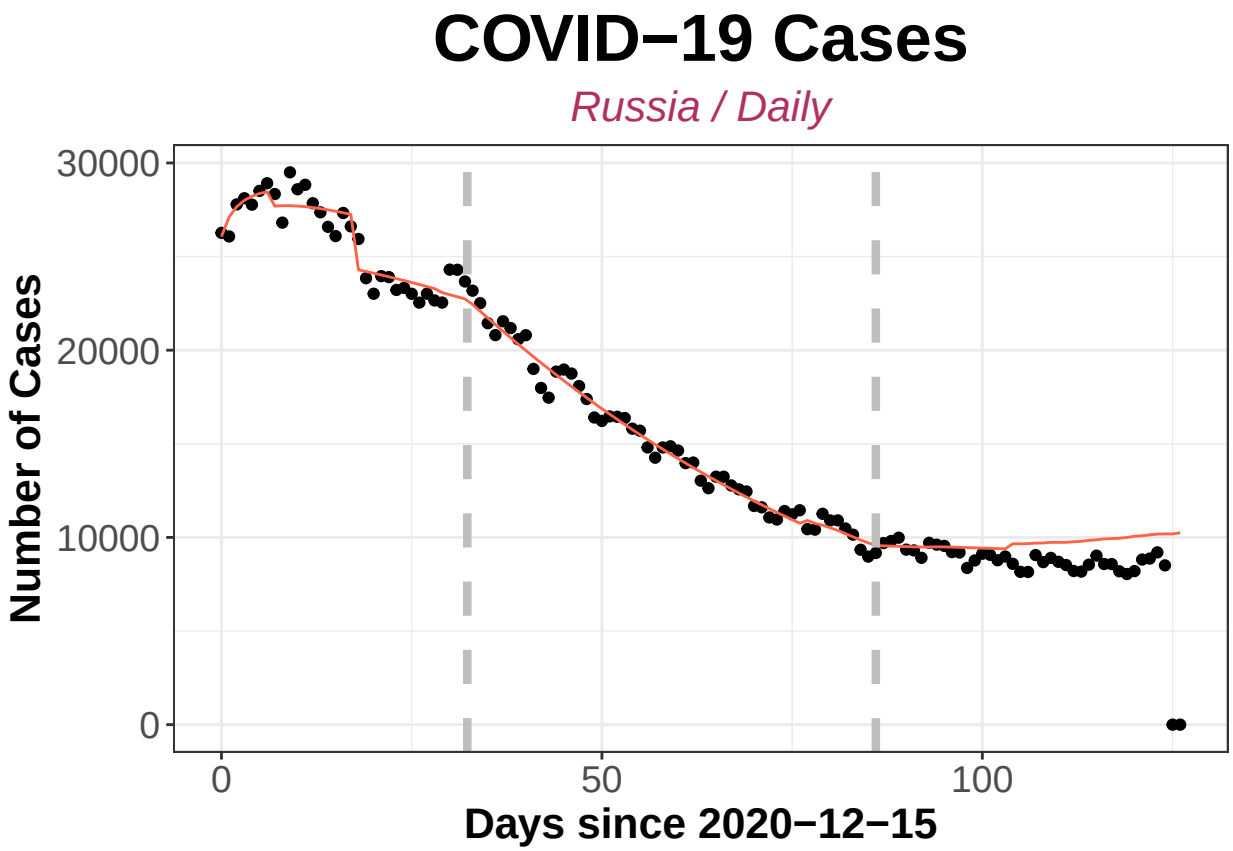
여기서는 2번의 (1) 에서 사용한 코드에다가 predictor에 people_vaccinated 변수를 추가하여 적합할 것이다.

```
fit_2 <- glm(Difference ~ log(Days_after_Start + 1) +
               log(people_vaccinated + 1) +
               Days_after_Start + people_vaccinated,
             data = RUS_vac_train,
             family = poisson)
seg_fit_2 <- segmented(fit_2, seg.Z = ~ Days_after_Start, npsi = 2,
                       control = seg.control(it.max = 10000, n.boot = 50))
summary(seg_fit_2)
```

```
    ***Regression Model with Segmented Relationship(s)***

Call:
segmented.glm(obj = fit_2, seg.Z = ~Days_after_Start, npsi = 2,
    control = seg.control(it.max = 10000, n.boot = 50))

Estimated Break-Point(s):
                         Est. St.Err
psi1.Days_after_Start 32.304  0.317
psi2.Days_after_Start 85.998  0.434

Meaningful coefficients of the linear terms:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                 1.068e+01  2.139e-02  499.30   <2e-16 ***
log(Days_after_Start + 1)   6.664e-02  3.336e-03   19.98   <2e-16 ***
log(people_vaccinated + 1) -4.972e-02  2.041e-03  -24.36   <2e-16 ***
Days_after_Start           -7.013e-03  4.422e-04  -15.86   <2e-16 ***
people_vaccinated           3.239e-08  1.516e-09   21.37   <2e-16 ***
```

```
U1.Days_after_Start        -1.136e-02  4.046e-04  -28.08       NA
U2.Days_after_Start         1.626e-02  1.132e-03   14.36       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null     deviance: 231522.5  on 95  degrees of freedom
Residual deviance:   1920.5  on 87  degrees of freedom
AIC: 3049.6

Convergence attained in 5 iter. (rel. change 5.3118e-06)
```

```r
df_predict_2 <- data.frame(Date = RUS_vac$Date,
                           Days = RUS_vac$Days_after_Start,
                           Cases = RUS_vac$Difference,
                           Predict = exp(predict(seg_fit_2, timeline_vac)))
psi_3 <- seg_fit_2$psi[1, 2]
psi_4 <- seg_fit_2$psi[2, 2]
p_3 <- ggplot(df_predict_2, aes(x = Days, y = Cases)) +
    geom_point() +
    geom_line(mapping = aes(x = Days, y = Predict),
              data = df_predict_2, color = "tomato") +
    geom_vline(xintercept = psi_3, color = "grey",
               size = 1.5, linetype = "dashed") +
    geom_vline(xintercept = psi_4, color = "grey",
               size = 1.5, linetype = "dashed") +
  labs(title = "COVID-19 Cases",
       subtitle = paste("Russia", "/", "Daily"),
       x = paste('Days since', start_date),
       y = 'Number of Cases') +
  theme_bw() +
  theme(plot.title = element_text(size = 25, hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(size = 16, hjust = 0.5,
                                     face = "italic", color = "maroon"),
        axis.title = element_text(size = 16, face = "bold"),
        axis.text = element_text(size = 14))

p_3
```



(2). test MSE

마찬가지로 segmented poisson regression 모델을 통해 구한 3/21 ~ 4/20일 까지의 일별 예측값을 이용하여 test MSE를 구해보자.

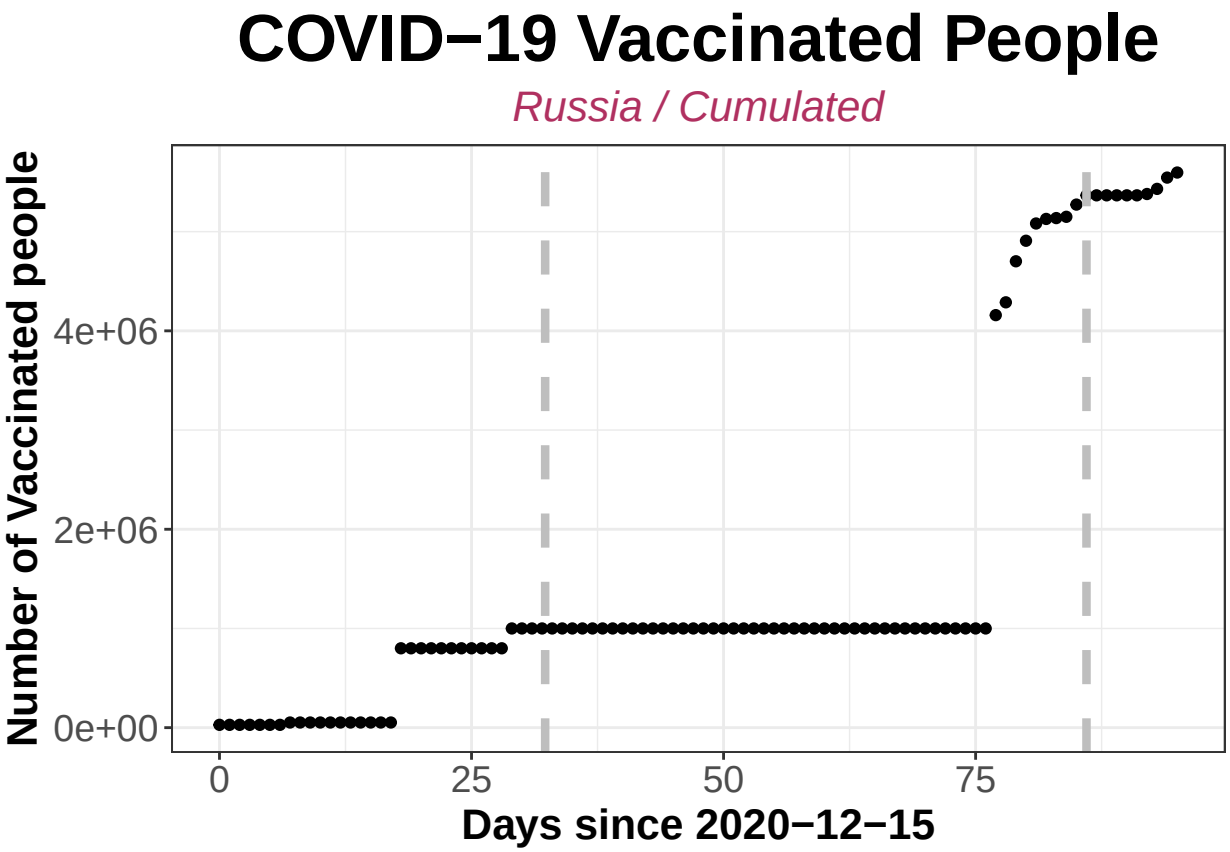
```
RUS_predict_2 <- df_predict_2 %>%
  filter(as.Date(Date) >= as.Date("2021-03-21"))
testMSE_seg_2 <- getMSE(RUS_predict_2$Cases, RUS_predict_2$Predict, 31 - 9)
testMSE_seg_2 # test MSE
```

```
[1] 11372929
```

(3). 백신의 효과

백신 접종자에 대한 정보를 넣지 않았을 때와 넣었을 때의 test MSE 값을 비교해 보면, 4851931 에서 11372929로 꽤 많이 증가한 것으로 보인다. 이는 일별 확진자 수의 모델이 푸아송 분포를 따른다고 가정했을 때, 백신 접종자에 대한 정보가 무의미하게 일별 확진자 수에 영향을 끼침을 알 수 있다.
우리가 처음 구했던 `seg_fit_1`은 백신 접종자에 대한 정보 없이 시간에 따른 일별 확진자 수를 적합한 것이다. 따라서 이는, 백신을 맞지 않았다고 가정한 후 적합한 것이라고 할 수 있다. 반면 `seg_fit_2`에서는 백신 접종자에 대한 정보를 포함하여 백신 접종을 맞았을 때를 가정하여 적합하였다. 실제로 백신 접종자에 대한 그래프를 그려보면 아래와 같다.

```r
p_4 <- ggplot(RUS_vac_train, aes(x = Days_after_Start, y = people_vaccinated)) +
  geom_point() +
    geom_vline(xintercept = psi_3, color = "grey",
               size = 1.5, linetype = "dashed") +
    geom_vline(xintercept = psi_4, color = "grey",
               size = 1.5, linetype = "dashed") +
  labs(title = "COVID-19 Vaccinated People",
       subtitle = paste("Russia", "/", "Cumulated"),
       x = paste('Days since', start_date),
       y = 'Number of Vaccinated people') +
  theme_bw() +
  theme(plot.title = element_text(size = 25, hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(size = 16, hjust = 0.5,
                                     face = "italic", color = "maroon"),
        axis.title = element_text(size = 16, face = "bold"),
        axis.text = element_text(size = 14))

p_4
```



그 결과 실제 데이터를 더 잘 예측하지 못하였고, 따라서 러시아에서의 백신 접종은 효과가 없다고 결론내릴 수 있다.