Poisson Regression

고영현 (2014-19984)

2020-07-11

1. Poisson vs Segmented Poisson Regression

(1-1) Data Preprocessing

```
corona <- read.csv("corona_ecdc.csv") %>%
  mutate(date = make_datetime(year, month, day),
         country = countriesAndTerritories) %>%
  filter(date < "2020-07-11") %>%
  select(country, date, cases, deaths) %>%
  arrange(country, date)

num <- corona %>%
    group_by(country) %>%
    mutate(total = cumsum(cases)) %>%
    filter(total > 0) %>%
    summarise(count = n())

## 국가명 통일
levels(corona$country)[which(num$country == "Dominican_Republic")] = "Dominican Republic"
levels(corona$country)[which(num$country == "South_Korea")] = "Korea, South"
levels(corona$country)[which(num$country == "Saudi_Arabia")] = "Saudi Arabia"
levels(corona$country)[which(num$country == "United_Kingdom")] = "United Kingdom"
levels(corona$country)[which(num$country == "United_States_of_America")] = "US"


## 40개 국가 간추리기
corona_40 <- corona %>%
  right_join(read.csv("data_total.csv"), by = "country") %>%
  select(1:4) %>%
  arrange(country)
```

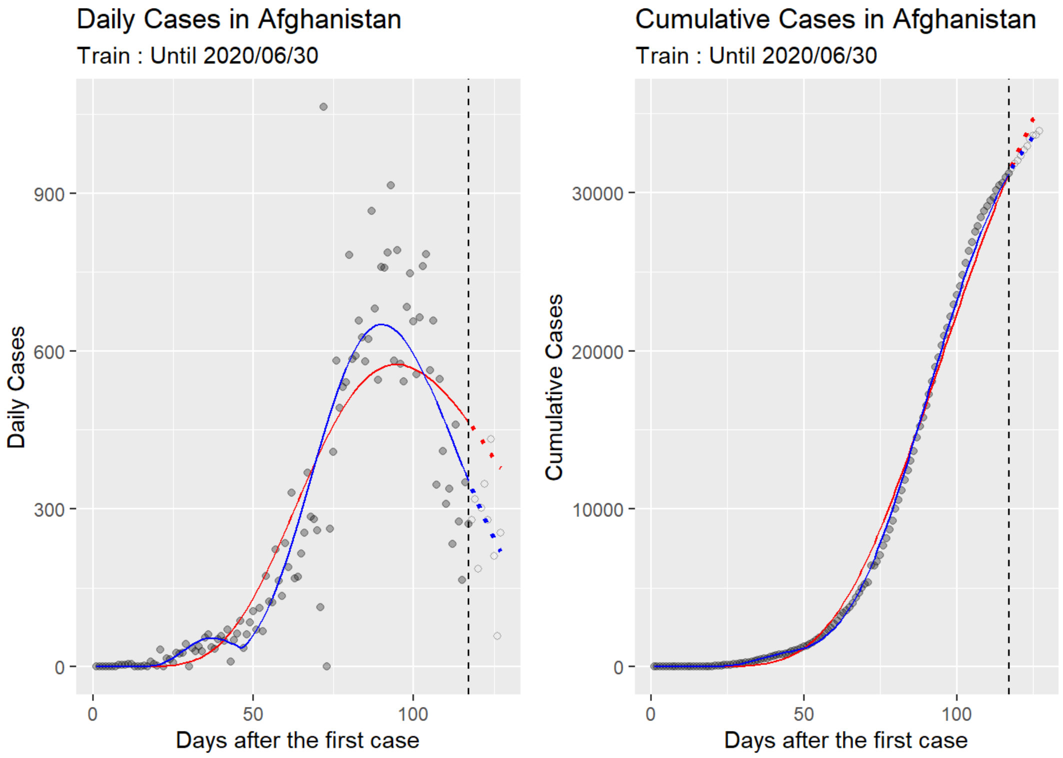
```
## Warning: Column `country` joining factors with different levels, coercing
## to character vector
```

```
num_40 <- corona_40 %>%
    group_by(country) %>%
    mutate(total = cumsum(cases)) %>%
    filter(total > 0) %>%
    summarise(count = n())
```

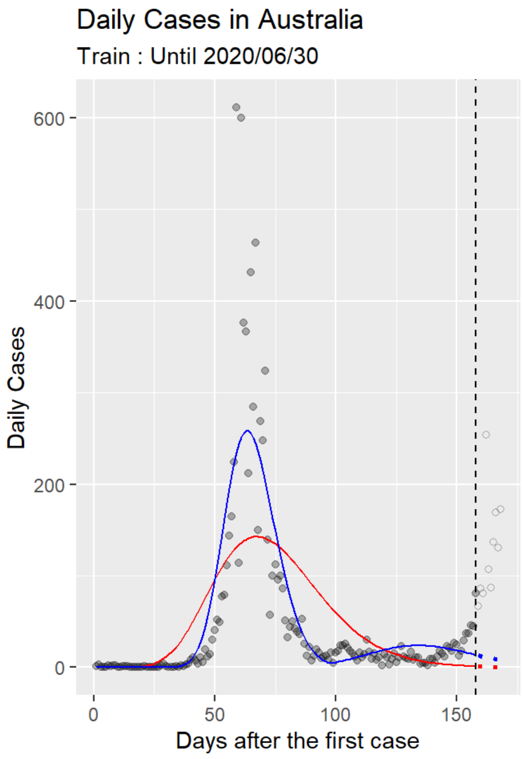
(1-2) Poisson & Segmented Poisson Regression Plot

```
#### 40개 국가 glm 및 segmented 돌리기 ####
n <- 0

set.seed(123)


for (i in unique(corona_40$country)){
  n <- n+1
  ## Case < 0 처리 (0으로 대체)
  for (j in which(corona_40$cases < 0)){corona_40$cases[j] = 0}

  ### Training set ( ~ 06/30)
  train <- corona_40 %>%
    filter(country == i) %>%
    mutate(total = cumsum(cases)) %>%
    filter(total >= 1) %>%
    mutate(Days_after_first_Case = 1:num_40$count[n]) %>%
    filter(date < "2020-07-01")


  ### Validation set (07/01 ~ 07/10)
  valid <- corona_40 %>%
    filter(country == i) %>%
    mutate(total = cumsum(cases)) %>%
    filter(total >= 1) %>%
    mutate(Days_after_first_Case = 1:num_40$count[n]) %>%
    filter(date >= "2020-07-01")


  ### Poisson & Segmented Poisson regression
  fit <- glm(cases ~ log(Days_after_first_Case) + Days_after_first_Case, data = train, family = pois
son)
  summary_fit <- summary(fit)

  seg_fit <- segmented(fit, npsi = 1)
  summary_seg_fit <- summary(seg_fit)


  ### 일일 확진자 plot
  Daily <- ggplot() +

    ## Training
    geom_point(aes(train$Days_after_first_Case, train$cases), alpha = 0.3) +
    geom_line(aes(train$Days_after_first_Case, fitted(fit)), col = 2) +
    geom_line(aes(train$Days_after_first_Case, fitted(seg_fit)), col = 4) +

    geom_vline(xintercept = nrow(train), linetype = "dashed") +

    ## Validation
    geom_point(aes(valid$Days_after_first_Case, valid$cases), shape = 1, alpha = 0.3) +
    geom_line(aes(valid$Days_after_first_Case, predict(fit, data.frame(Days_after_first_Case =
valid$Days_after_first_Case), type = "response")), col = 2, linetype = 3, size = 1) +
    geom_line(aes(valid$Days_after_first_Case, predict(seg_fit, data.frame(Days_after_first_Case = v
alid$Days_after_first_Case), type = "response")), col = 4, linetype = 3, size = 1) +

    labs(title = paste0("Daily Cases in ", i),
         subtitle = "Train : Until 2020/06/30",
         x = "Days after the first case", y = "Daily Cases")


  ### 누적 확진자 plot
  Cumulative <- ggplot() +

    ## Training
    geom_point(aes(train$Days_after_first_Case, cumsum(train$cases)), alpha = 0.3) +
    geom_line(aes(train$Days_after_first_Case, cumsum(fitted(fit))), col = 2) +
    geom_line(aes(train$Days_after_first_Case, cumsum(fitted(seg_fit))), col = 4) +

    geom_vline(xintercept = nrow(train), linetype = "dashed") +

    ## Validation
    geom_point(aes(valid$Days_after_first_Case, valid$total), shape = 1, alpha = 0.3) +
    geom_line(aes(valid$Days_after_first_Case, cumsum(predict(fit, data.frame(Days_after_first_Case
    geom_line(aes(valid$Days_after_first_Case, cumsum(predict(fit, data.frame(Days_after_first_Case
= valid$Days_after_first_Case), type = "response")) + cumsum(fitted(fit))[nrow(train)]), col = 2, li
netype = 3, size = 1) +
    geom_line(aes(valid$Days_after_first_Case, cumsum(predict(seg_fit,
data.frame(Days_after_first_Case = valid$Days_after_first_Case), type = "response")) +
cumsum(fitted(seg_fit))[nrow(train)]), col = 4, linetype = 3, size = 1) +
    labs(title = paste0("Cumulative Cases in ", i),
         subtitle = "Train : Until 2020/06/30",
         x = "Days after the first case", y = "Cumulative Cases")

  grid.arrange(Daily, Cumulative, ncol = 2)
}
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred

## Warning: glm.fit: fitted rates numerically 0 occurred
```

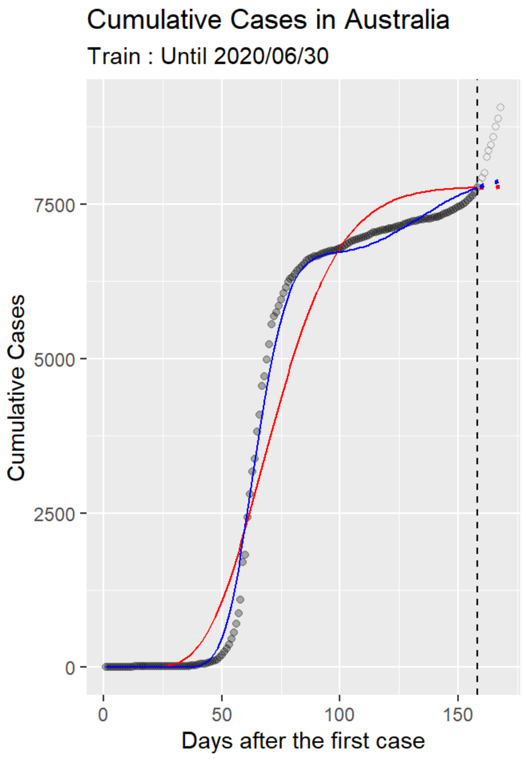


```
## Warning: glm.fit: fitted rates numerically 0 occurred
```



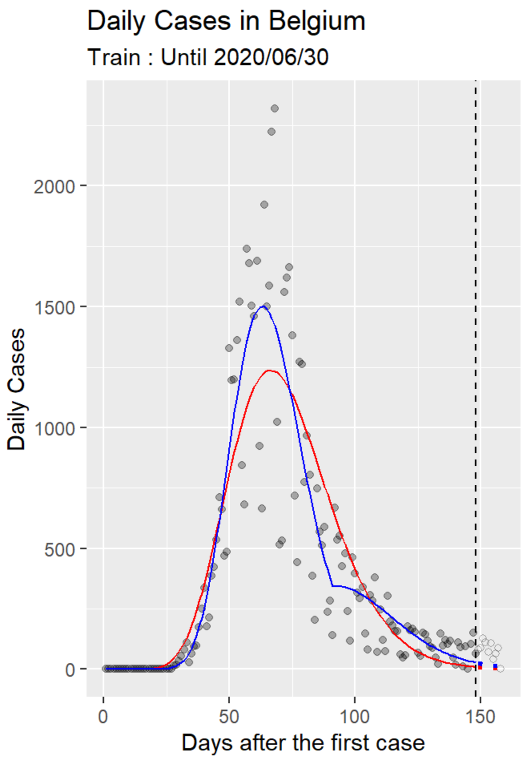
Daily Cases in Australia

Train : Until 2020/06/30



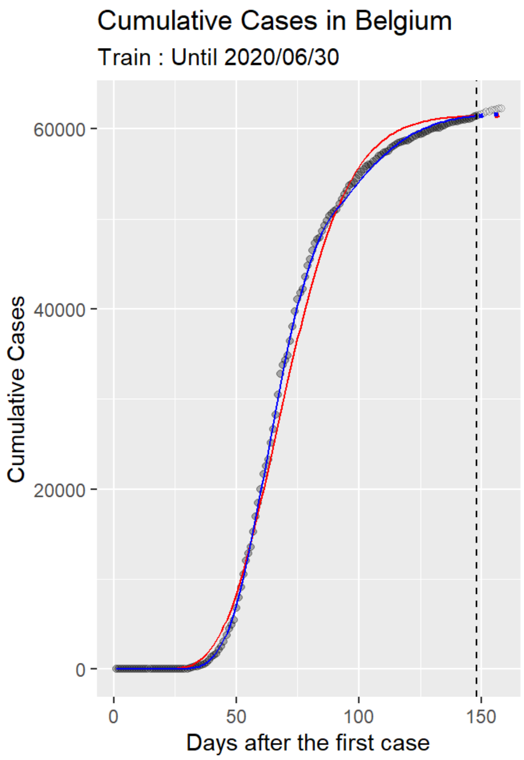
Cumulative Cases in Australia

Train : Until 2020/06/30



Daily Cases in Belgium

Train : Until 2020/06/30



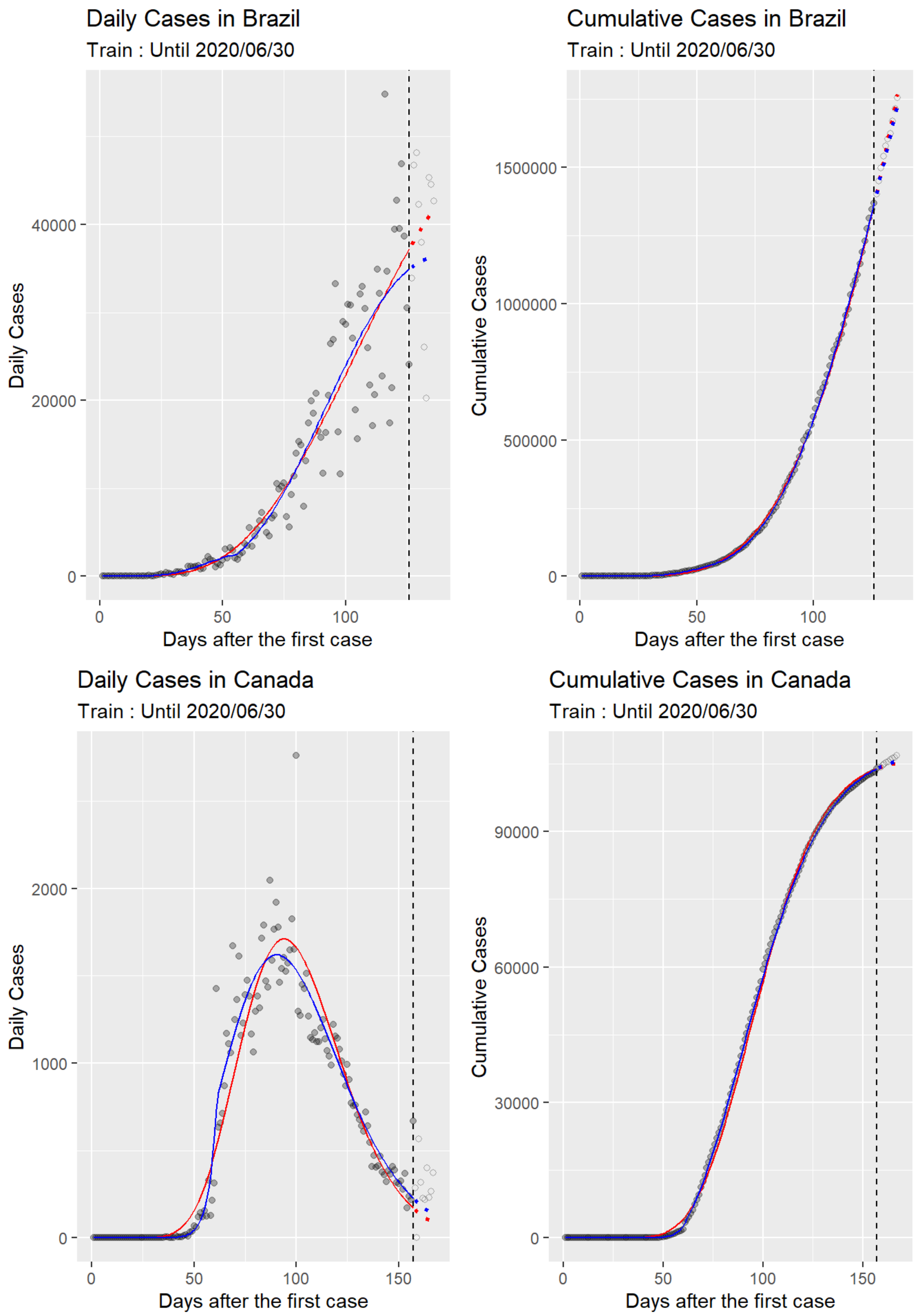
Cumulative Cases in Belgium

Train : Until 2020/06/30

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```
## Warning: max number of iterations (30) attained
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

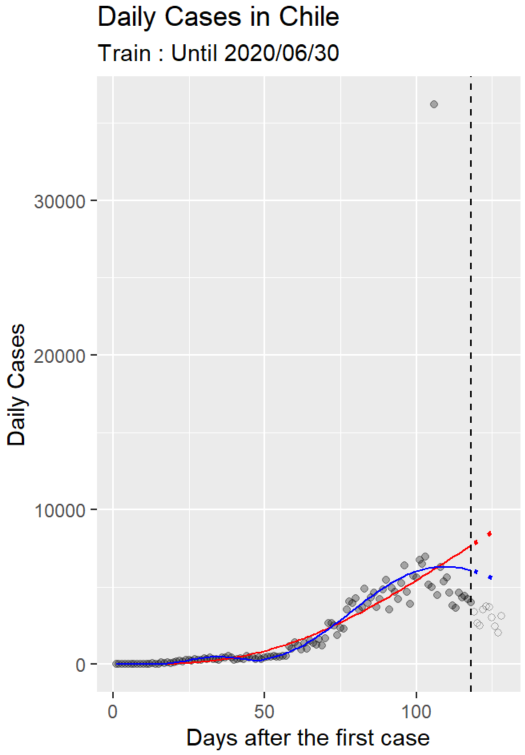


```
## Warning: max number of iterations (1) attained
```

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

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

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

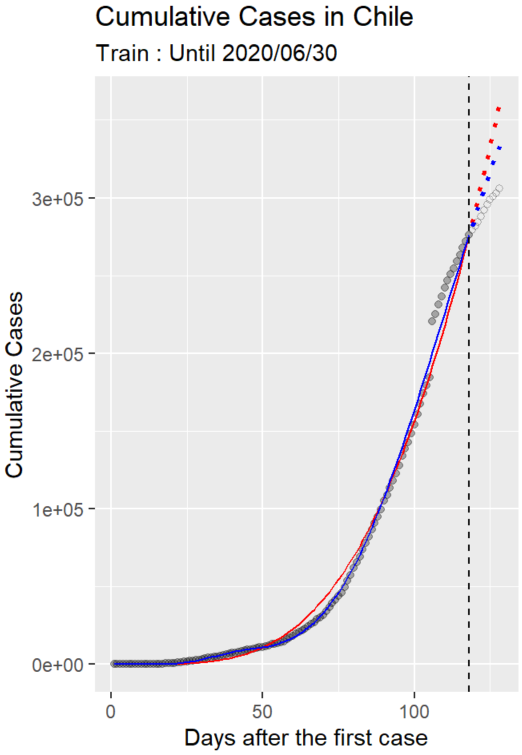
```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```



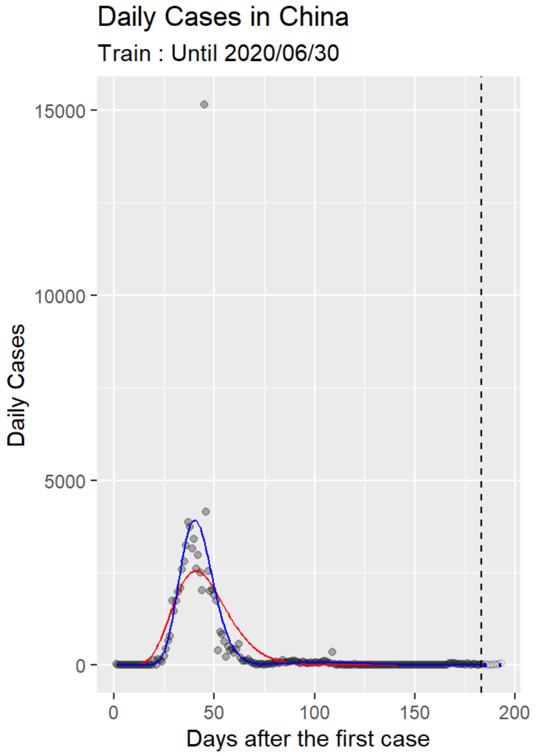
Daily Cases in Chile

Train : Until 2020/06/30



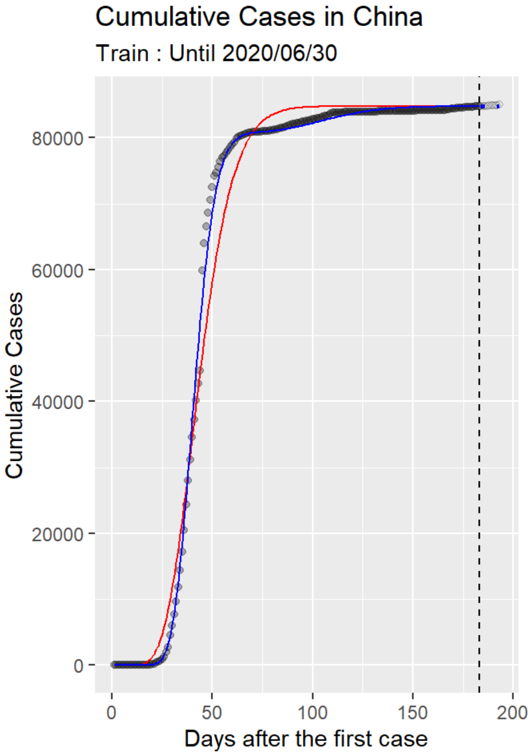
Cumulative Cases in Chile

Train : Until 2020/06/30



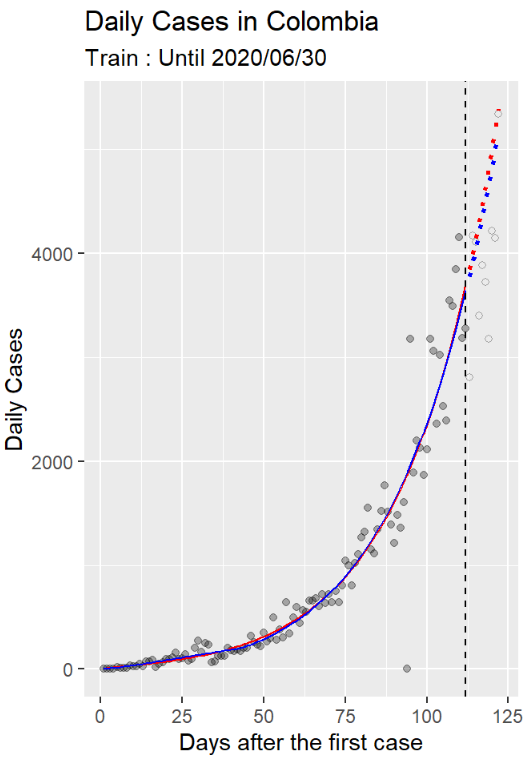
Daily Cases in China

Train : Until 2020/06/30



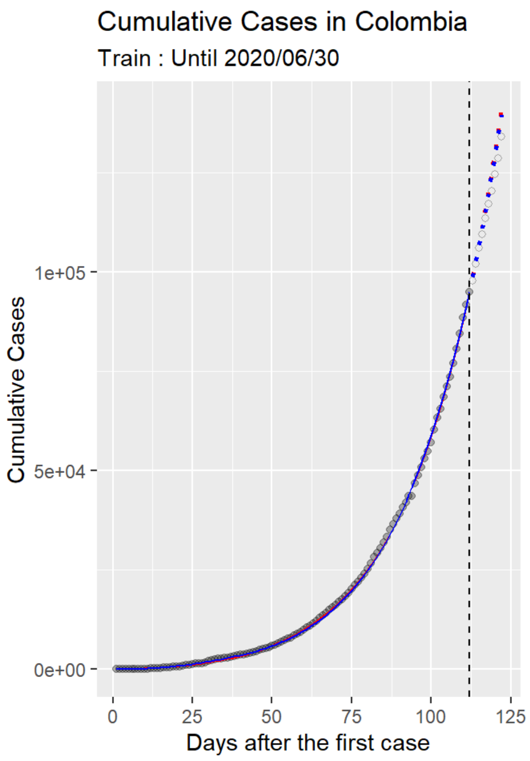
Cumulative Cases in China

Train : Until 2020/06/30



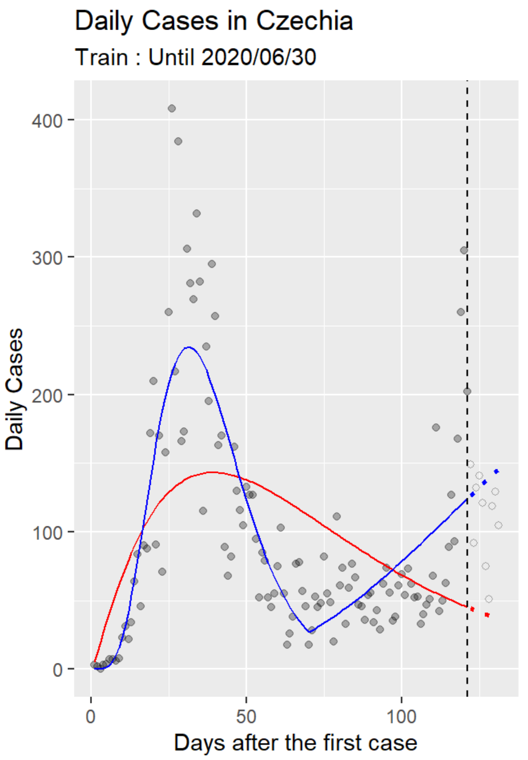
Daily Cases in Colombia

Train : Until 2020/06/30



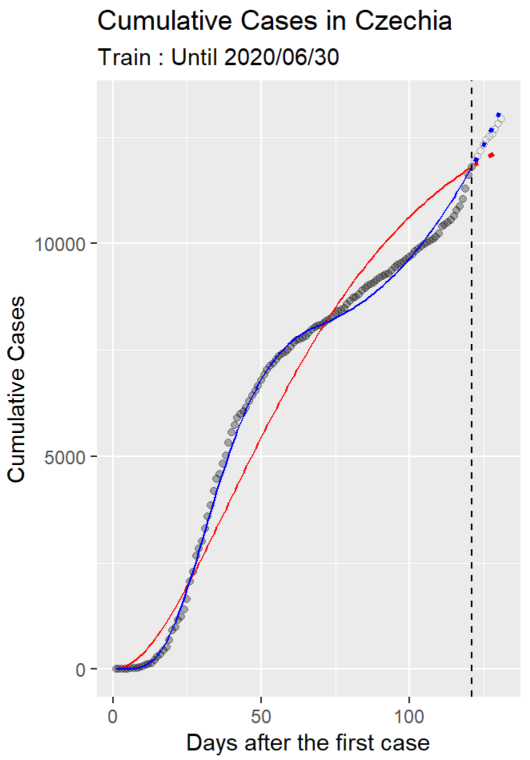
Cumulative Cases in Colombia

Train : Until 2020/06/30



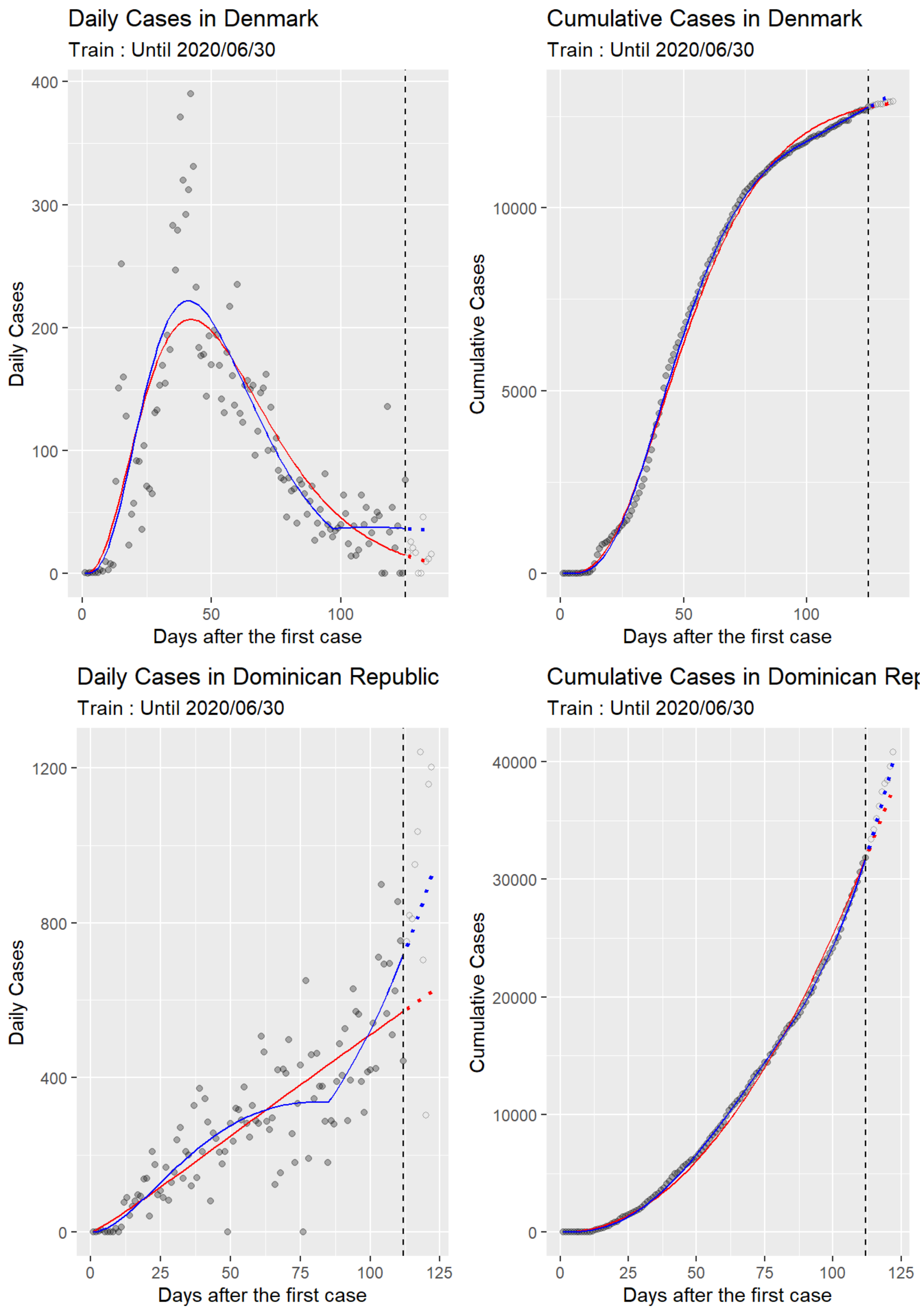
Daily Cases in Czechia

Train : Until 2020/06/30



Cumulative Cases in Czechia

Train : Until 2020/06/30

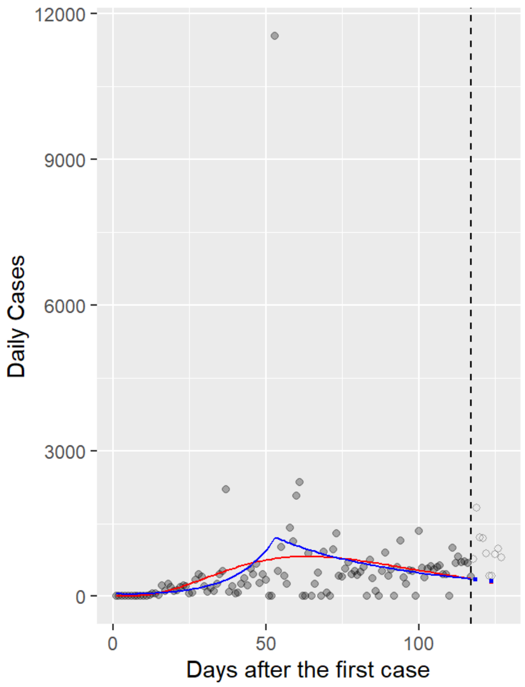


```
## Warning: max number of iterations (34) attained
```

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

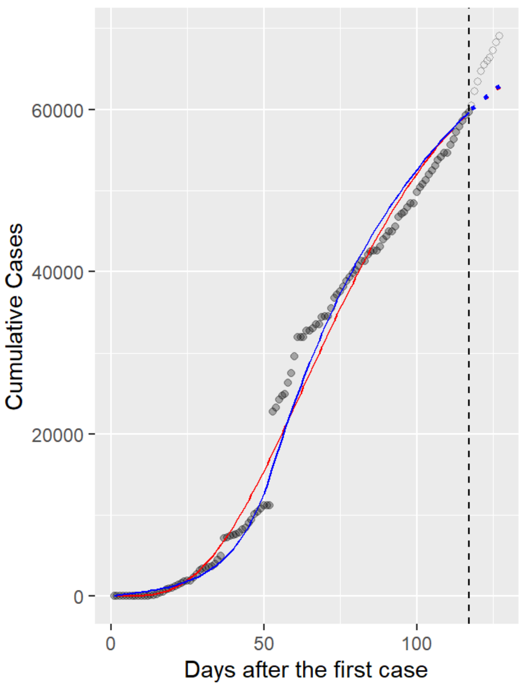
Daily Cases in Ecuador

Train : Until 2020/06/30



Cumulative Cases in Ecuador

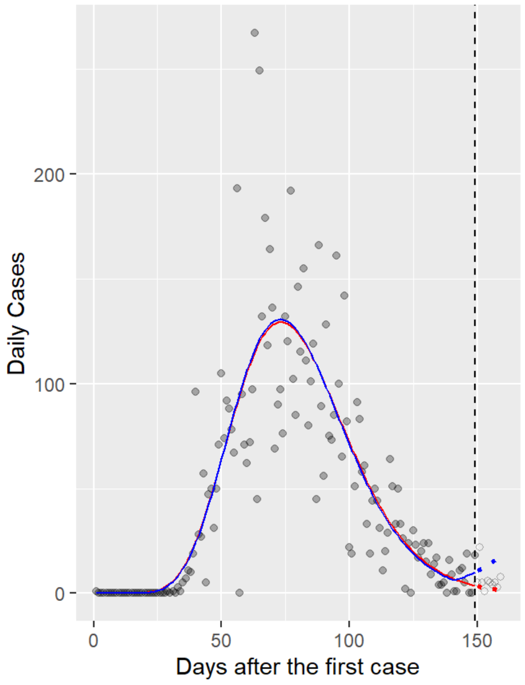
Train : Until 2020/06/30



```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

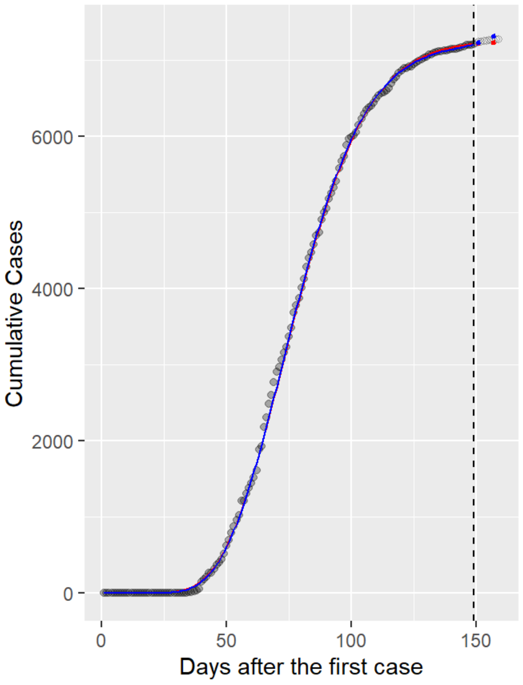
Daily Cases in Finland

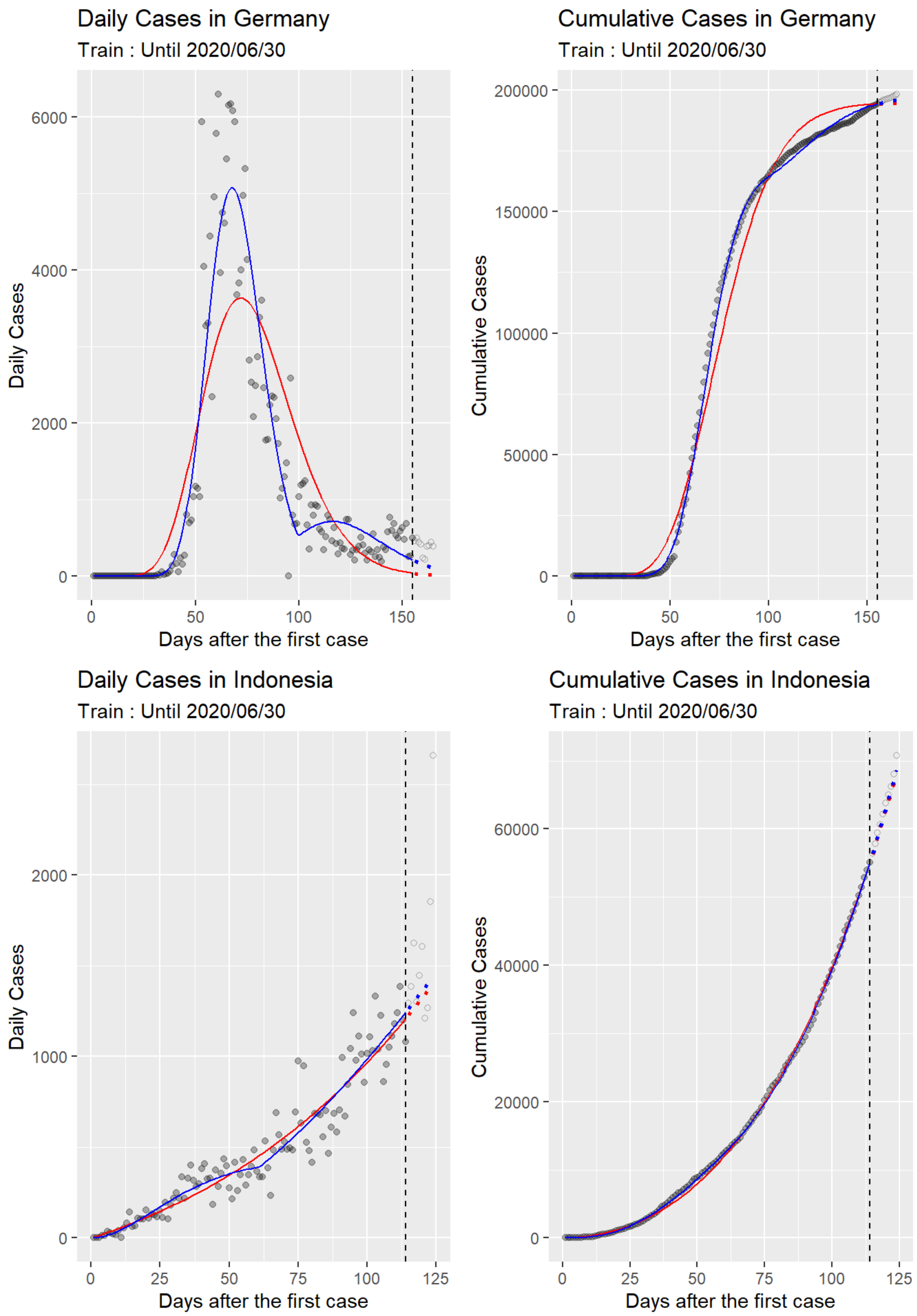
Train : Until 2020/06/30

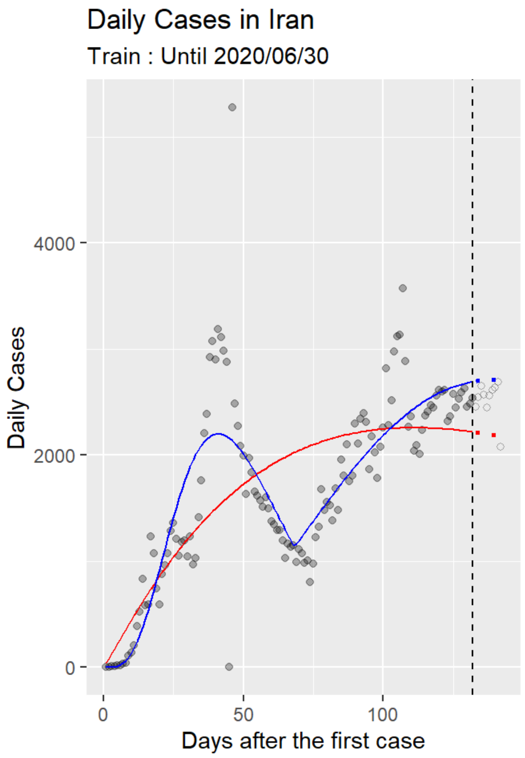


Cumulative Cases in Finland

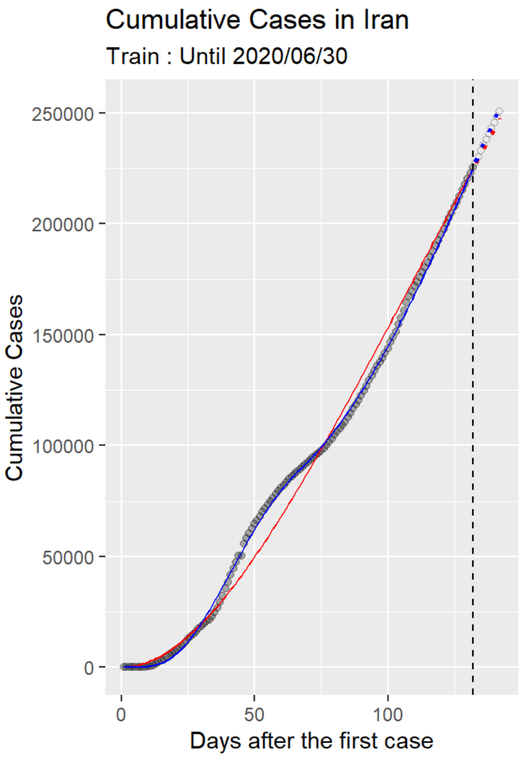
Train : Until 2020/06/30



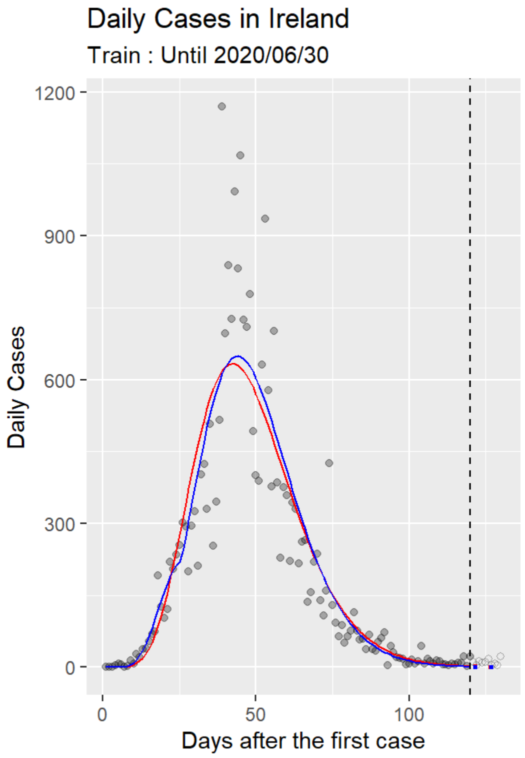

Daily Cases in Germany
Train : Until 2020/06/30
Daily Cases
0
2000
4000
6000
0
50
100
150
Days after the first case
Cumulative Cases in Germany
Train : Until 2020/06/30
Cumulative Cases
0
50000
100000
150000
200000
0
50
100
150
Days after the first case
Daily Cases in Indonesia
Train : Until 2020/06/30
Daily Cases
0
1000
2000
0
25
50
75
100
125
Days after the first case
Cumulative Cases in Indonesia
Train : Until 2020/06/30
Cumulative Cases
0
20000
40000
60000
0
25
50
75
100
125
Days after the first case


Daily Cases in Iran

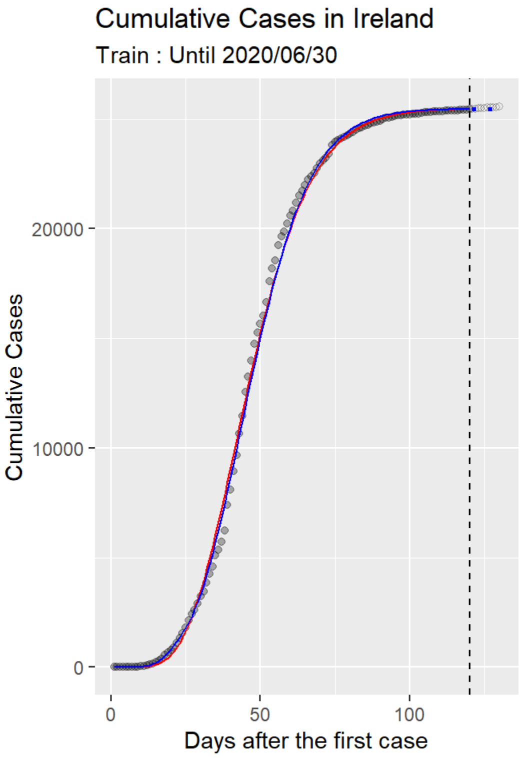
Train : Until 2020/06/30


Cumulative Cases in Iran

Train : Until 2020/06/30

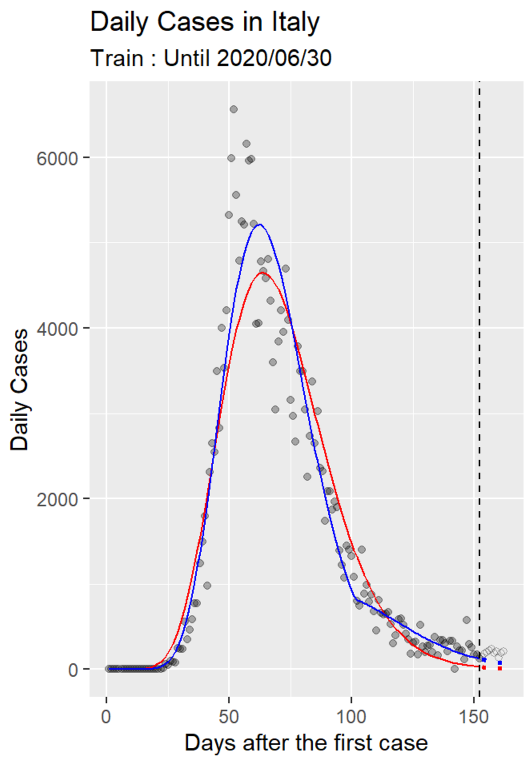

Daily Cases in Ireland

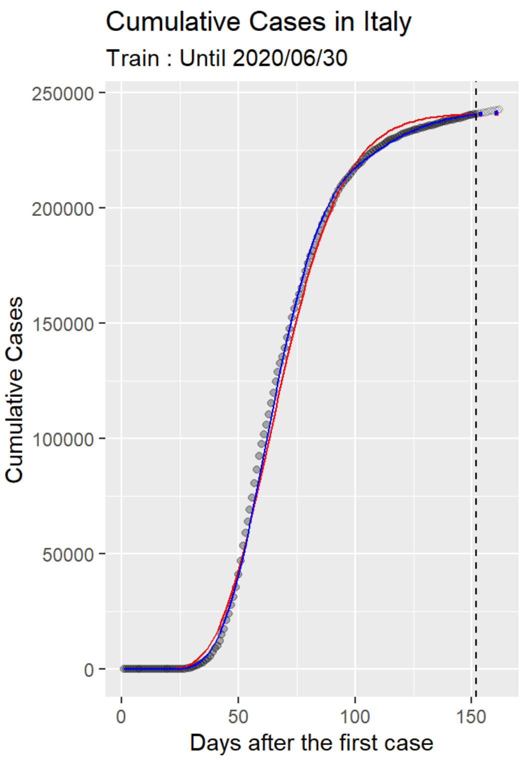
Train : Until 2020/06/30


Cumulative Cases in Ireland

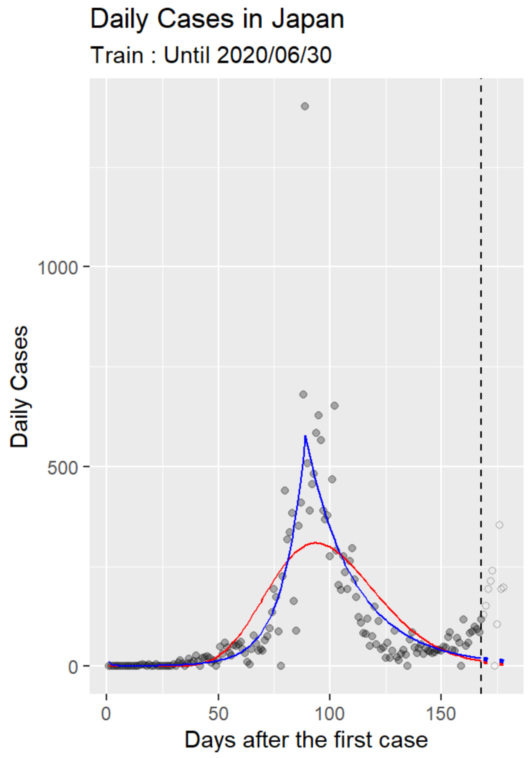
Train : Until 2020/06/30

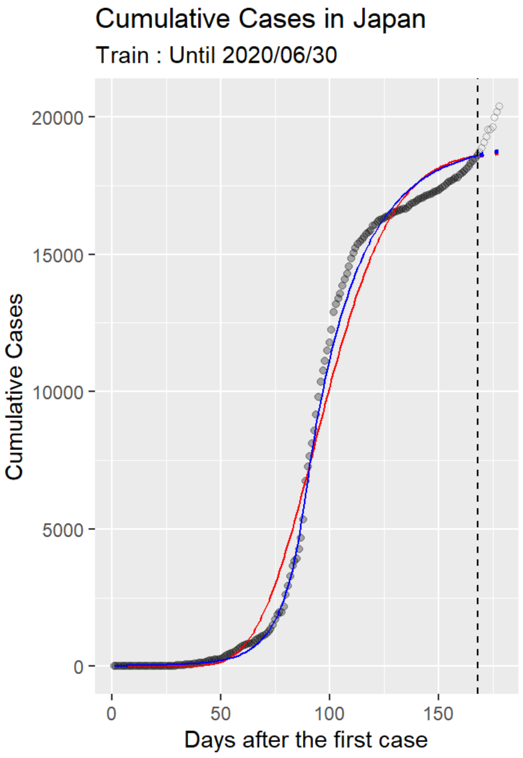
```
## Warning: glm.fit: fitted rates numerically 0 occurred

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

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

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

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

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

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

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

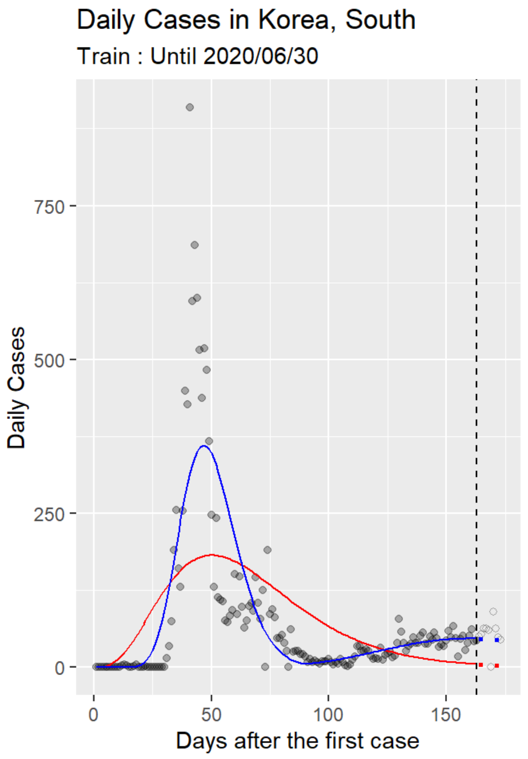




```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

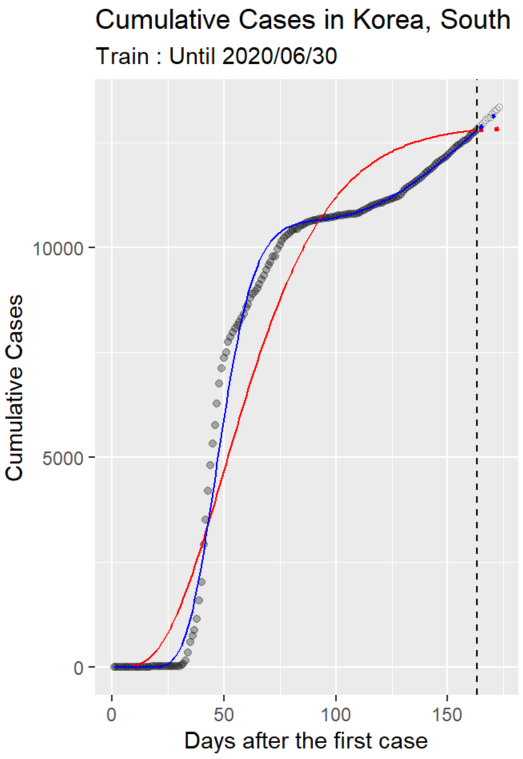






Daily Cases in Korea, South

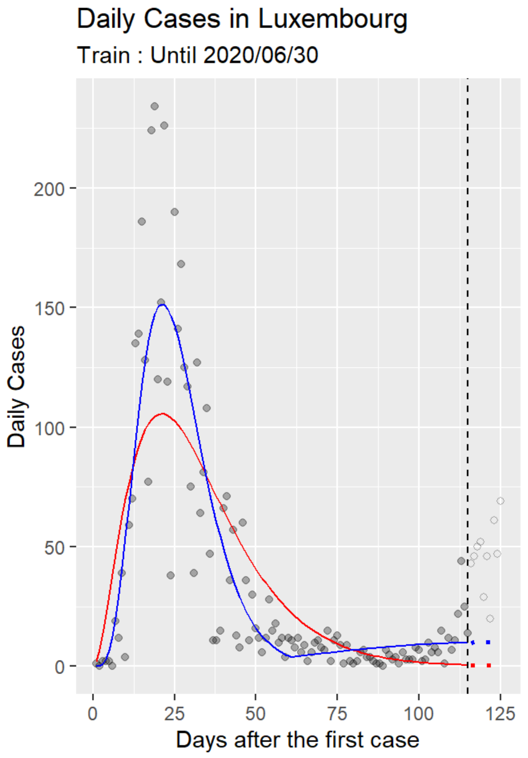
Train : Until 2020/06/30



Cumulative Cases in Korea, South

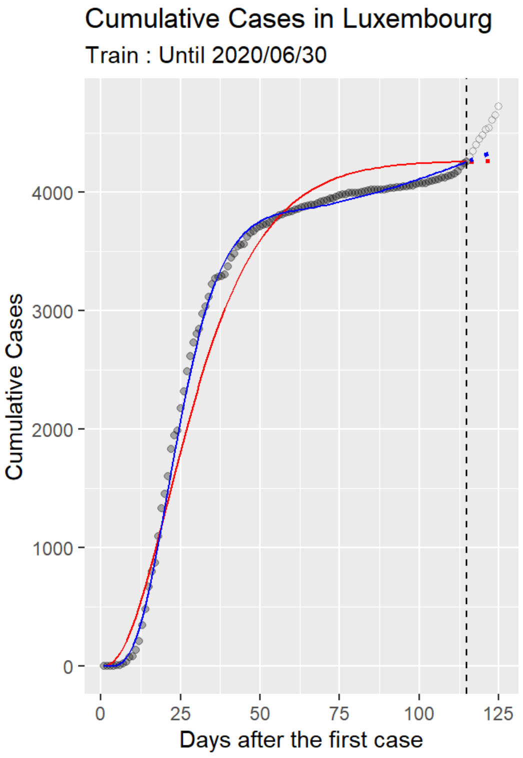
Train : Until 2020/06/30

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```



Daily Cases in Luxembourg

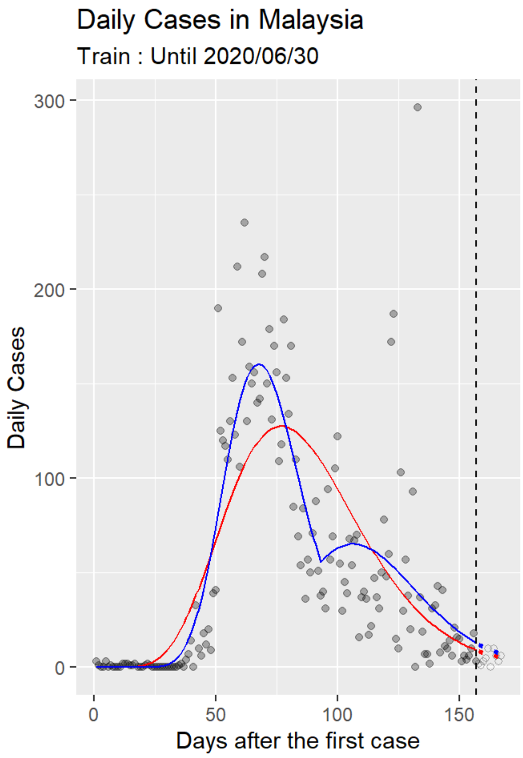
Train : Until 2020/06/30

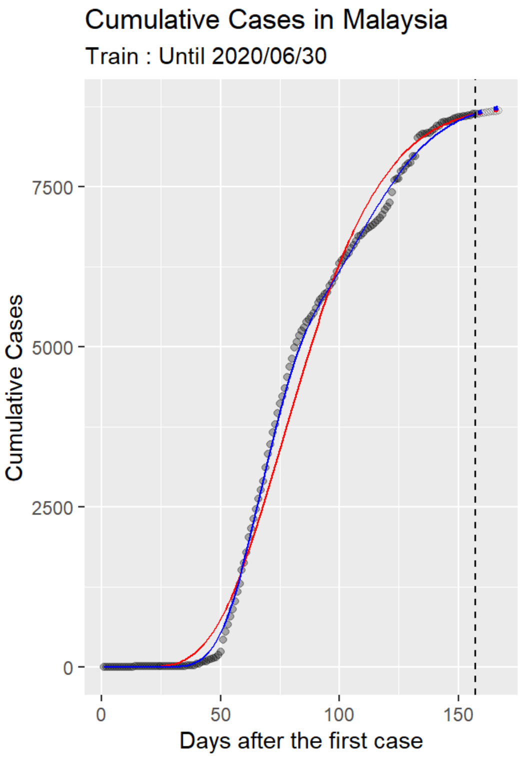


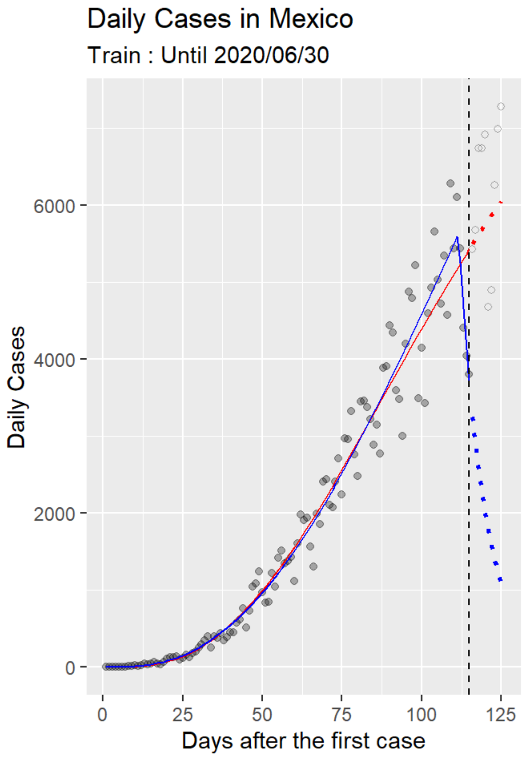
Cumulative Cases in Luxembourg

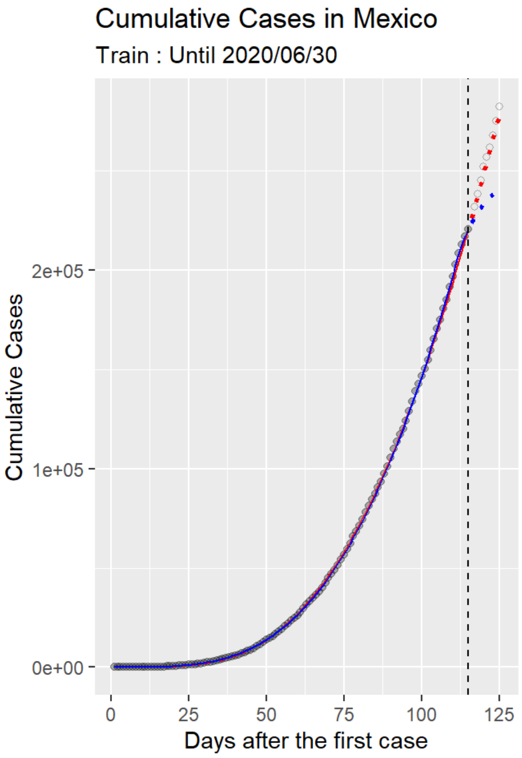
Train : Until 2020/06/30

```
## Warning: max number of iterations (31) attained
```


Daily Cases in Malaysia
Train : Until 2020/06/30
300
200
100
0
Daily Cases
0
50
100
150
Days after the first case

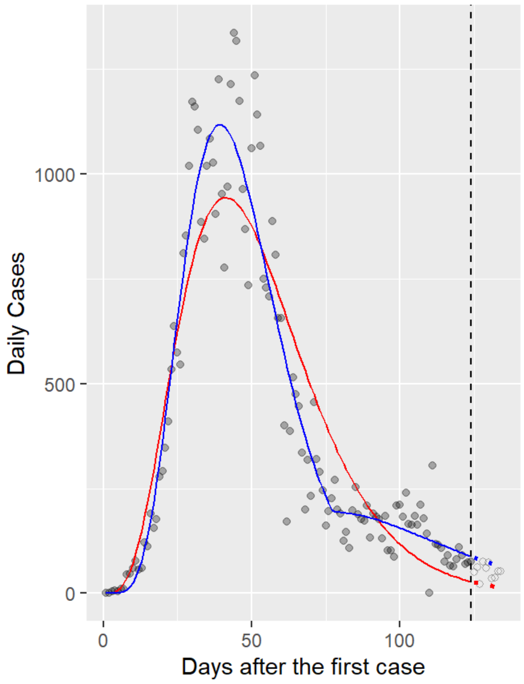

Cumulative Cases in Malaysia
Train : Until 2020/06/30
7500
5000
2500
0
Cumulative Cases
0
50
100
150
Days after the first case


Daily Cases in Mexico
Train : Until 2020/06/30
6000
4000
2000
0
Daily Cases
0
25
50
75
100
125
Days after the first case


Cumulative Cases in Mexico
Train : Until 2020/06/30
2e+05
1e+05
0e+00
Cumulative Cases
0
25
50
75
100
125
Days after the first case

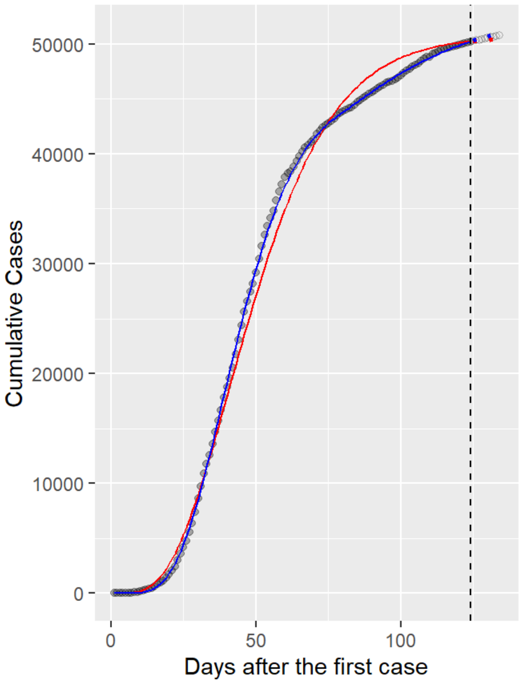
Daily Cases in Netherlands

Train : Until 2020/06/30



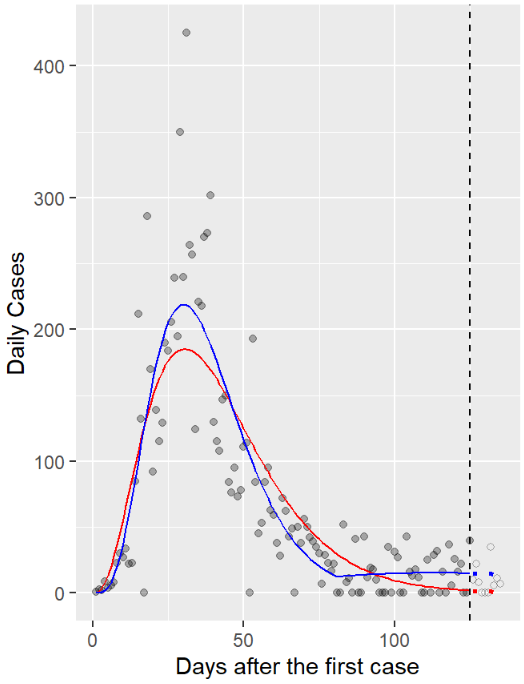
Cumulative Cases in Netherlands

Train : Until 2020/06/30



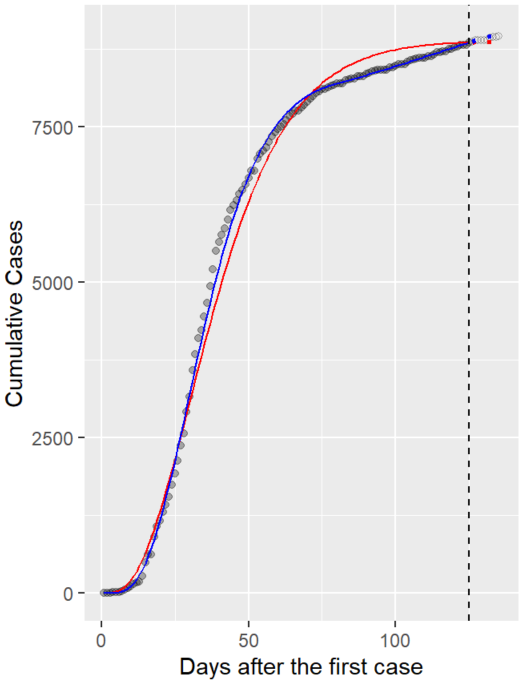
Daily Cases in Norway

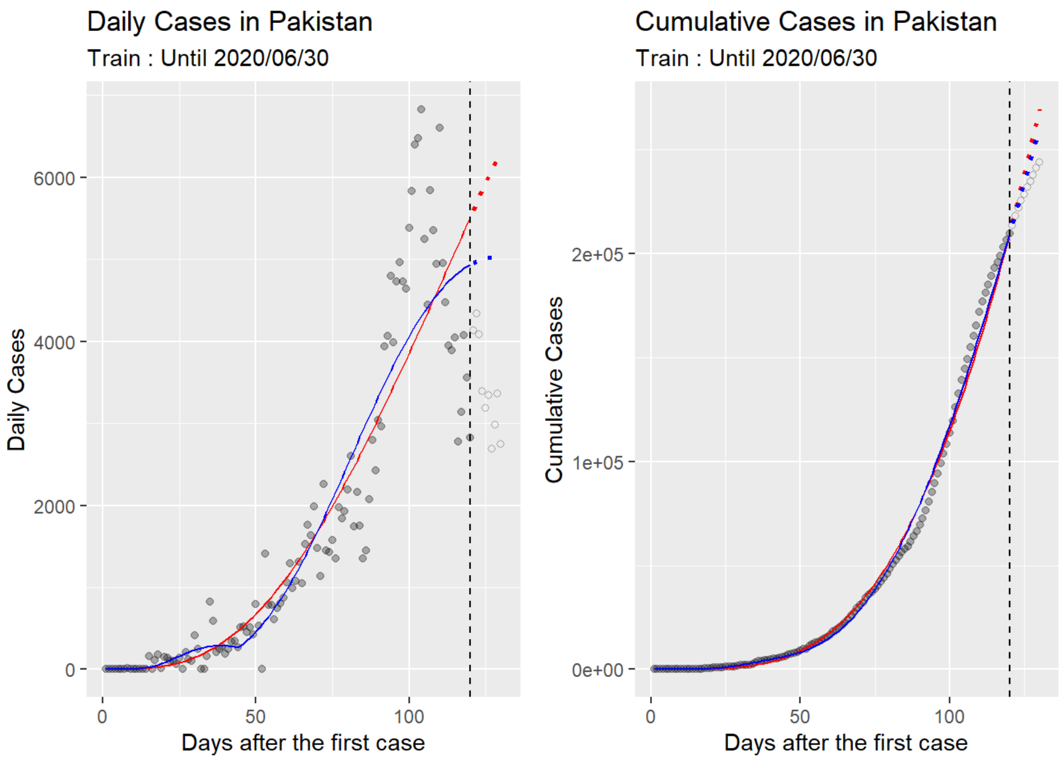
Train : Until 2020/06/30



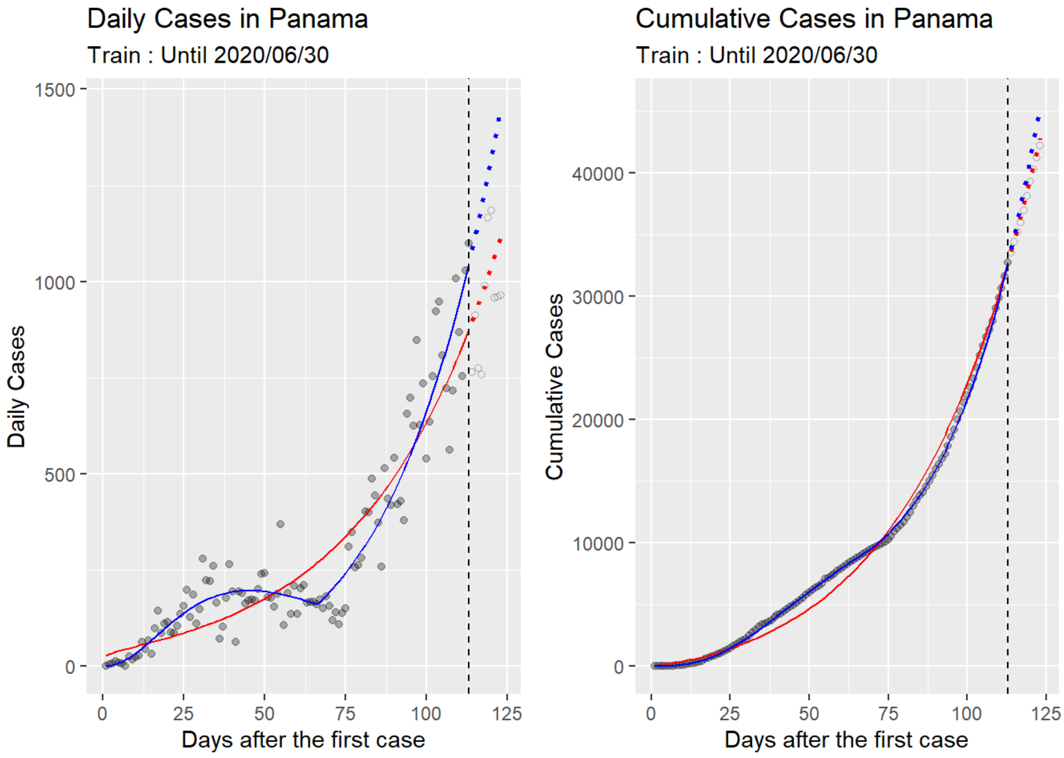
Cumulative Cases in Norway

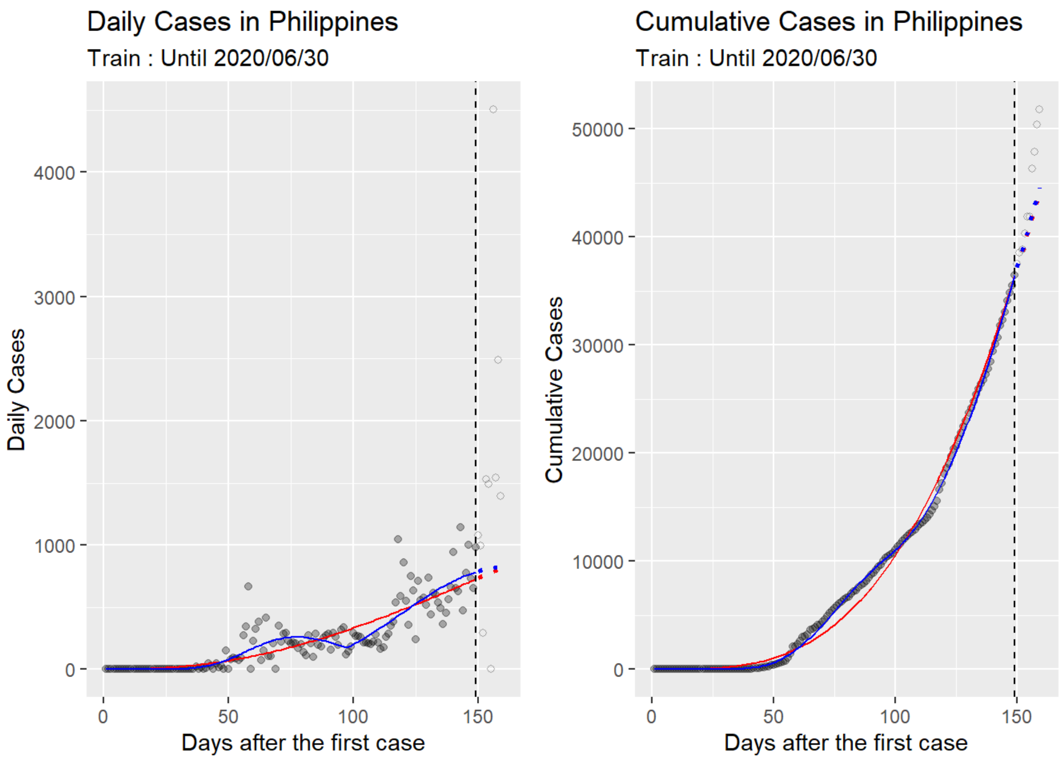
Train : Until 2020/06/30





```
## Warning: glm.fit: fitted rates numerically 0 occurred
```





Daily Cases in Philippines

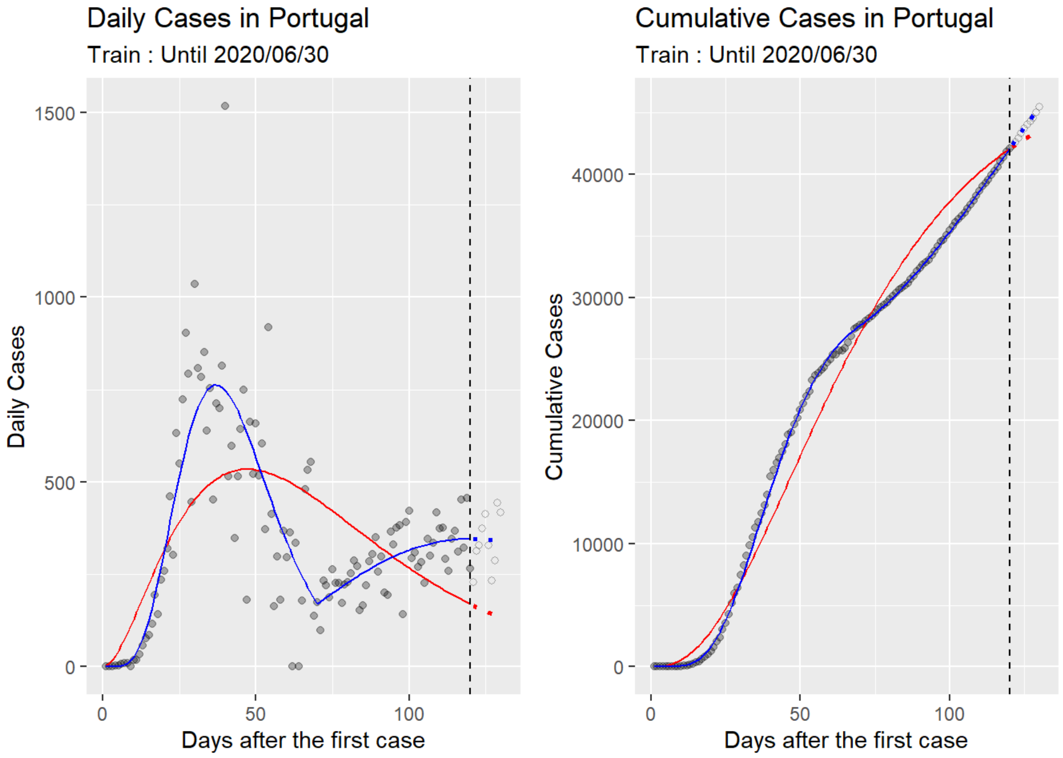
Train : Until 2020/06/30

Cumulative Cases in Philippines

Train : Until 2020/06/30

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```



Daily Cases in Portugal

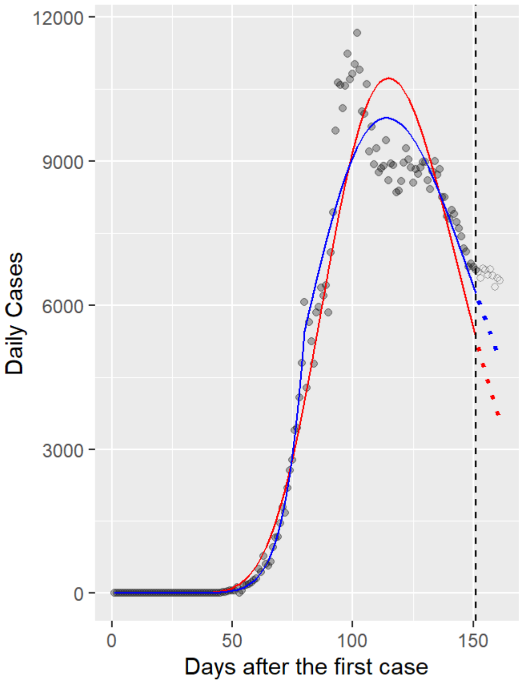
Train : Until 2020/06/30

Cumulative Cases in Portugal

Train : Until 2020/06/30

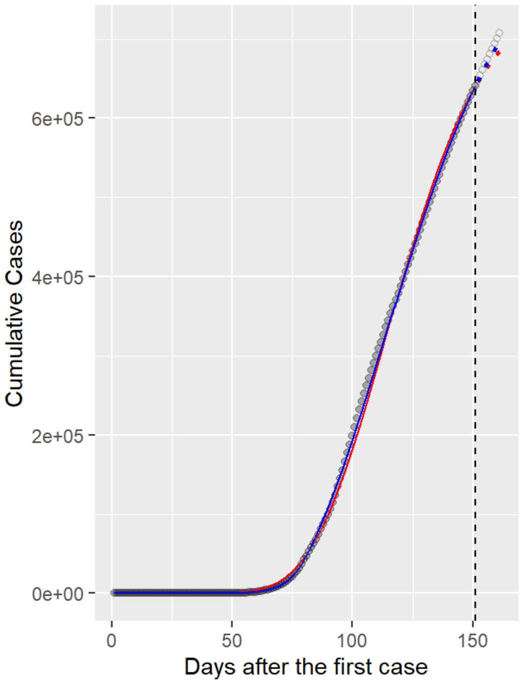
Daily Cases in Russia

Train : Until 2020/06/30



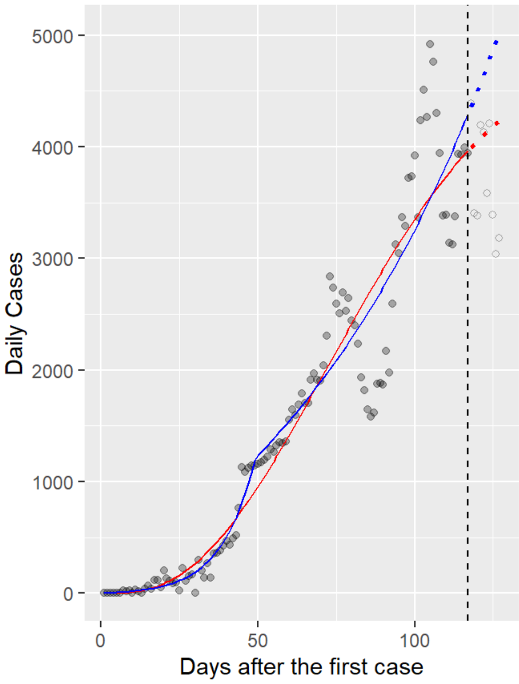
Cumulative Cases in Russia

Train : Until 2020/06/30



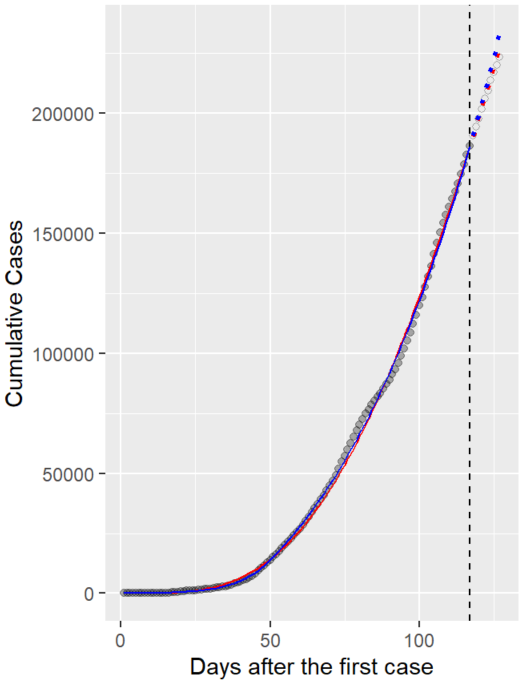
Daily Cases in Saudi Arabia

Train : Until 2020/06/30

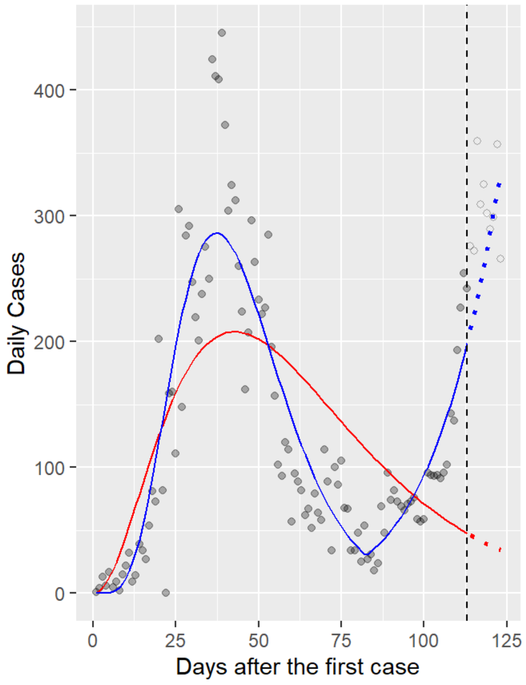


Cumulative Cases in Saudi Arabia

Train : Until 2020/06/30

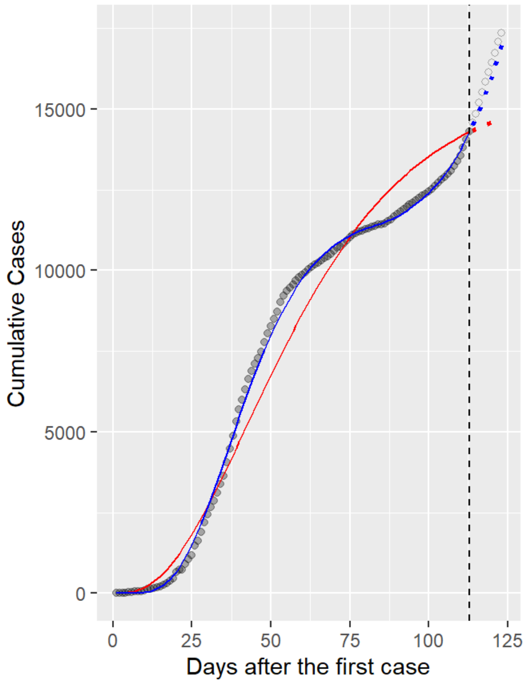


```
## Warning: glm.fit: fitted rates numerically 0 occurred
```



Daily Cases in Serbia

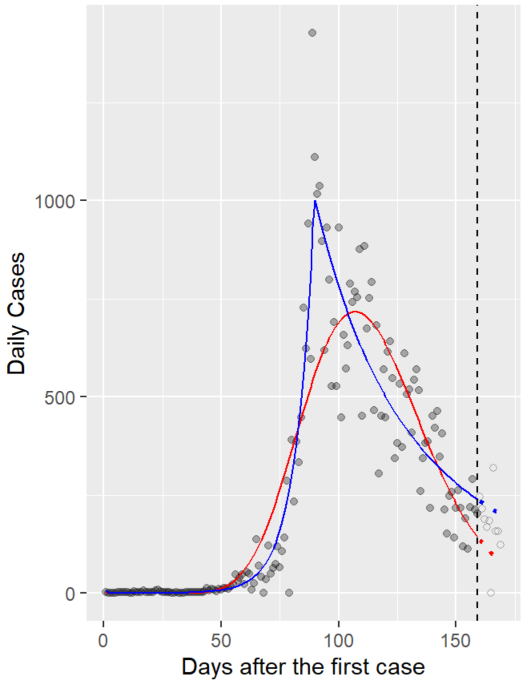
Train : Until 2020/06/30



Cumulative Cases in Serbia

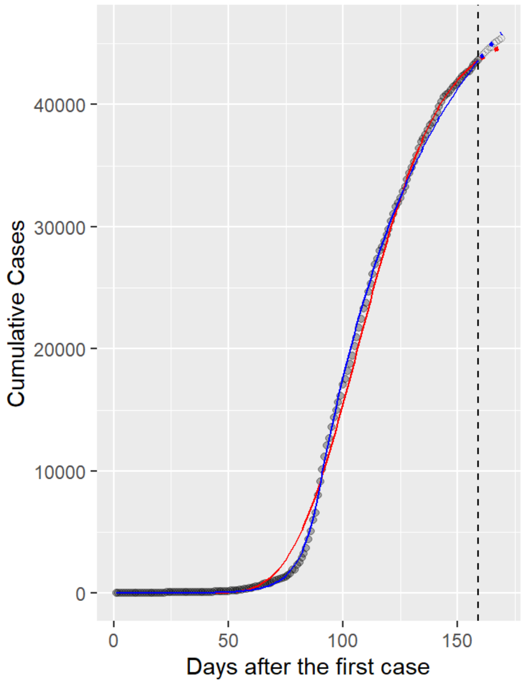
Train : Until 2020/06/30

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```



Daily Cases in Singapore

Train : Until 2020/06/30

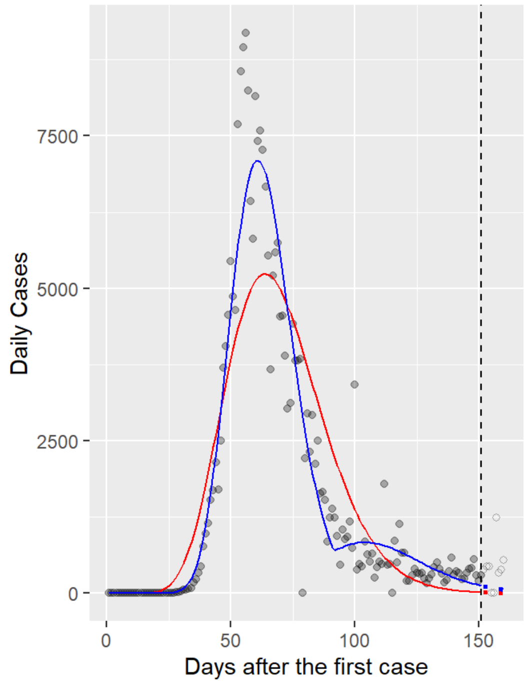


Cumulative Cases in Singapore

Train : Until 2020/06/30

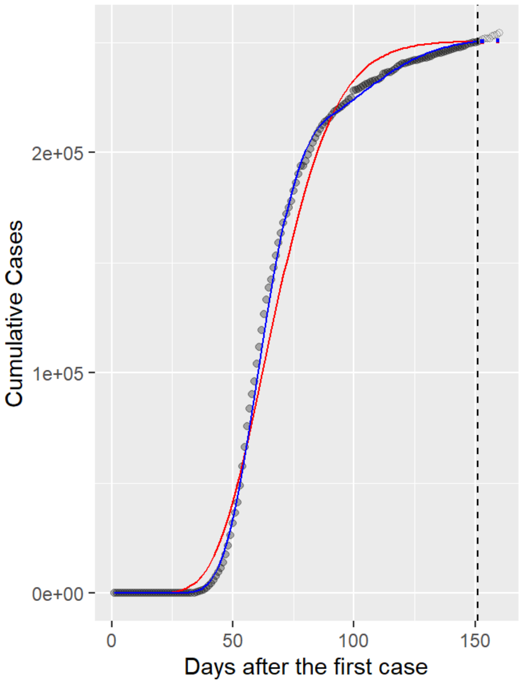
Daily Cases in Spain

Train : Until 2020/06/30



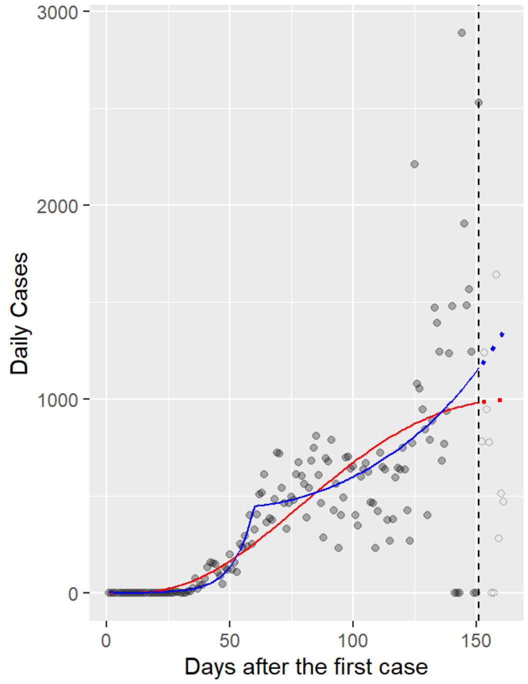
Cumulative Cases in Spain

Train : Until 2020/06/30



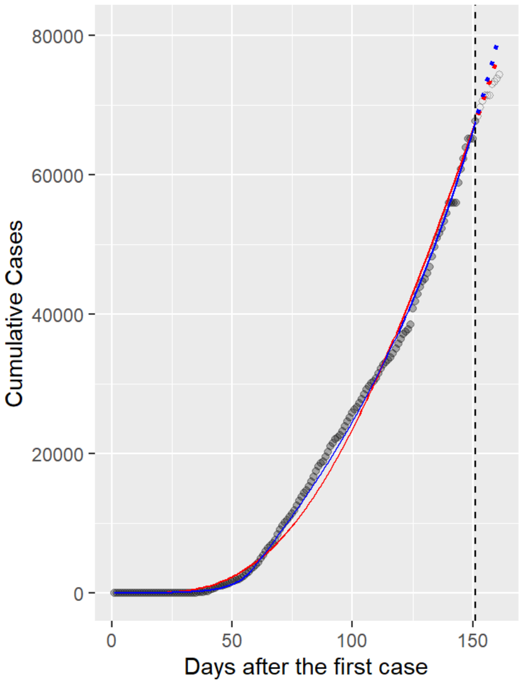
Daily Cases in Sweden

Train : Until 2020/06/30



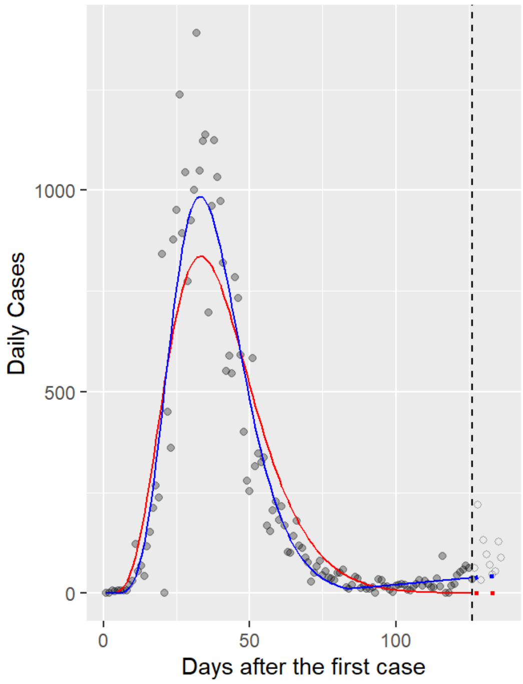
Cumulative Cases in Sweden

Train : Until 2020/06/30



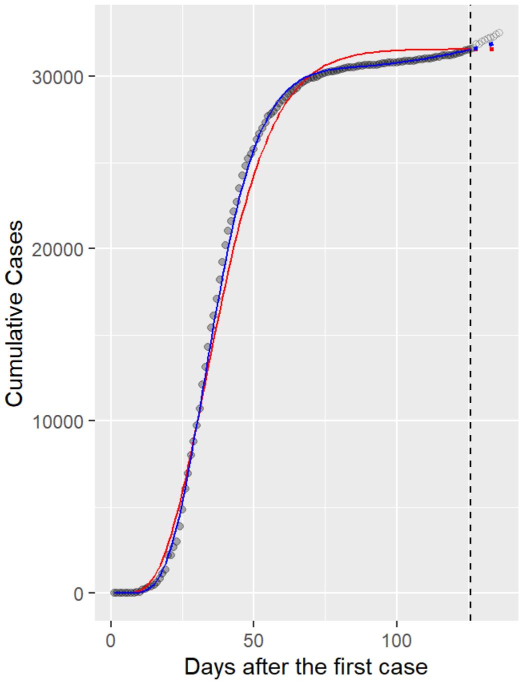
```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

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

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

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

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



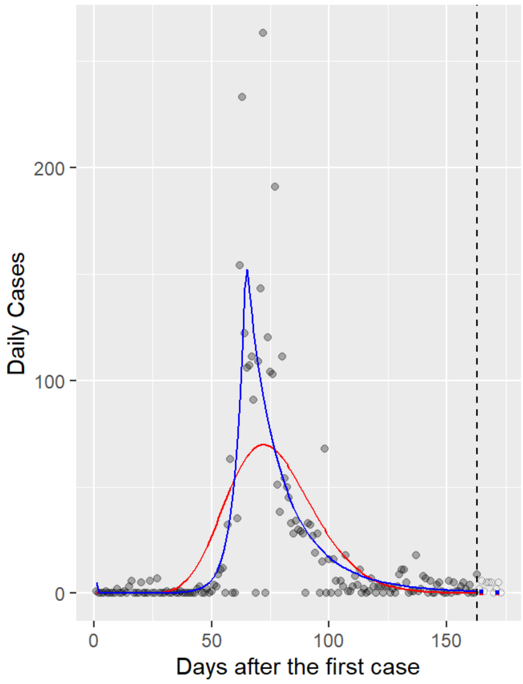
Daily Cases in Switzerland

Train : Until 2020/06/30



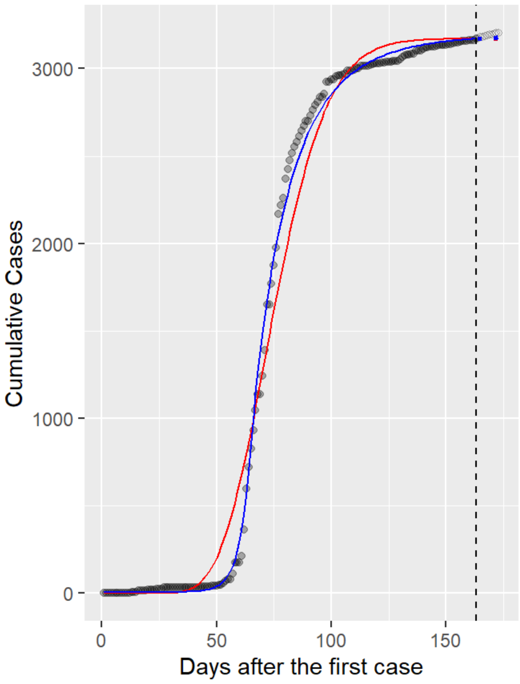
Cumulative Cases in Switzerland

Train : Until 2020/06/30



Daily Cases in Thailand

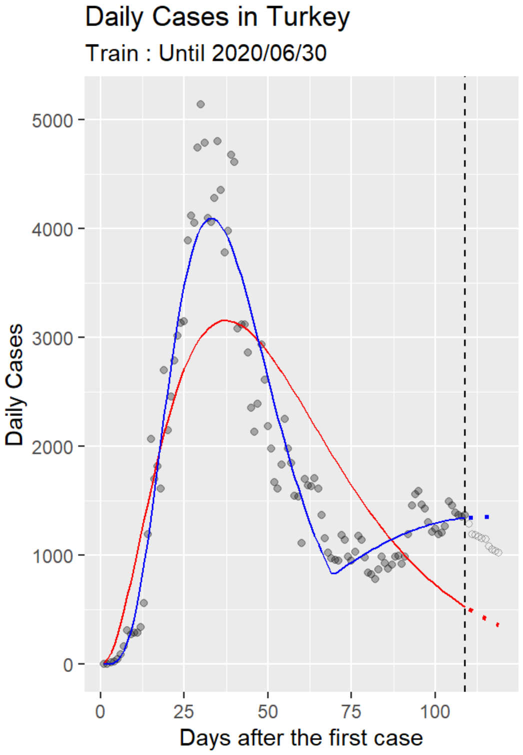
Train : Until 2020/06/30

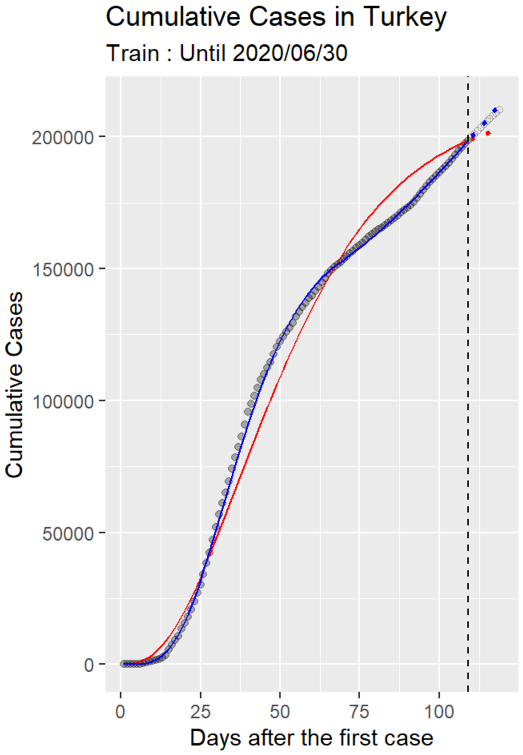


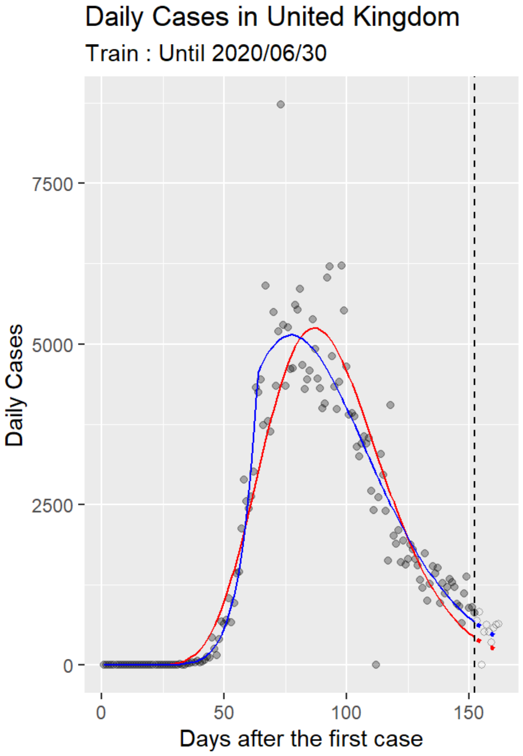
Cumulative Cases in Thailand

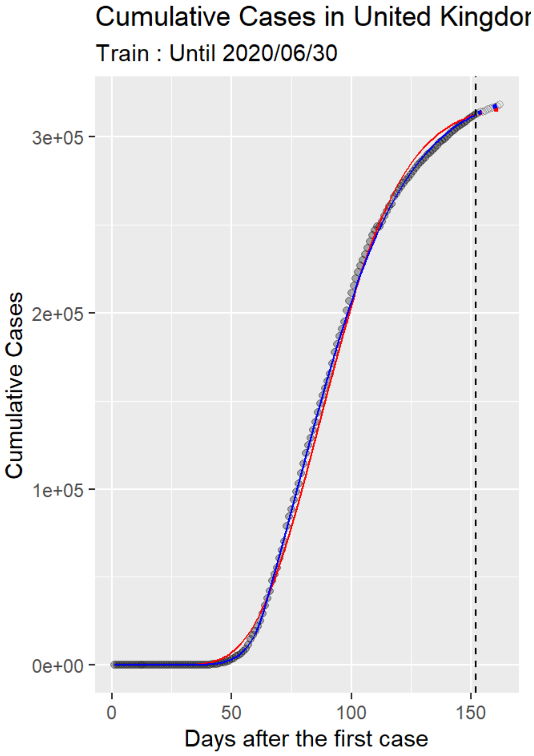
Train : Until 2020/06/30

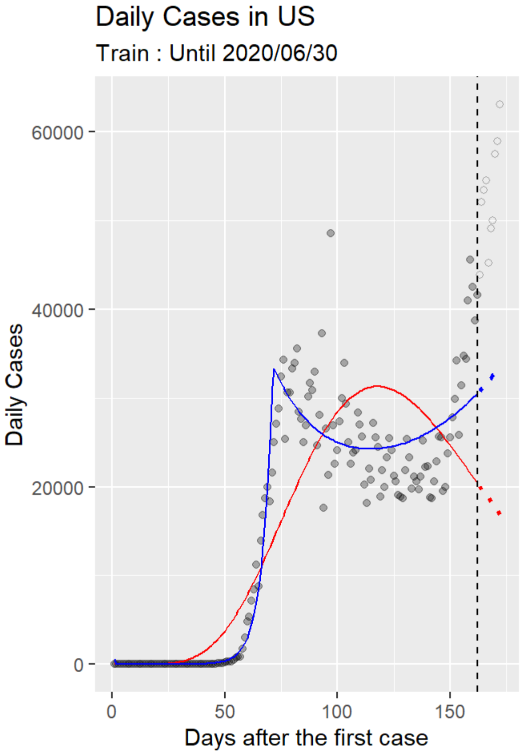
```
## Warning: glm.fit: fitted rates numerically 0 occurred
```


Daily Cases in Turkey
Train : Until 2020/06/30
Daily Cases
5000
4000
3000
2000
1000
0
0
25
50
75
100
Days after the first case


Cumulative Cases in Turkey
Train : Until 2020/06/30
Cumulative Cases
200000
150000
100000
50000
0
0
25
50
75
100
Days after the first case

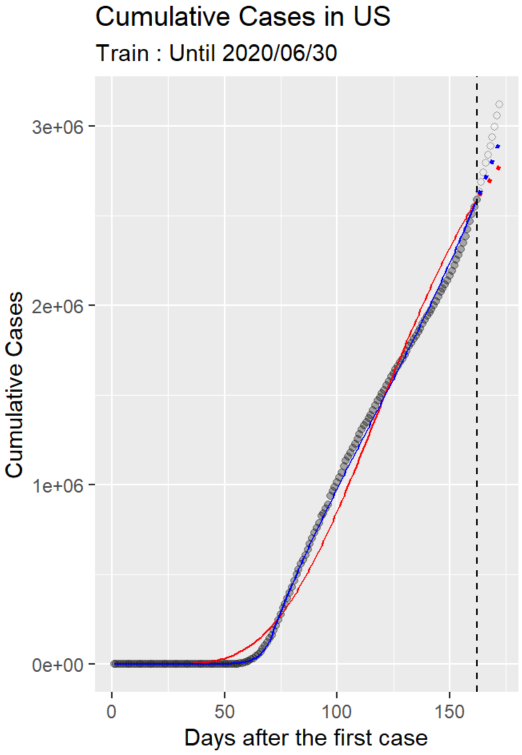

Daily Cases in United Kingdom
Train : Until 2020/06/30
Daily Cases
7500
5000
2500
0
0
50
100
150
Days after the first case


Cumulative Cases in United Kingdor
Train : Until 2020/06/30
Cumulative Cases
3e+05
2e+05
1e+05
0e+00
0
50
100
150
Days after the first case



Daily Cases in US

Train : Until 2020/06/30



Cumulative Cases in US

Train : Until 2020/06/30