

EPI Info CDC

Nick Lauerman

3/16/2020

Contents

Libraries Used	1
Load CDC data	1
Read CSV File	1
Clean data and calculate cumulative number of cases	2
Visualize all data	2
Filter to remove incomplete reporting	6
Visualize	6

Libraries Used

```
library(tidyverse)
```

```
## -- Attaching packages -----
## v ggplot2 3.3.0      v purrr  0.3.3
## v tibble  3.0.0      v dplyr  0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

## -- Conflicts -----
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

Load CDC data

Read CSV File

```
cdc <- read_csv(file = "./data/CDC_data.csv")
```

```
## Warning: Missing column names filled in: 'X3' [3]
## Warning: Duplicated column names deduplicated: '25 mar data' => '25 mar
## data_1' [12]
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   cases = col_double(),
##   X3 = col_logical(),
##   `16 mar data` = col_double(),
##   `17 mar data` = col_double(),
##   `18 mar data` = col_double(),
##   `19 mar data` = col_double(),
##   `20 mar data` = col_double(),
##   `23 mar data` = col_double(),
##   `24 mar data` = col_double(),
##   `25 mar data` = col_double(),
##   `25 mar data_1` = col_double(),
##   `26 mar data` = col_double(),
##   `29 mar data` = col_double(),
##   `30 mar data` = col_double(),
##   `1 Apr data` = col_double()
## )
```

Clean data and calculate cumulative number of cases

```
names(cdc)[1:2] <- c("Date",
                    "Number of new cases")
cdc$cum <- cumsum(cdc$`Number of new cases`)
cdc$Date <- as.Date(cdc$Date,
                   format = "%d-%b-%y")
```

Visualize all data

```
epi <- ggplot(data = cdc)

epi + geom_point(aes(x = Date,
                    y = `Number of new cases`))+
#   geom_line(aes(x = Date,
#                 y = `Number of new cases`),
#             linetype = 2) +
geom_smooth(aes(x = Date,
                y = `Number of new cases`),
            color = "red",
            fill = "blue") +
labs(y = "Cases",
     title = "Number of New Cases of COVID-19 Reported to the CDC")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

epi + geom_col(aes(x=Date,
                  y=`Number of new cases`),
              fill= "blue") +
```

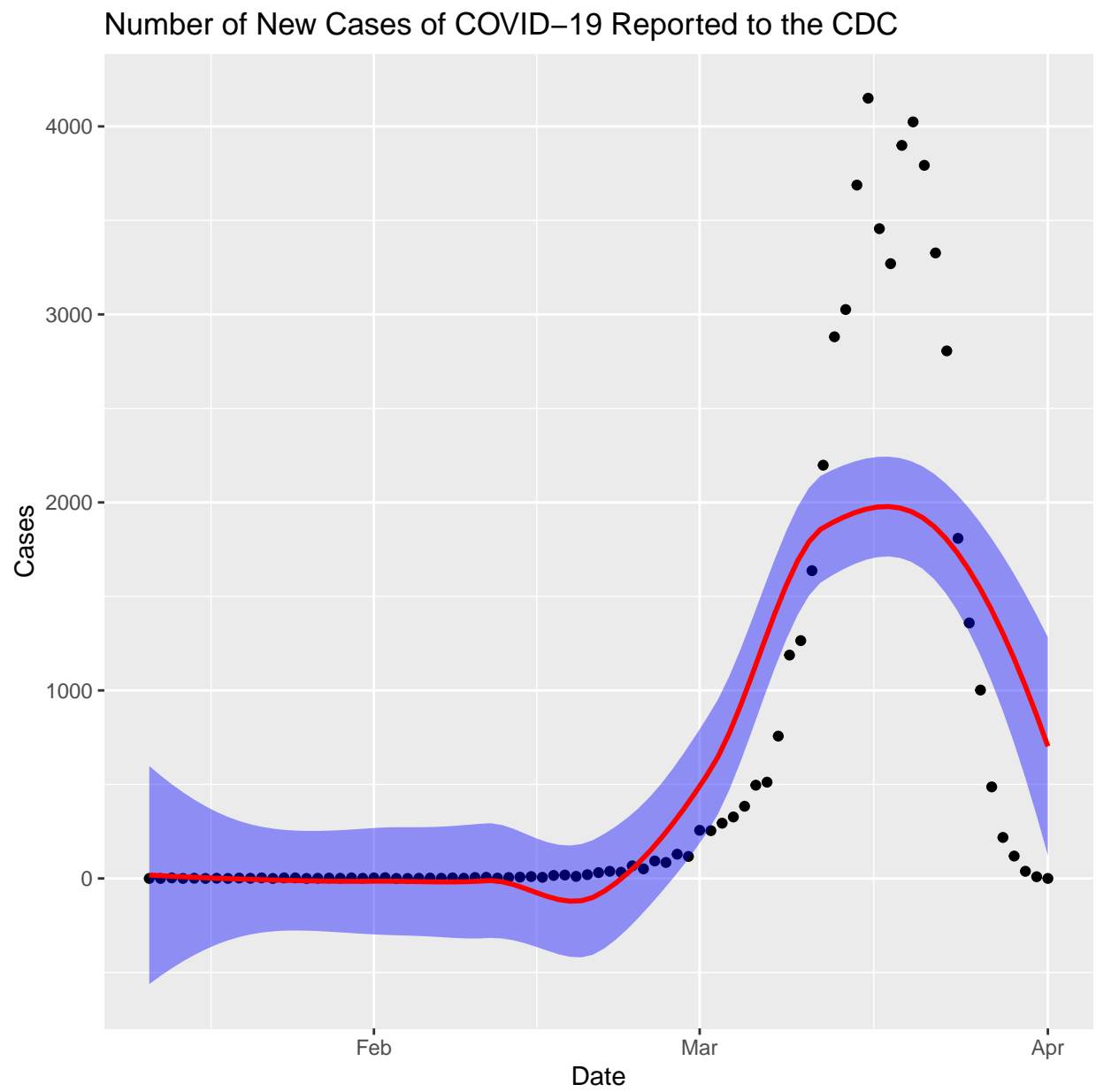


Figure 1: Epi curve 1

```
geom_smooth(aes(x=Date,
                y=`Number of new cases`),
            color = "red") +
labs(y = "Cases",
     title = "Number of New Cases of COVID-19 Reported to the CDC")
```

`geom_smooth()` using method = 'loess' and formula 'y ~ x'

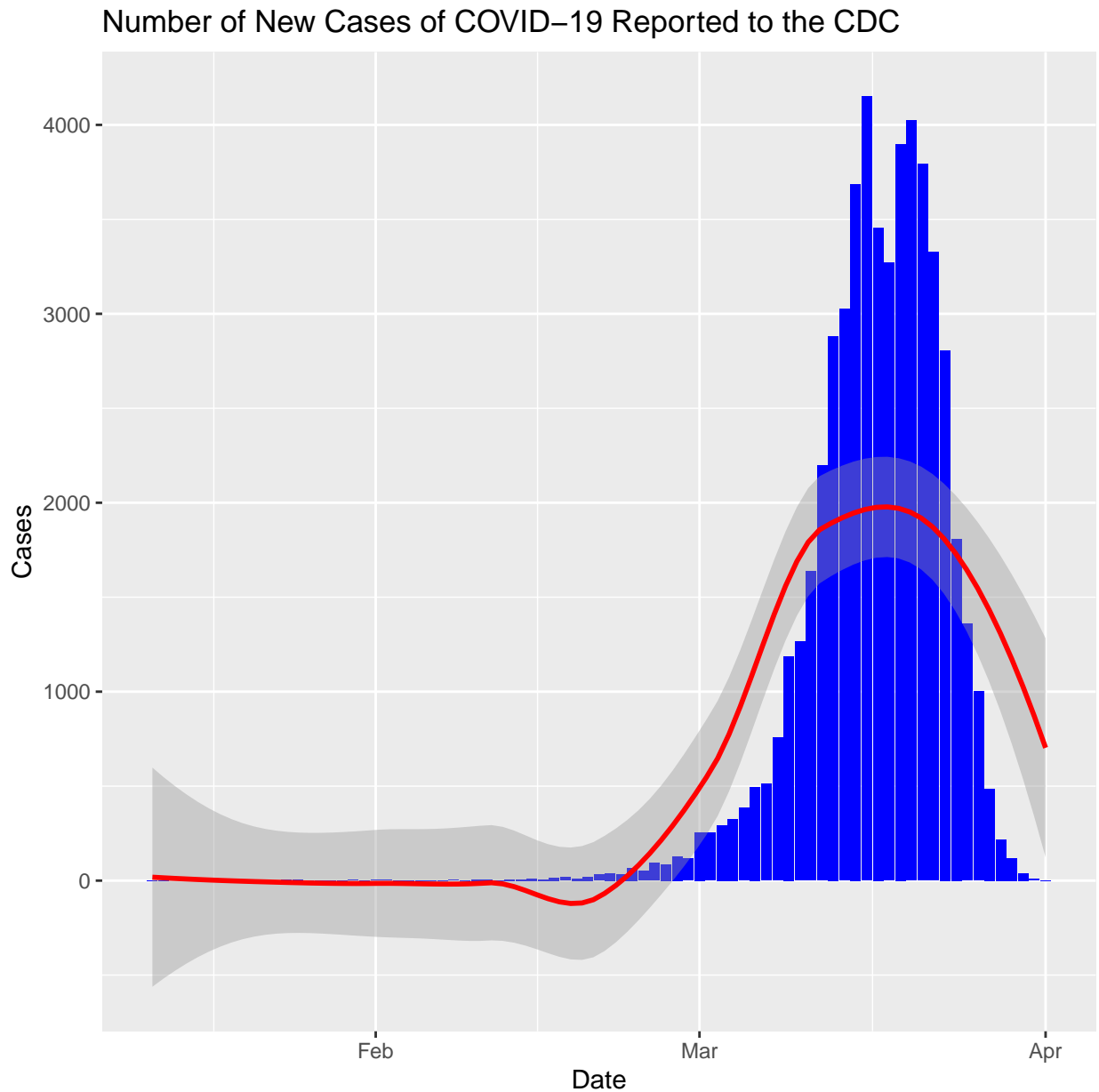


Figure 2: Epi curve 2, traditional

```
epi + geom_point(aes(x = Date,
                     y = cum)) +
  # geom_line(aes(x = Date,
  #               y = `cum`)) +
```

```
geom_smooth(aes(x = Date,
  y = `cum`),
  color = "red",
  fill = "blue") +
labs(y = "Cumulative number of cases",
  title = "Cumulative Number of Cases of COVID-19 Reported to the CDC") +
geom_hline(yintercept = mean(cdc$cum)) +
geom_hline(yintercept = median(cdc$cum),
  lty = 2)
```

`geom_smooth()` using method = 'loess' and formula 'y ~ x'

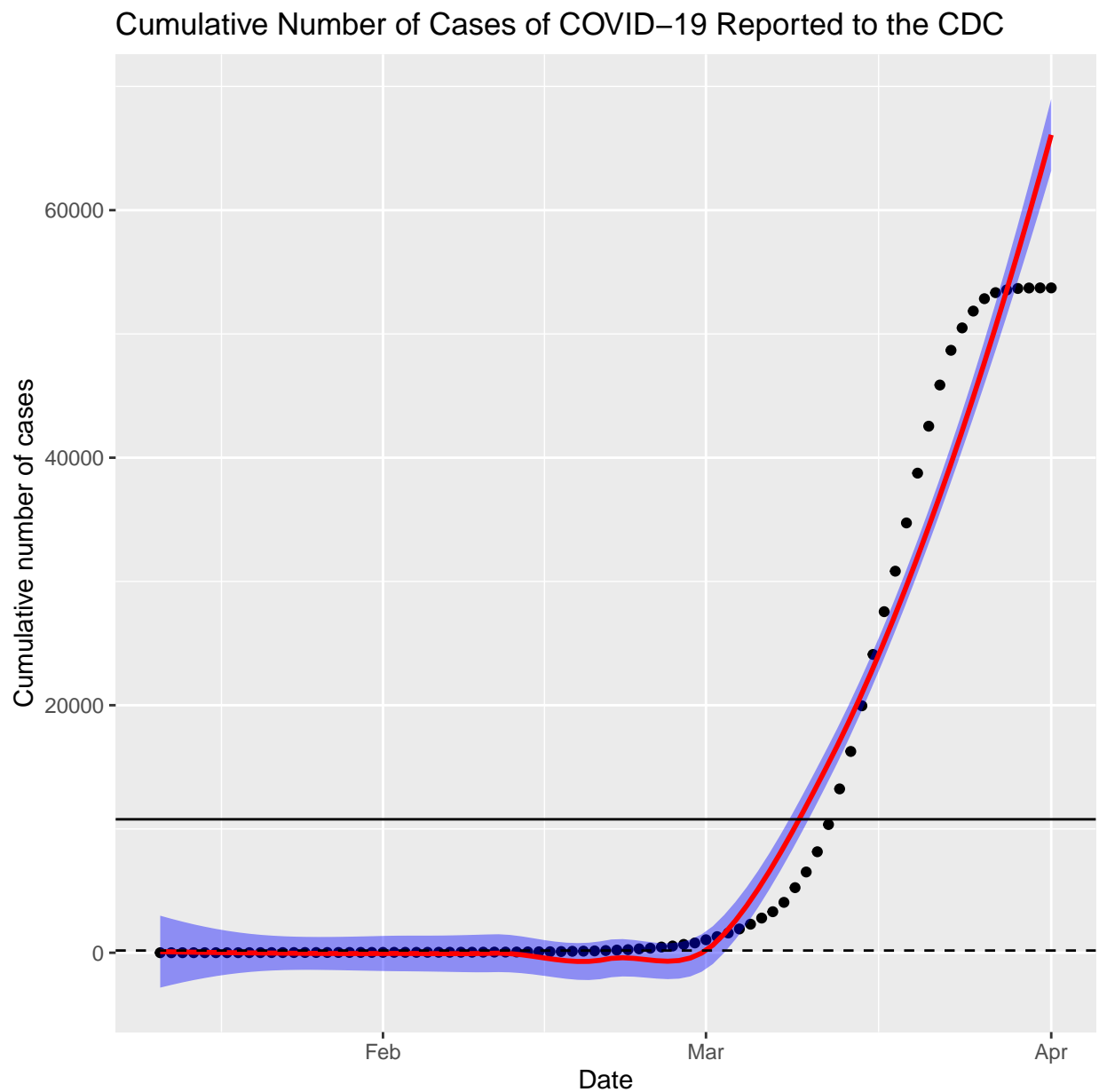


Figure 3: Cumulative cases

Filter to remove incomplete reporting

remove dates on or after 15 March as this data may not be completely reported

```
cdc <- cdc %>%  
  filter(Date < as.Date("2020-03-22"))
```

Visualize

```
epi <- ggplot(data = cdc)
```

```
epi + geom_point(aes(x = Date,  
  y = `Number of new cases`))+  
#   geom_line(aes(x = Date,  
#     y = `Number of new cases`),  
#     linetype = 2) +  
  geom_smooth(aes(x = Date,  
    y = `Number of new cases`),  
    color = "red",  
    fill = "blue") +  
  labs(y = "Cases",  
    title = "Number of New Cases of COVID-19 Reported to the CDC")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
epi + geom_col(aes(x=Date,  
  y=`Number of new cases`),  
  fill= "blue") +  
  geom_smooth(aes(x=Date,  
    y=`Number of new cases`),  
    color = "red") +  
  labs(y = "Cases",  
    title = "Number of New Cases of COVID-19 Reported to the CDC")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
epi + geom_point(aes(x = Date,  
  y = cum))+  
#   geom_line(aes(x = Date,  
#     y = `cum`)) +  
  geom_smooth(aes(x = Date,  
    y = `cum`),  
    color = "red",  
    fill = "blue") +  
  labs(y = "Cumulative number of cases",  
    title = "Cumulative Number of Cases of COVID-19 Reported to the CDC") +  
  geom_hline(yintercept = mean(cdc$cum)) +  
  geom_hline(yintercept = median(cdc$cum),  
    lty = 2)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

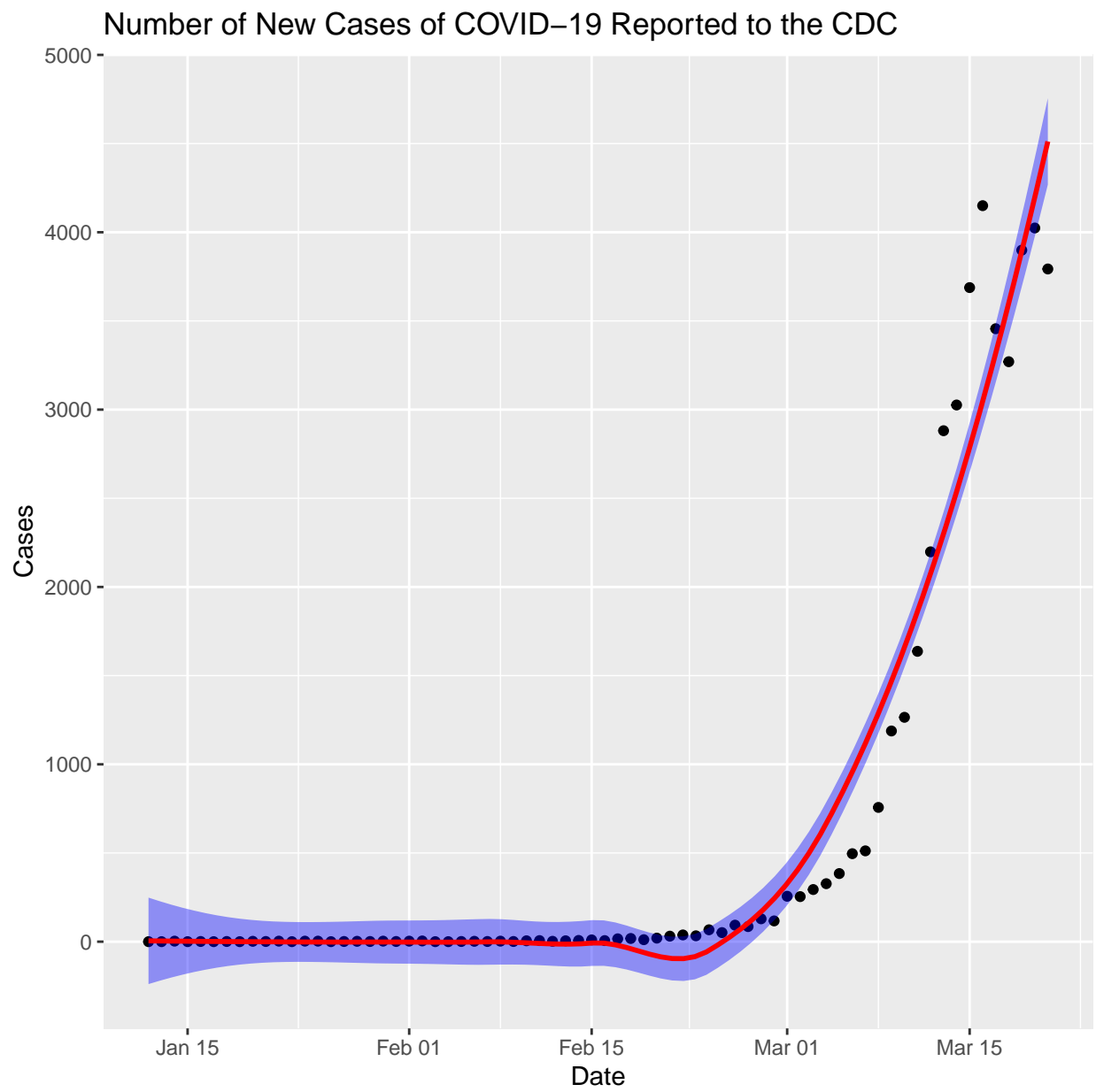


Figure 4: Epi curve 1

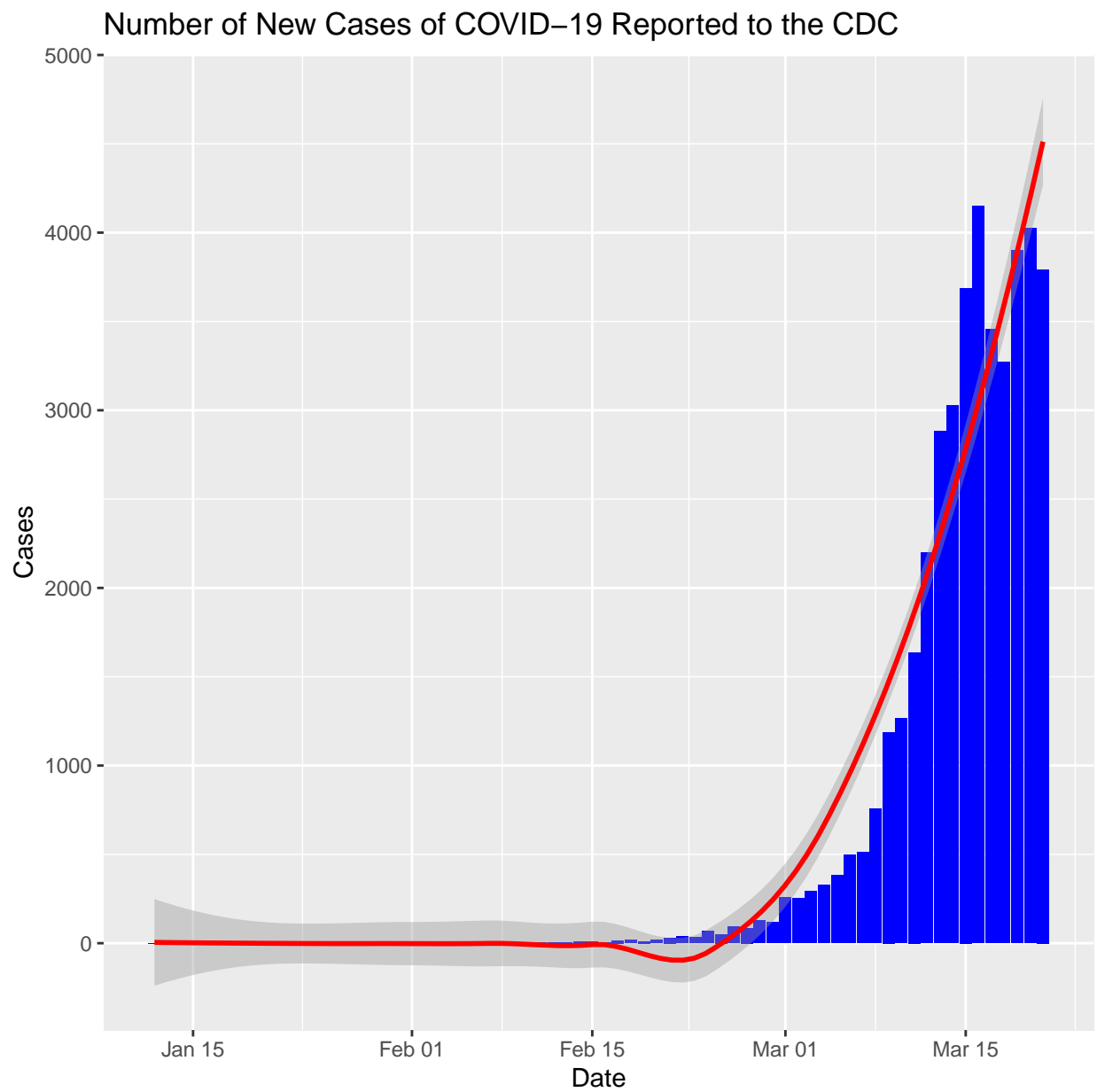


Figure 5: Epi curve 2, traditional

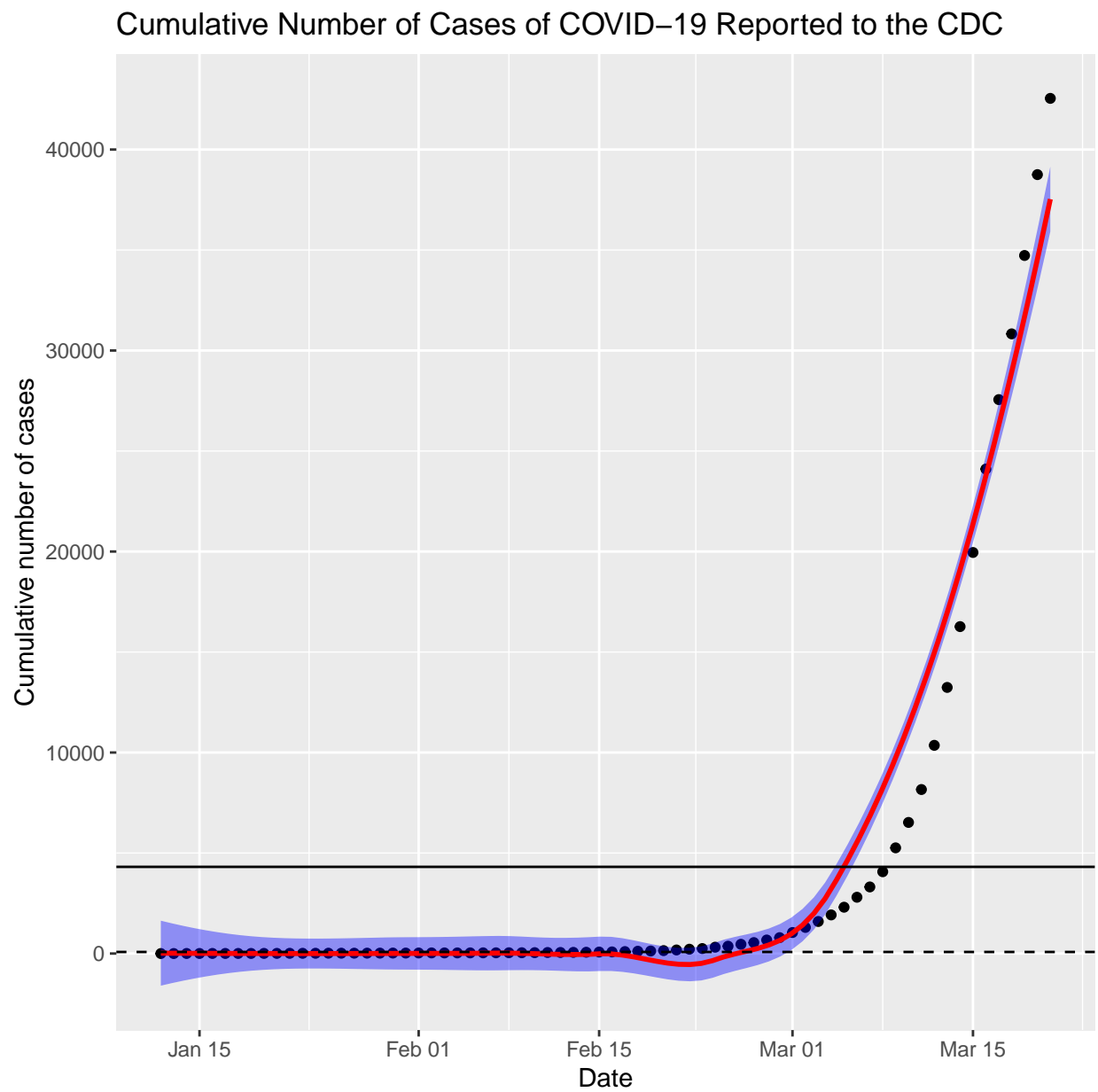


Figure 6: Cumulative cases