EPI Info CDC

Nick Lauerman

3/16/2020

Contents

Libraries Used

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     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")
```

```
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   cases = 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")
```

Data

```
cdc %>%
  data.frame
```

```
##          Date Number.of.new.cases   cum
## 1  2020-01-12                    0     0
## 2  2020-01-13                    0     0
## 3  2020-01-14                    2     2
## 4  2020-01-15                    0     2
## 5  2020-01-16                    1     3
## 6  2020-01-17                    0     3
## 7  2020-01-18                    0     3
## 8  2020-01-19                    0     3
## 9  2020-01-20                    1     4
## 10 2020-01-21                    1     5
## 11 2020-01-22                    2     7
## 12 2020-01-23                    0     7
## 13 2020-01-24                    1     8
## 14 2020-01-25                    3    11
## 15 2020-01-26                    0    11
## 16 2020-01-27                    0    11
## 17 2020-01-28                    3    14
## 18 2020-01-29                    1    15
## 19 2020-01-30                    1    16
## 20 2020-01-31                    0    16
## 21 2020-02-01                    1    17
## 22 2020-02-02                    4    21
## 23 2020-02-03                    0    21
## 24 2020-02-04                    0    21
## 25 2020-02-05                    0    21
## 26 2020-02-06                    2    23
## 27 2020-02-07                    1    24
```

```
## 28 2020-02-08                  1    25
## 29 2020-02-09                  0    25
## 30 2020-02-10                  3    28
## 31 2020-02-11                  6    34
## 32 2020-02-12                  0    34
## 33 2020-02-13                  4    38
## 34 2020-02-14                  3    41
## 35 2020-02-15                  7    48
## 36 2020-02-16                  3    51
## 37 2020-02-17                 13    64
## 38 2020-02-18                 14    78
## 39 2020-02-19                  8    86
## 40 2020-02-20                 12    98
## 41 2020-02-21                 23   121
## 42 2020-02-22                 23   144
## 43 2020-02-23                 23   167
## 44 2020-02-24                 50   217
## 45 2020-02-25                 39   256
## 46 2020-02-26                 73   329
## 47 2020-02-27                 68   397
## 48 2020-02-28                 98   495
## 49 2020-02-29                 91   586
## 50 2020-03-01                165   751
## 51 2020-03-02                180   931
## 52 2020-03-03                213  1144
## 53 2020-03-04                235  1379
## 54 2020-03-05                257  1636
## 55 2020-03-06                331  1967
## 56 2020-03-07                349  2316
## 57 2020-03-08                496  2812
## 58 2020-03-09                692  3504
## 59 2020-03-10                659  4163
## 60 2020-03-11                791  4954
## 61 2020-03-12               1048  6002
## 62 2020-03-13               1105  7107
## 63 2020-03-14               1051  8158
## 64 2020-03-15                991  9149
## 65 2020-03-16                916 10065
## 66 2020-03-17                455 10520
## 67 2020-03-18                281 10801
## 68 2020-03-19                180 10981
## 69 2020-03-20                 52 11033
## 70 2020-03-21                 13 11046
## 71 2020-03-22                  5 11051
## 72 2020-03-23                  4 11055
## 73 2020-03-24                  0 11055
```

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'
```

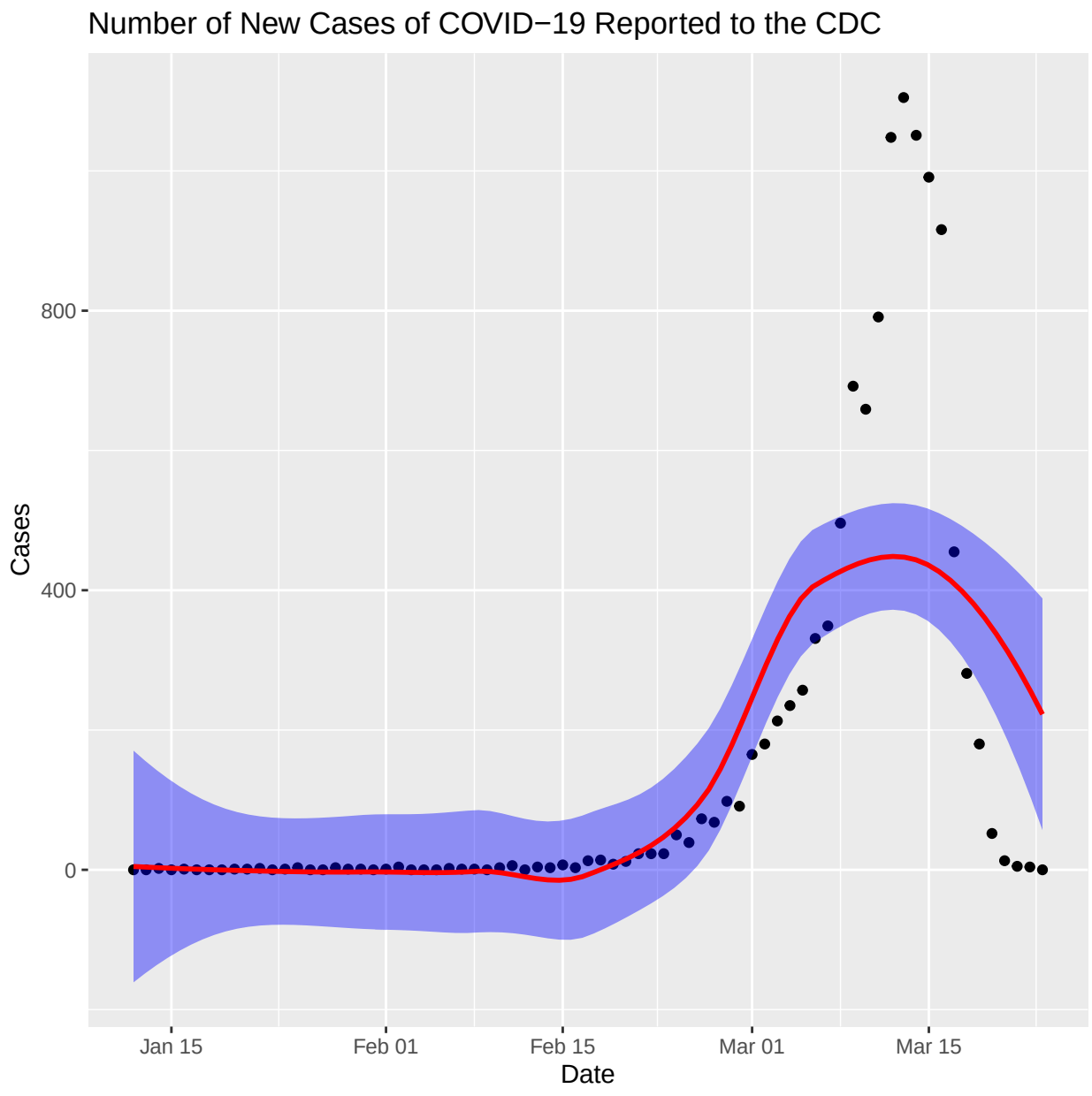

Figure 1: Epi curve 1

```
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'
```

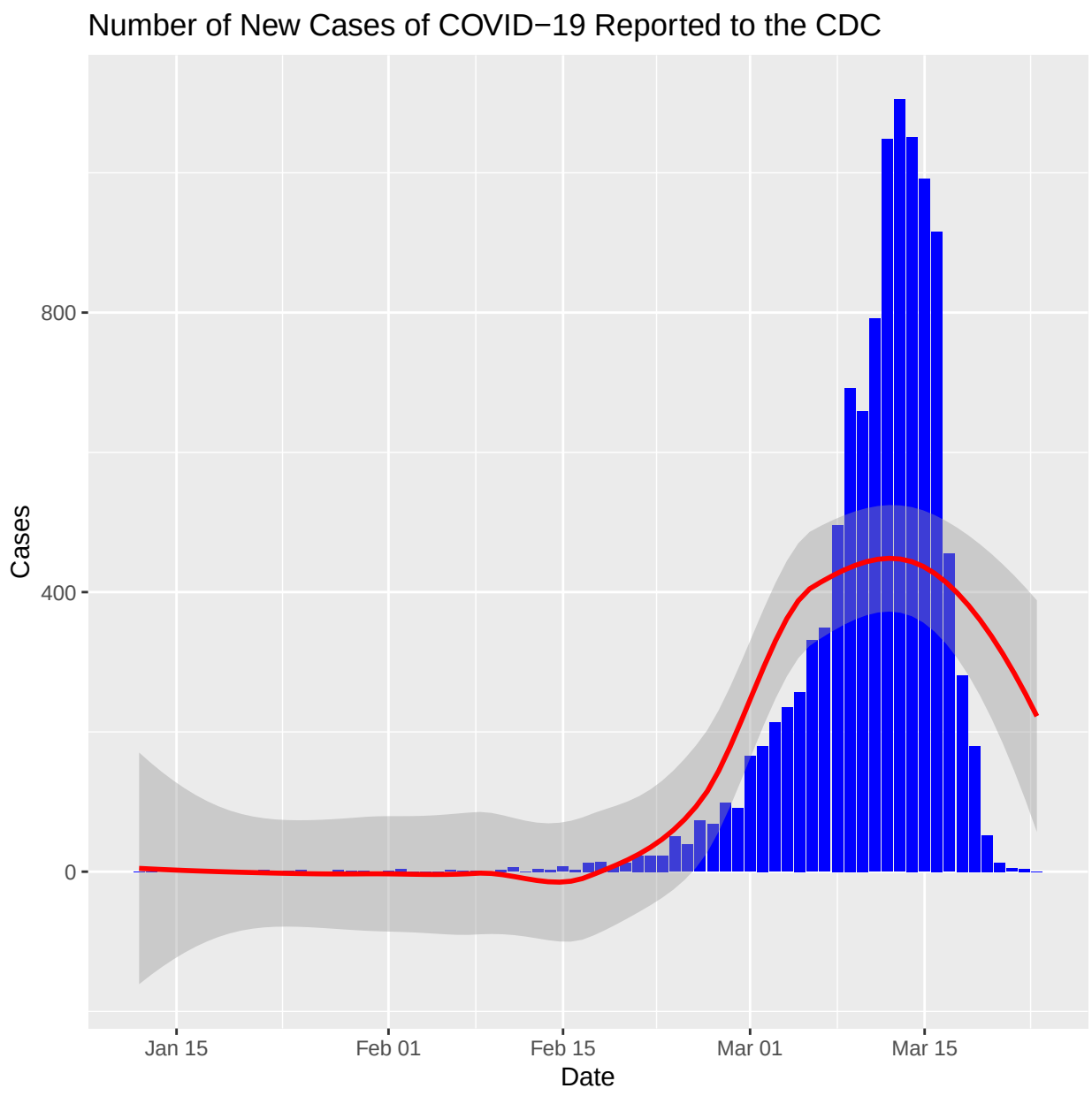

Figure 2: Epi curve 2, traditional

```
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'
```

Filter to remove incomplete reporting

remove dates on or after 5 March as this data may not be completely reported

```
cdc <- cdc %>%
     filter(Date < as.Date("2020-03-14"))
```

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,
```

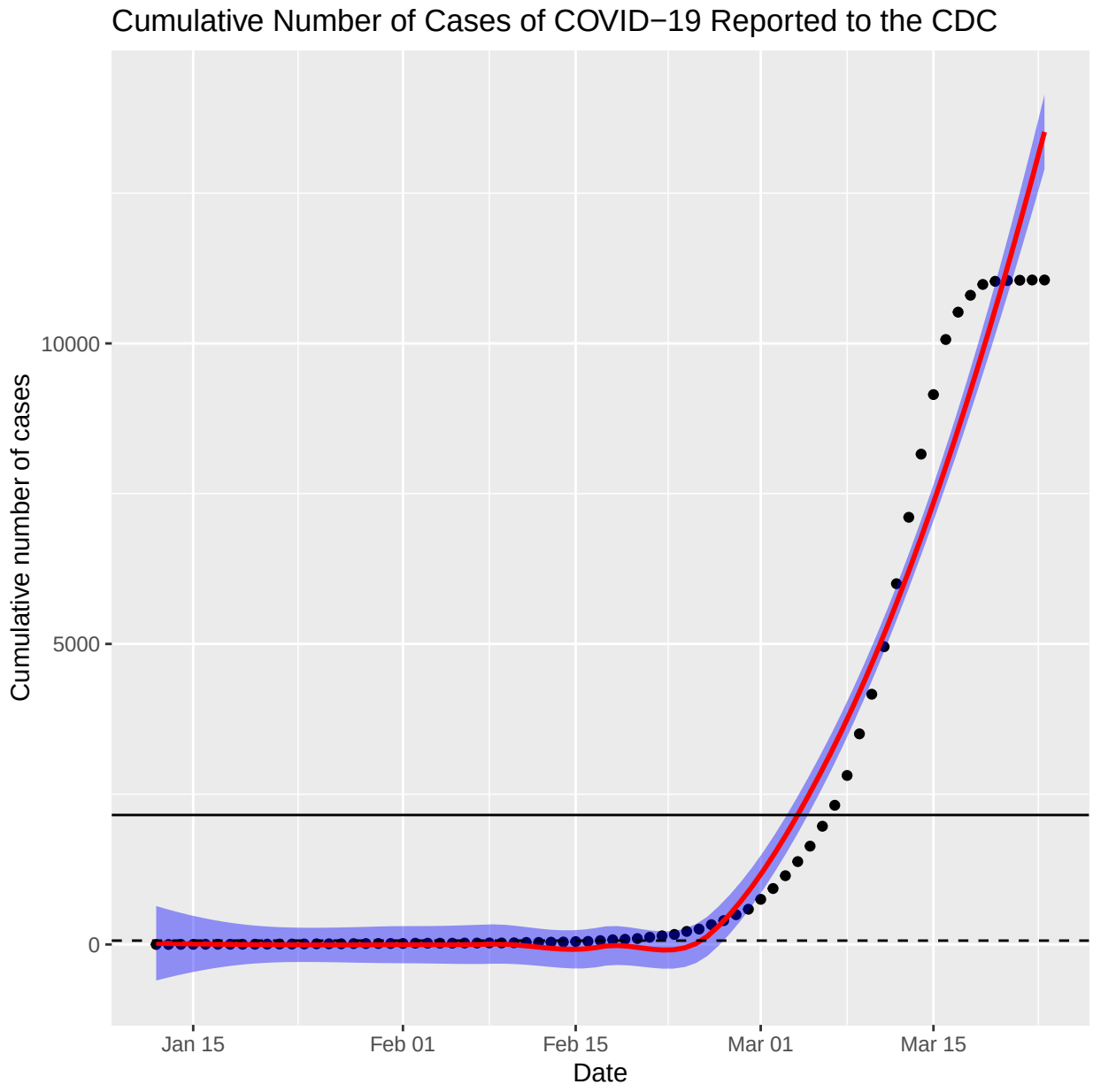

Figure 3: Cumulative cases

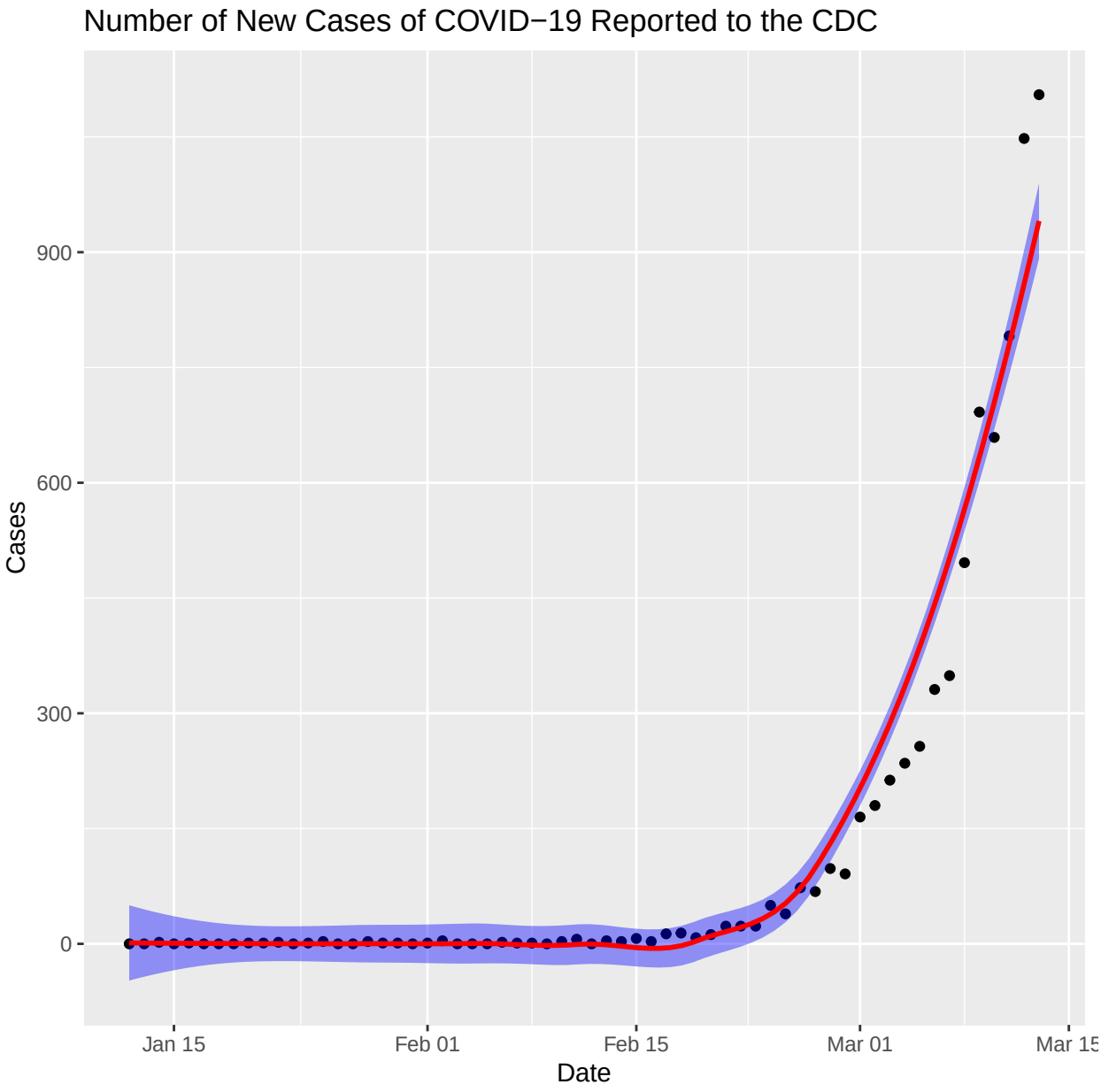

Figure 4: Epi curve 1

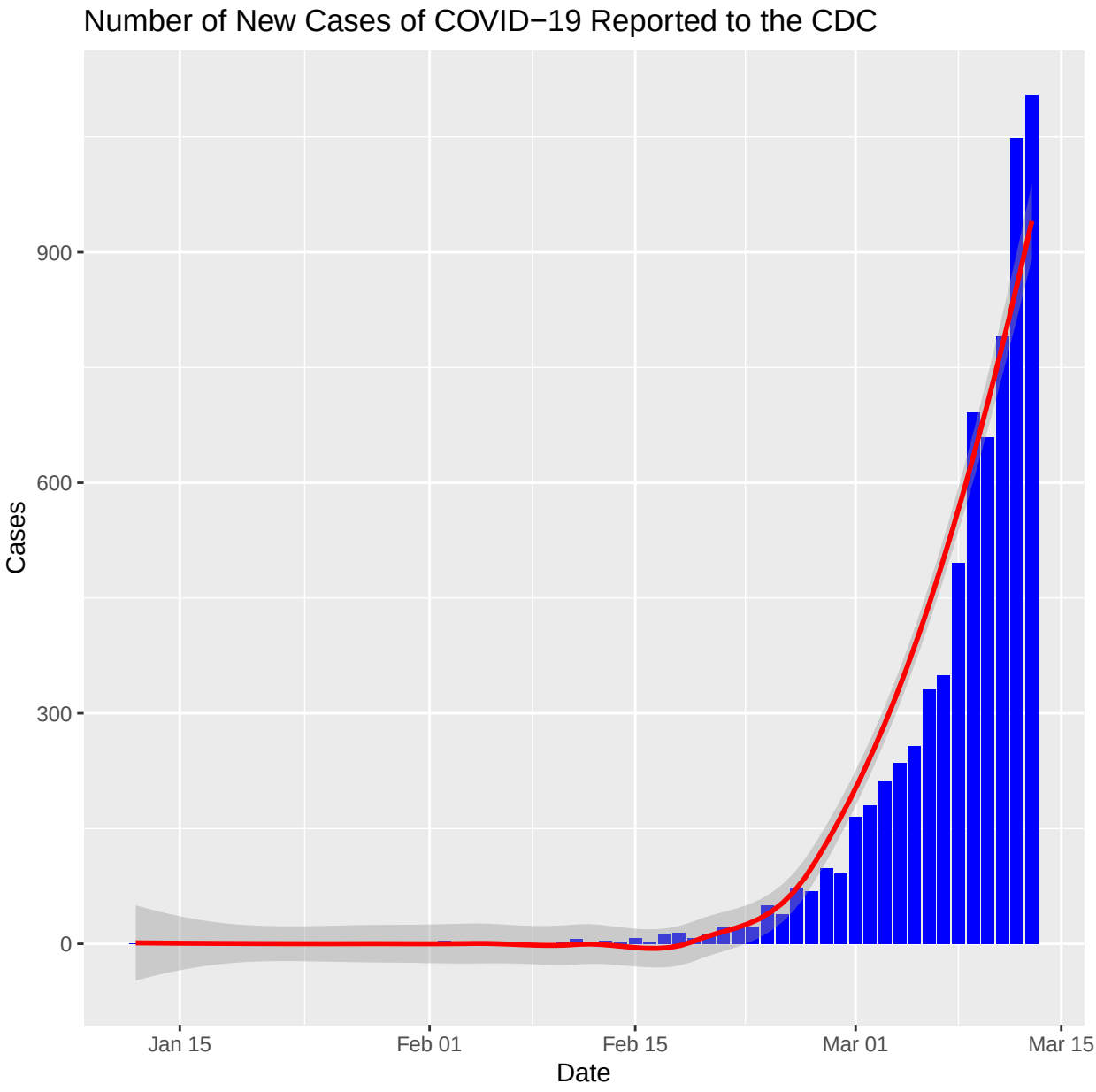

Figure 5: Epi curve 2, traditional

```
  #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'
```

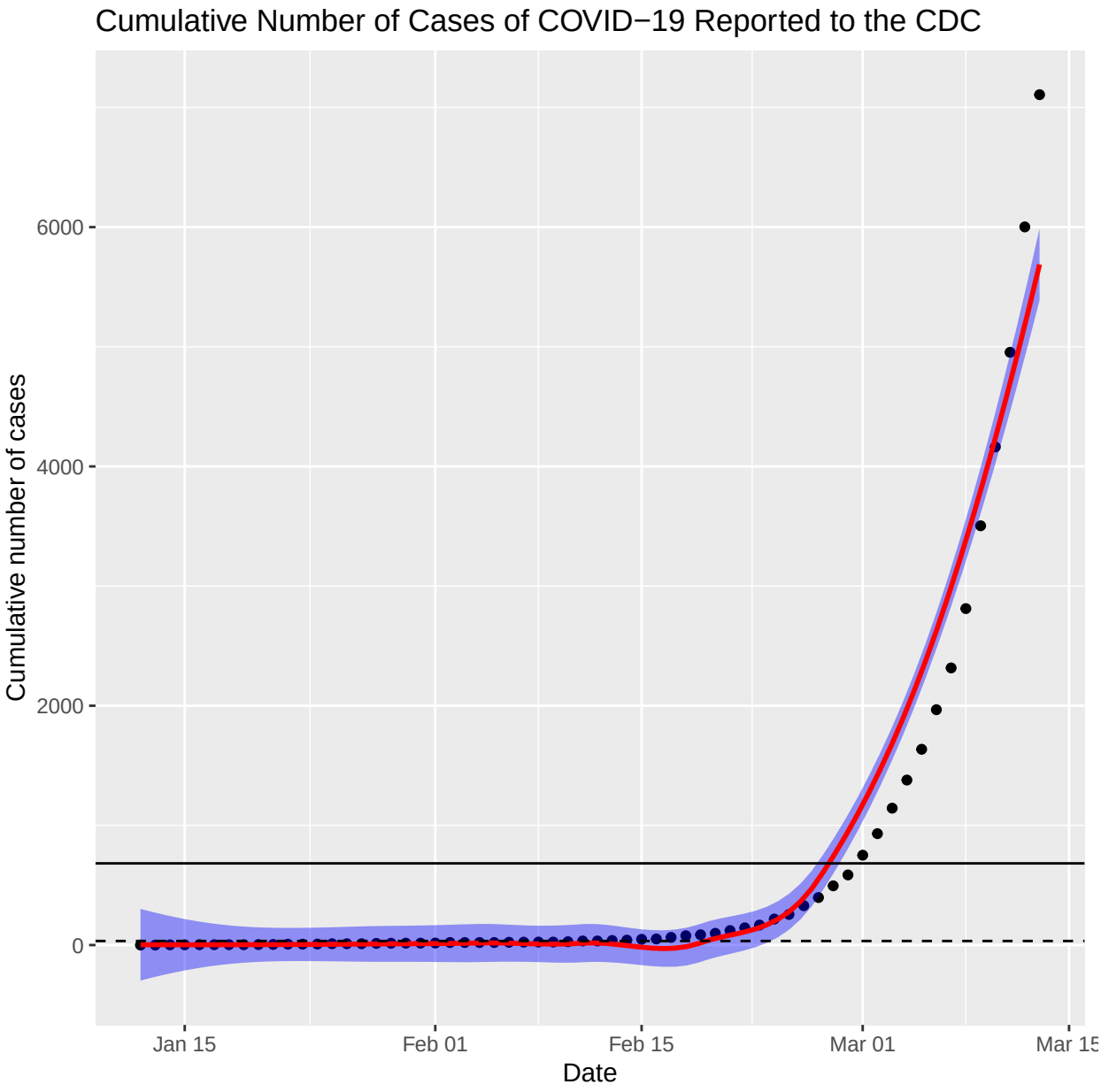

Figure 6: Cumulative cases