

EPI Info CDC

Nick Lauerman

3/16/2020

Contents

Libraries Used	1
Load CDC data	2
Read CSV File	2
Clean data and calculate cumulative number of cases	2
Data	2
Visualize all data	3
Filter to remove incomplete reporting	7
Visualize	7

Libraries Used

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## 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 ----- tidyverse_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	1	6
## 12	2020-01-23	0	6
## 13	2020-01-24	1	7
## 14	2020-01-25	3	10
## 15	2020-01-26	0	10
## 16	2020-01-27	0	10
## 17	2020-01-28	3	13
## 18	2020-01-29	1	14
## 19	2020-01-30	1	15
## 20	2020-01-31	0	15
## 21	2020-02-01	1	16
## 22	2020-02-02	1	17
## 23	2020-02-03	0	17
## 24	2020-02-04	0	17
## 25	2020-02-05	0	17
## 26	2020-02-06	2	19
## 27	2020-02-07	0	19

## 28	2020-02-08	1	20
## 29	2020-02-09	0	20
## 30	2020-02-10	4	24
## 31	2020-02-11	5	29
## 32	2020-02-12	0	29
## 33	2020-02-13	4	33
## 34	2020-02-14	3	36
## 35	2020-02-15	5	41
## 36	2020-02-16	3	44
## 37	2020-02-17	9	53
## 38	2020-02-18	10	63
## 39	2020-02-19	6	69
## 40	2020-02-20	10	79
## 41	2020-02-21	20	99
## 42	2020-02-22	22	121
## 43	2020-02-23	18	139
## 44	2020-02-24	43	182
## 45	2020-02-25	34	216
## 46	2020-02-26	52	268
## 47	2020-02-27	47	315
## 48	2020-02-28	66	381
## 49	2020-02-29	57	438
## 50	2020-03-01	103	541
## 51	2020-03-02	95	636
## 52	2020-03-03	121	757
## 53	2020-03-04	125	882
## 54	2020-03-05	119	1001
## 55	2020-03-06	139	1140
## 56	2020-03-07	120	1260
## 57	2020-03-08	140	1400
## 58	2020-03-09	194	1594
## 59	2020-03-10	172	1766
## 60	2020-03-11	174	1940
## 61	2020-03-12	122	2062
## 62	2020-03-13	95	2157
## 63	2020-03-14	55	2212
## 64	2020-03-15	35	2247
## 65	2020-03-16	12	2259
## 66	2020-03-17	8	2267
## 67	2020-03-18	0	2267
## 68	2020-03-19	0	2267

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'
```

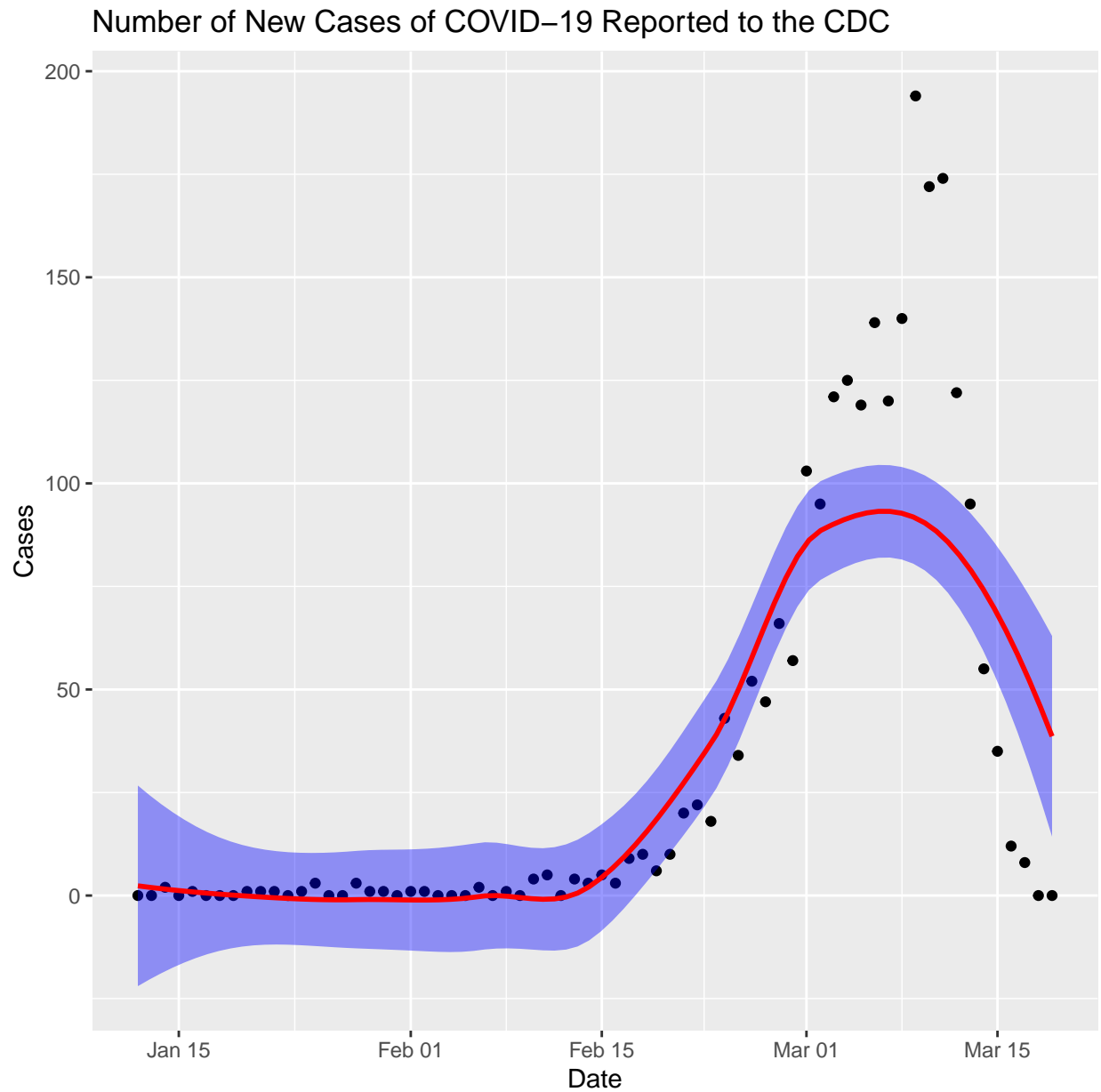


Figure 1: Epi curve 1

```

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'
```

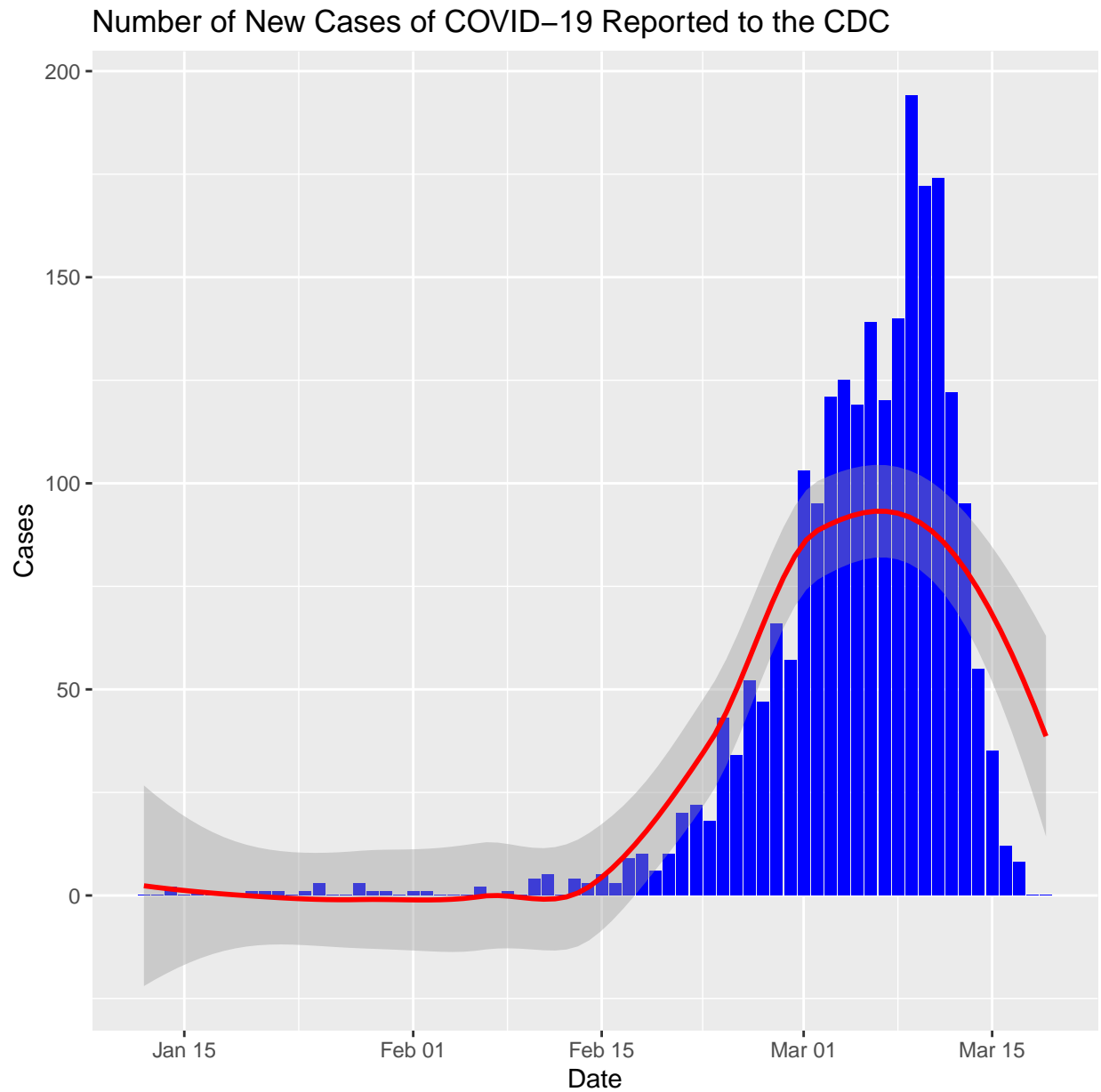


Figure 2: Epi curve 2, traditional

```

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'
```

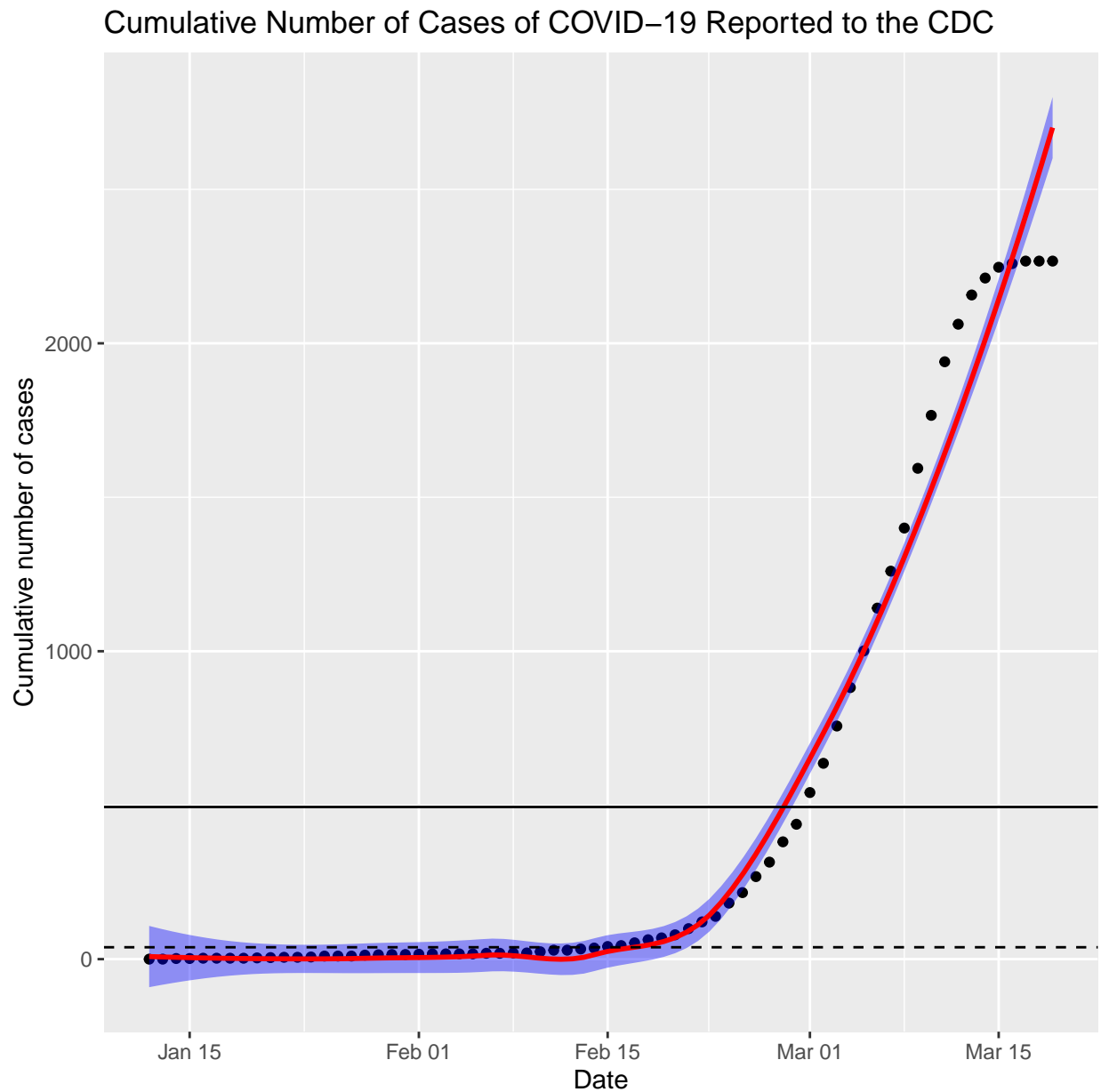


Figure 3: Cumulative cases

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("2020-03-12"))
```

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'
```

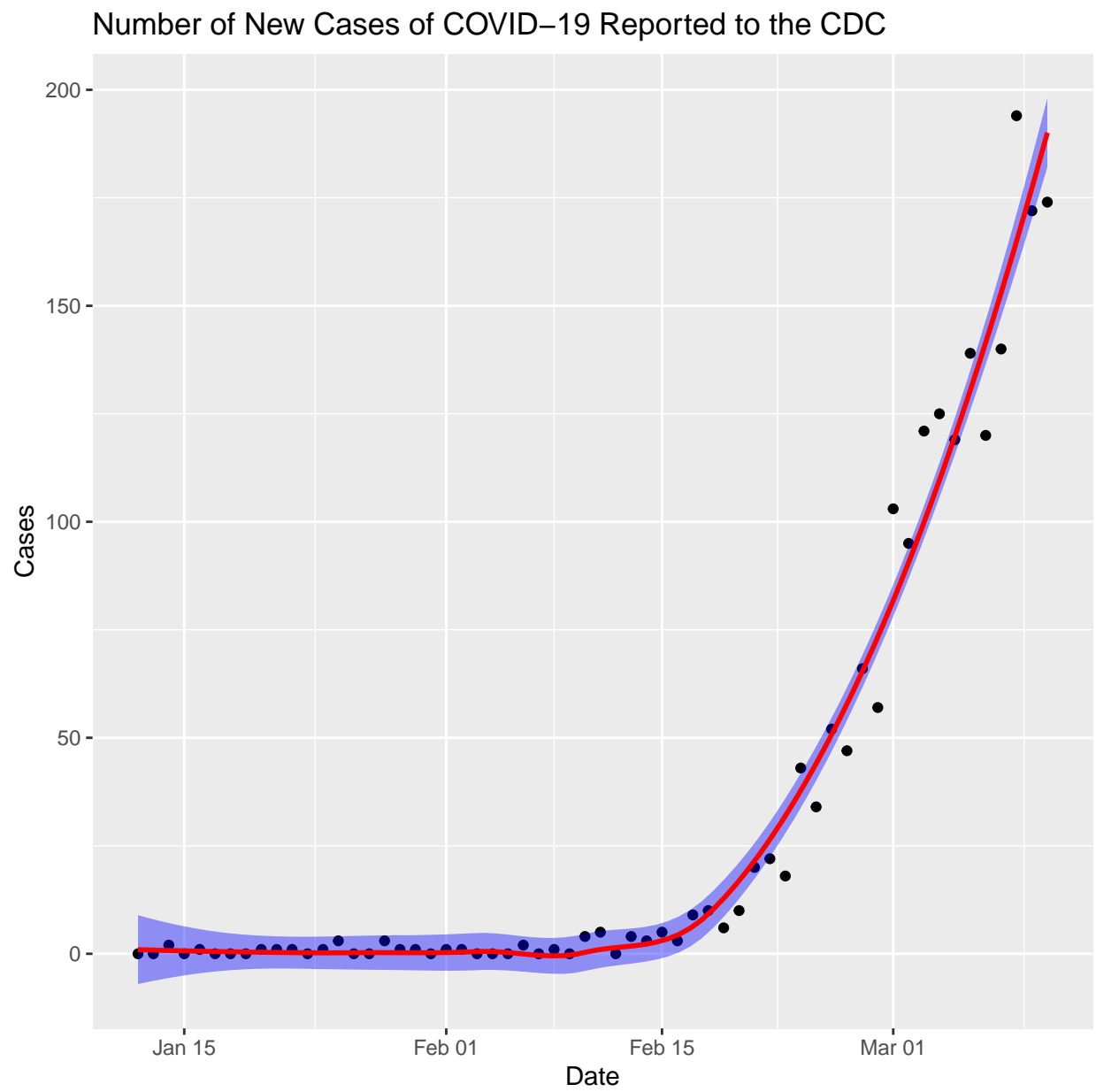


Figure 4: Epi curve 1

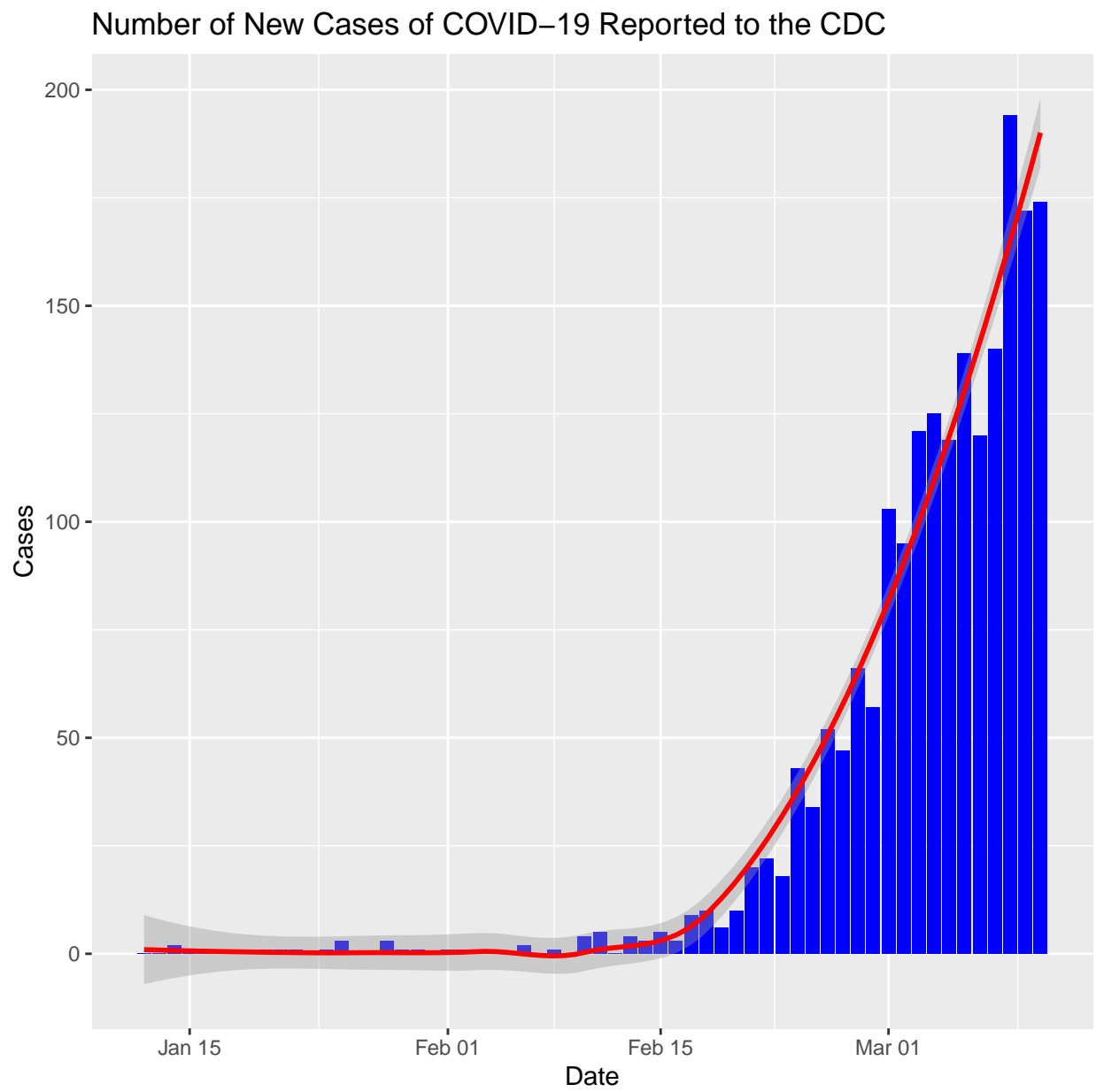


Figure 5: Epi curve 2, traditional

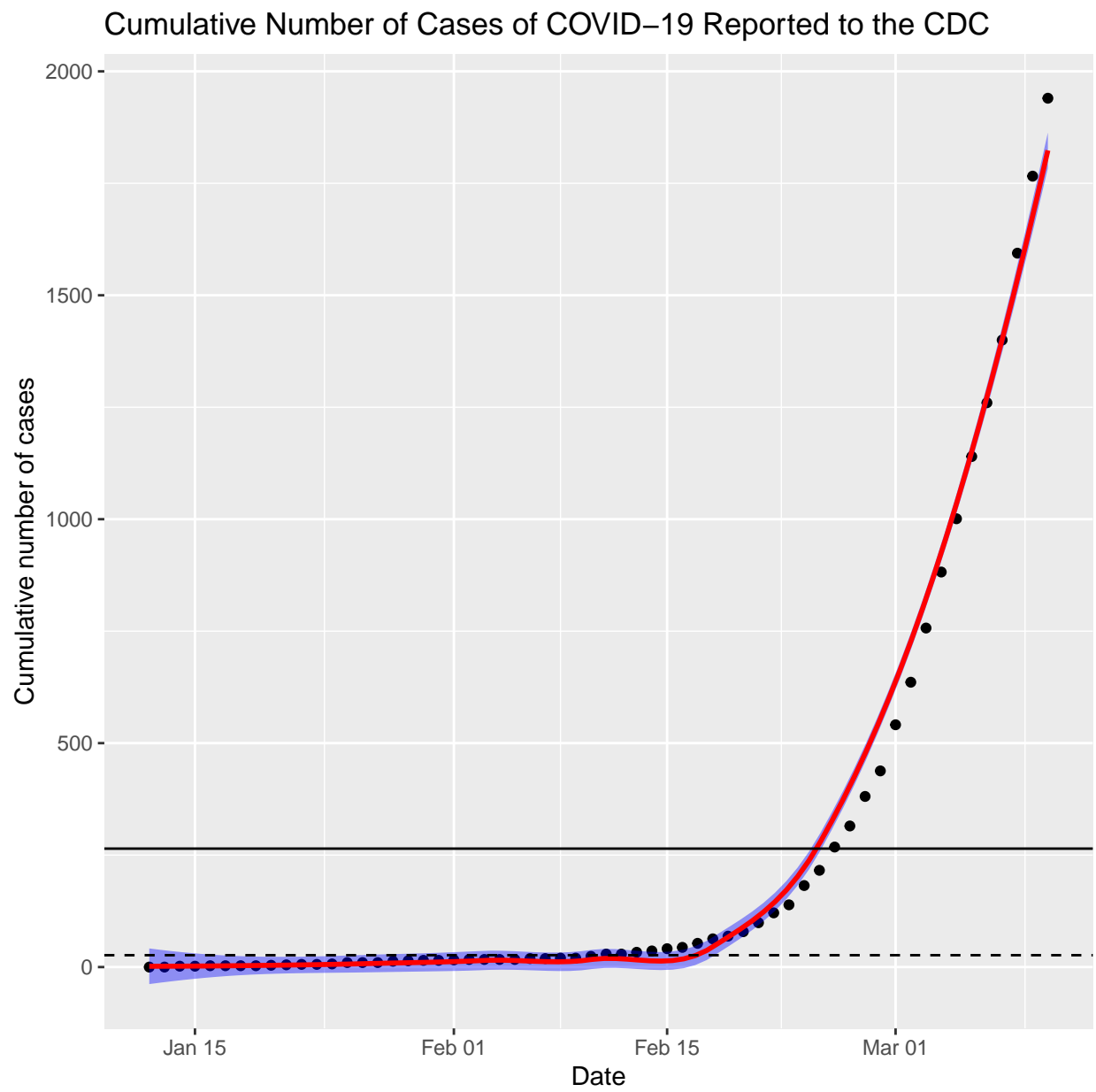


Figure 6: Cumulative cases