Bayesian Estimation of the COVID-19 Effective Reproduction Rate

Tuan Nguyen

5/7/2020

1. FUNCTION SMOOTHS THE DATA IN A WINDOW OF ONE-WEEK INTERVAL

Smooth.Case():
* Input: dataframe of observations the selected state's date, cases
* Output: dataframe of observations with the state's cases, smoothed cases, and date

```
library(dplyr)      # for mutate(), rename() of columns
library(magrittr)   # for pipe %>% operator
library(smoother)   # for smth()

Smooth.Cases <- function(Cases) {
  Cases %>%
  arrange(Date) %>%
  mutate(Cases_Smth=round(smth(Cases, window=7, tails=TRUE))) %>%
  select(Date, Cases, Cases_Smth)
}
```

---

2. FUNCTION PLOTS THE ORIGINAL AND THE SMOOTHED DATA

```
library(plotly)   # for interactive ggplotly()

Plot.Smth <- function(Smoothed_Cases) {
  plot <- Smoothed_Cases %>% ggplot(aes(x=Date, y=Cases)) +
    geom_line(linetype='dotted', color='#429890') +
    geom_line(aes(y=Cases_Smth), color='#E95D0F') +
    labs(title='Daily Confirmed Cases (Original & Smoothed)', x=NULL, y=NULL) +
    theme(plot.title=element_text(hjust=0.5, color='steelblue'))
}
```

---

3. FUNCTION COMPUTES THE EFFECTIVE REPRODUCTION RATE AND ITS LOG-LIKELIHOOD

Gamma = 1/serial interval
The serial interval of COVID-19 is defined as the time duration between a primary case-patient (infector) having symptom onset and a secondary case-patient (infectee) having symptom onset.
The mean interval was 3.96 days according to CDC sources:
https://wwwnc.cdc.gov/eid/article/26/6/20-0357_article (https://wwwnc.cdc.gov/eid/article/26/6/20-0357_article)

Comp.Log_Likelihood()
* Input: dataframe of observations with the selected state's date, cases, smoothed cases
* Output: dataframe of observations with the state's cases, smoothed cases, Rt, Rt's log-likelihood

```r
library(purrr)    # for map() and map2()
library(tidyr)    # for unnest()

RT_MAX <- 10      # the max value of Effective Reproduction Rate Rt
# Generate a set of RT_MAX * 100 + 1 Effective Reproduction Rate value Rt
rt_set <- seq(0, RT_MAX, length=RT_MAX * 100 + 1)

GAMMA <- 1/4

Comp.Log_Likelihood <- function(Acc_Cases) {
  likelihood <- Acc_Cases %>%
    filter(Cases_Smth > 0) %>%
    # Vectorize rt_set to form Rt column
    mutate(Rt=list(rt_set),
           # Compute lambda starting from the second to the last observation
           Lambda=map(lag(Cases_Smth, 1), ~ .x * exp(GAMMA * (rt_set - 1))),
           # Compute the log likelihood for every observation
           Log_Likelihood=map2(Cases_Smth, Lambda, dpois, log=TRUE)) %>%
    # Remove the first observation
    slice(-1) %>%
    # Remove Lambda column
    select(-Lambda) %>%
    # Flatten the table in columns Rt, Log_Likelihood
    unnest(Log_Likelihood, Rt)
}
```

4. FUNCTION COMPUTES THE POSTERIOR OF THE EFFECTIVE REPRODUCTION RATE

Comp.Posterior()

* Input: dataframe of observations with the selected state's date, cases, smoothed cases, Rt, Rt's log-likelihood
* Output: dataframe of observations with the state's cases, smoothed cases, Rt, Rt's posterior

```r
library(zoo)     # for rollapplyr()

Comp.Posterior <- function(likelihood) {
  likelihood %>%
  arrange(Date) %>%
  group_by(Rt) %>%
  # Compute the posterior for every Rt by a sum of 7-day log-likelihood
  mutate(Posterior=exp(rollapplyr(Log_Likelihood, 7, sum, partial=TRUE))) %>%
  group_by(Date) %>%
  # Normalize the posterior
  mutate(Posterior=Posterior/sum(Posterior, na.rm=TRUE)) %>%
  # Fill missing value of posterior with 0
  mutate(Posterior=ifelse(is.nan(Posterior), 0, Posterior)) %>%
  ungroup() %>%
  # Remove Log_Likelihood column
  select(-Log_Likelihood)
}
```

5. FUNCTION PLOTS POSTERIOR OF THE EFFECTIVE REPRODUCTION RATE

```
Plot.Posterior <- function(posteriors) {
  posteriors %>% ggplot(aes(x=Rt, y=Posterior, group=Date)) +
                 geom_line(color='#E95D0F', alpha=0.4) +
                 labs(title='Daily Posterior of Rt', subtitle=state) +
                 coord_cartesian(xlim=c(0.2, 5)) +
                 theme(plot.title=element_text(hjust=0.5, color='steelblue'))
}
```

6. FUNCTION ESTIMATES THE EFFECTIVE REPRODUCTION RATE

Estimate.Rt()

* Input: csv dataframe of observations with the selected state's cases, smoothed cases, Rt, Rt's posterior
* Output: dataframe of observations with the state's Rt, Rt_max, Rt_min

```
library(HDInterval)
Estimate.Rt <- function(posteriors) {
  posteriors %>%
  group_by(Date) %>%
  summarize(Rts_sampled=list(sample(rt_set, 10000, replace=TRUE, prob=Posterior)),
            Rt_MLL=rt_set[which.max(Posterior)]) %>%
  mutate(Rt_MIN=map_dbl(Rts_sampled, ~ hdi(.x)[1]),
         Rt_MAX=map_dbl(Rts_sampled, ~ hdi(.x)[2])) %>%
  select(-Rts_sampled)
}
```

7. FUNCTION PLOTS THE EFFECTIVE REPRODUCTION RATE'S APPROXIMATION

```
Plot.Rt <- function(Rt_estimated) {
  plot <- Rt_estimated %>%
            ggplot(aes(x=Date, y=Rt_MLL)) +
            geom_point(color='#429890', alpha=0.5, size=1) +
            geom_line(color='#E95D0F') +
            geom_hline(yintercept=1, linetype='dashed', color='red') +
            geom_ribbon(aes(ymin=Rt_MIN, ymax=Rt_MAX), fill='black', alpha=0.5) +
            labs(title='Estimated Effective Reproduction Rate Rt', x=NULL, y='Rt') +
            coord_cartesian(ylim=c(0, 4)) +
            theme(plot.title=element_text(hjust=0.5, color='steelblue'))
}
```

8. SET UP FOR COMPUTATIONS

```
# Read the local data file into dataframe
library(readr)
cv19 <- read_csv('./state.csv')
library(knitr)
library(kableExtra)
col <- c('Date','NY','CA','MI','LA','NH','IN','AR','ID','AZ','HI','AK','FL','TX')
kable(cv19[seq(0,80,5), col],
      caption='Samples of Daily Counts of COVID-19 Cases in Some States') %>%
kable_styling(full_width=F,  position='center',
              bootstrap_options=c('striped', 'bordered', 'hover'))
```

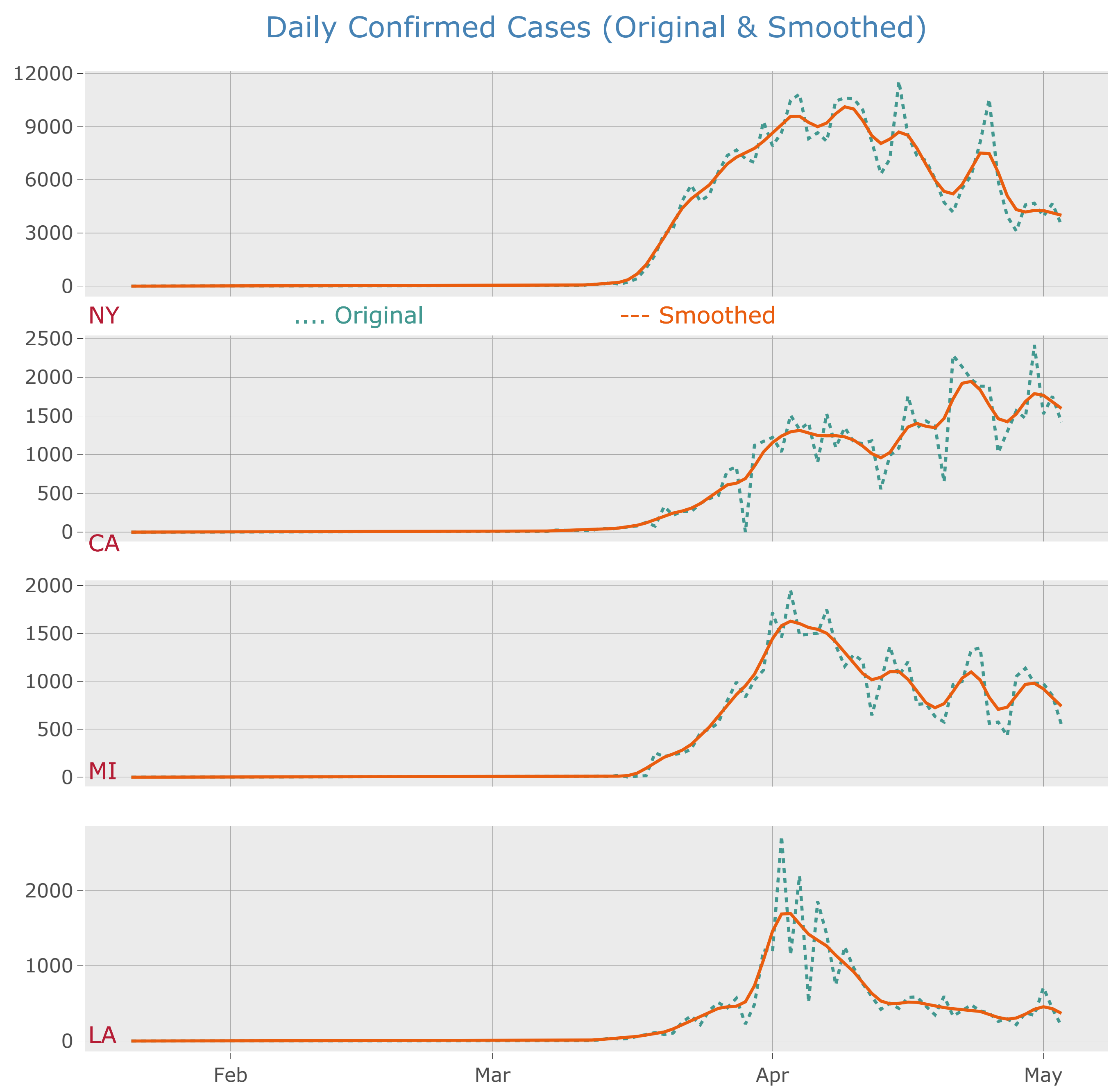
Samples of Daily Counts of COVID-19 Cases in Some States

Date	NY	CA	MI	LA	NH	IN	AR	ID	AZ	HI	AK	FL	TX
2020-01-30	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-02-15	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-02-28	0	1	0	0	0	0	0	0	0	0	0	0	0
2020-03-04	9	4	0	0	0	0	0	0	0	0	0	0	1
2020-03-09	36	19	0	1	0	2	0	0	0	0	0	2	1
2020-03-14	192	39	8	41	0	4	0	4	3	2	0	39	12
2020-03-19	1770	77	254	112	5	23	29	12	17	10	3	104	60
2020-03-24	4790	369	463	216	7	106	21	23	92	13	6	185	58
2020-03-29	7195	0	836	225	44	282	17	49	146	24	12	912	500
2020-04-03	10482	1510	1953	1147	61	398	55	122	171	34	10	1260	661
2020-04-08	10453	1092	1376	746	41	436	3	22	151	25	13	1194	1091
2020-04-13	6337	554	997	421	35	308	130	27	163	5	5	1124	422
2020-04-18	7090	1435	768	462	55	487	44	13	212	21	5	739	889
2020-04-23	6244	1973	1325	481	82	601	207	34	310	4	2	1072	875
2020-04-28	3110	1567	1052	218	72	627	58	35	232	2	6	708	874
2020-05-03	3438	1419	547	200	89	638	59	0	276	0	3	615	1026

```
# Select a list of the U.S. states for modeling & computations
states <- col[2:5]          # Select New York, California, Michigan, Louisiana
```

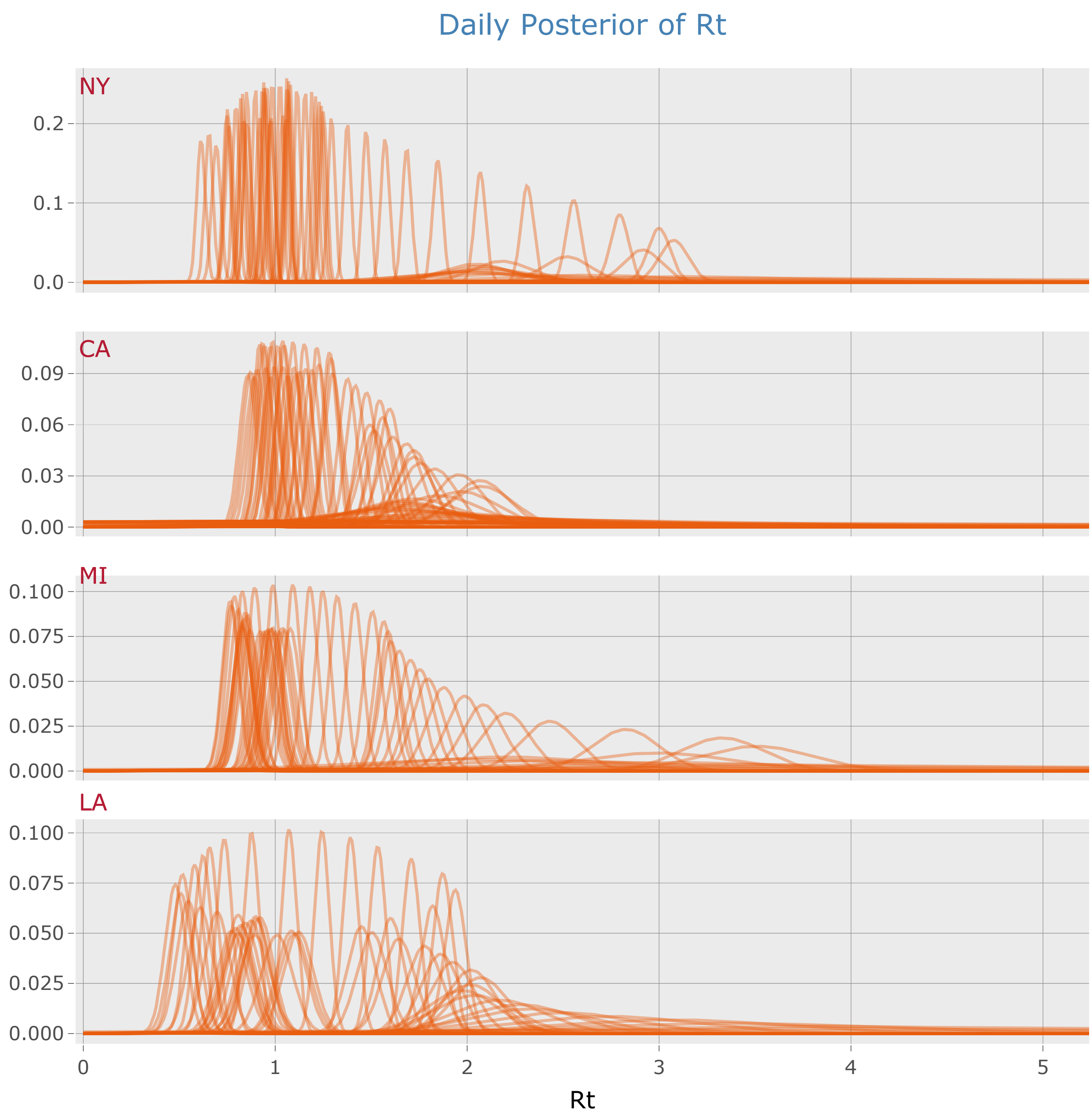
9. COMPUTE THEN PLOT THE ORIGINAL AND SMOOTHED CASES

```
df_cv19 <- list()                          # initialize list of plots for each of states
for (i in 1:length(states)) {
  state <- states[i]
  df_S <- cv19 %>% select(Date, state) %>% rename(Cases=state) %>% Smooth.Cases()
  gplot <- df_S %>% Plot.Smth()
  plot <- ggplotly(gplot) %>% add_annotations(text=state,font=list(size=14, color='#B51C35'),
                                              xref='paper', yref='paper', x=0, y=0, showarrow=FALS
E)
  if (i == 1) {
    plot <- plot %>% add_annotations(text='.... Original', font=list(size=14, color='#429890'),
                                     xref='paper', yref='paper', x=0.2, y=0, showarrow=FALSE) %>%
                     add_annotations(text='--- Smoothed', font=list(size=14, color='#E95D0F'),
                                      xref='paper', yref='paper', x=0.6, y=0, showarrow=FALSE)
  }
  df_cv19[[i]] <- plot
}
df_cv19 %>% subplot(nrows=length(states), shareX=TRUE)
```



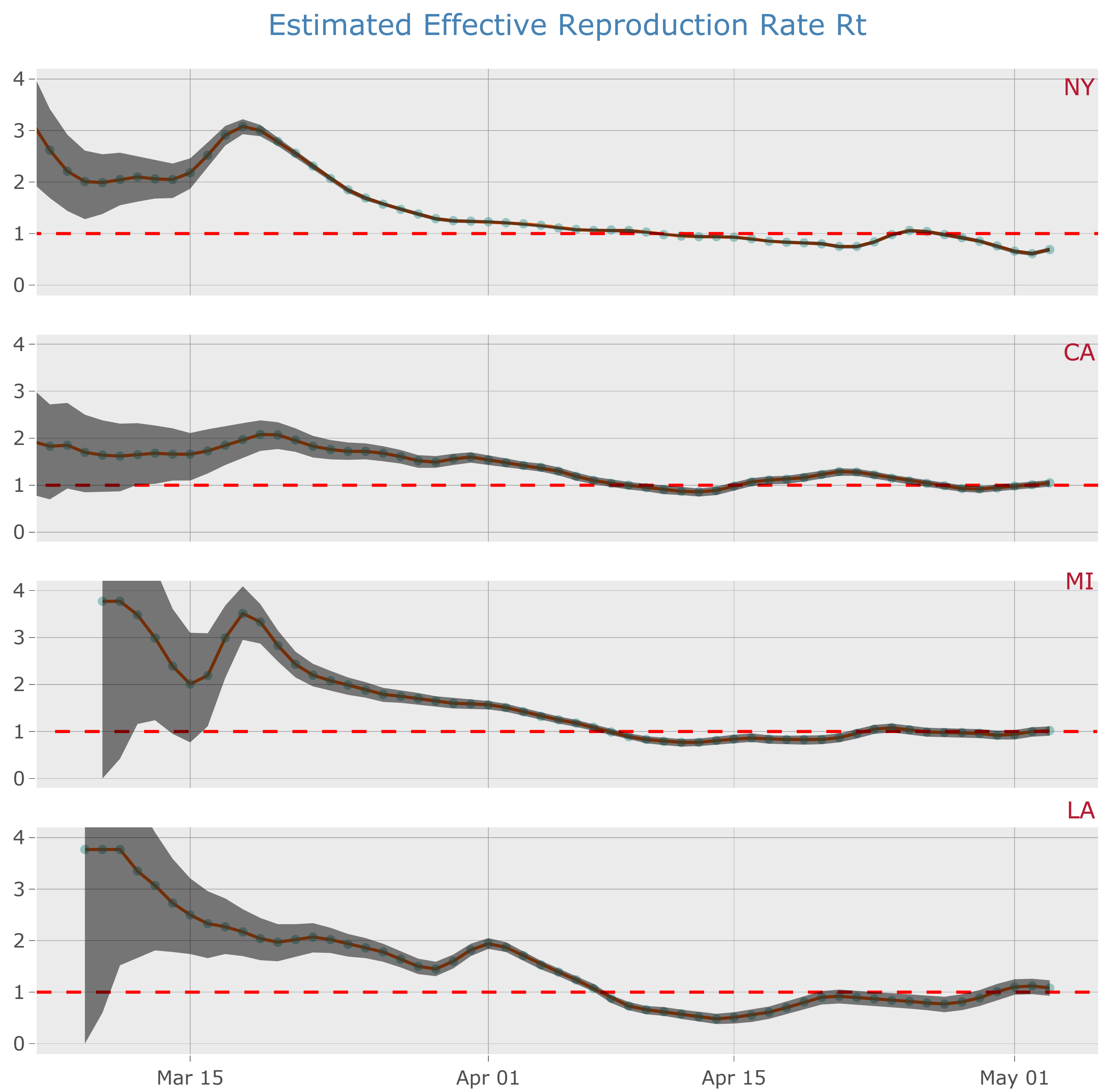
10. PLOT THE POSTERIORS OF RT FOR EACH STATE IN THE LIST

```
df_cv19 <- list()                          # reset list of plots for each of states
for (i in 1:length(states)) {
  state <- states[i]
  df_P <- cv19 %>% select(Date, state) %>% rename(Cases=state) %>% Smooth.Cases %>%
          Comp.Log_Likelihood() %>% Comp.Posterior()
  gplot <- df_P %>% Plot.Posterior()
  plot <- ggplotly(gplot) %>% add_annotations(text=state,
                                              font=list(size=14, color='#B51C35'),
                                              xref='paper', yref='paper', x=0, y=1, showarrow=FALS
E)
  df_cv19[[i]] <- plot
}
df_cv19 %>% subplot(nrows=length(states), shareX=TRUE)
```



11. COMPUTE AND PLOT THE MAX, MIN, AND MOST-LIKELY VALUES OF EFFECTIVE REPRODUCTION RATE FOR EACH STATE IN THE LIST

```
df_cv19 <- list()                  # reset list of plots for each of states
Rt_estimate_list <- list()         # initialize a list of estimated Rt dataframe for each state
for (i in 1:length(states)) {
  state <- states[i]
  df_R <- cv19 %>% select(Date, state) %>% rename(Cases=state) %>% Smooth.Cases %>%
          Comp.Log_Likelihood() %>% Comp.Posterior() %>% Estimate.Rt()
  Rt_estimate_list[[state]] <- df_R
  gplot <- df_R %>% Plot.Rt()
  plot <- ggplotly(gplot) %>%
          add_annotations(text=state,font=list(size=14, color='#B51C35'),
                          xref='paper', yref='paper', x=1, y=1, showarrow=FALSE)
  df_cv19[[i]] <- plot
}
df_cv19 %>% subplot(nrows=length(states), shareX=TRUE)
```



12. TABLE VALUES OF EFFECTIVE REPRODUCTION RATE FOR EACH STATE IN THE LIST

*****************|————New York————|————California————|————Michigan————|————Louisiana————-|

Date	Rt_MLL	Rt_MIN	Rt_MAX	Rt_MLL	Rt_MIN	Rt_MAX	Rt_MLL	Rt_MIN	Rt_MAX	Rt_MLL	Rt_MIN	Rt_MAX
2020-03-24	1.85	1.79	1.89	1.72	1.54	1.91	1.99	1.78	2.15	1.93	1.69	2.13
2020-03-25	1.69	1.64	1.73	1.72	1.55	1.89	1.88	1.72	2.05	1.86	1.66	2.05
2020-03-26	1.57	1.53	1.61	1.68	1.51	1.83	1.79	1.63	1.93	1.78	1.59	1.94
2020-03-27	1.47	1.44	1.51	1.61	1.46	1.75	1.75	1.61	1.88	1.64	1.48	1.80
2020-03-28	1.38	1.34	1.41	1.52	1.37	1.64	1.70	1.57	1.82	1.50	1.35	1.65
2020-03-29	1.29	1.25	1.32	1.49	1.37	1.62	1.65	1.52	1.75	1.45	1.31	1.59
2020-03-30	1.25	1.21	1.28	1.56	1.43	1.67	1.60	1.49	1.70	1.60	1.46	1.73
2020-03-31	1.24	1.21	1.27	1.60	1.48	1.70	1.59	1.48	1.68	1.82	1.70	1.94
2020-04-01	1.23	1.20	1.26	1.54	1.43	1.64	1.57	1.47	1.65	1.94	1.84	2.05
2020-04-02	1.21	1.18	1.24	1.48	1.38	1.57	1.51	1.42	1.59	1.87	1.78	1.97
2020-04-03	1.19	1.16	1.22	1.42	1.32	1.50	1.42	1.33	1.49	1.71	1.61	1.79
2020-04-04	1.16	1.12	1.18	1.38	1.29	1.46	1.32	1.25	1.41	1.53	1.44	1.61
2020-04-05	1.11	1.08	1.14	1.30	1.21	1.38	1.25	1.17	1.32	1.39	1.31	1.46
2020-04-06	1.08	1.04	1.10	1.19	1.10	1.27	1.18	1.10	1.25	1.24	1.16	1.31
2020-04-07	1.06	1.04	1.09	1.10	1.02	1.18	1.09	1.01	1.16	1.07	1.01	1.16
2020-04-08	1.07	1.03	1.09	1.04	0.96	1.12	0.99	0.92	1.06	0.88	0.79	0.94
2020-04-09	1.06	1.03	1.09	1.00	0.91	1.07	0.89	0.82	0.96	0.73	0.65	0.81
2020-04-10	1.03	1.00	1.05	0.95	0.87	1.03	0.83	0.75	0.90	0.66	0.57	0.73
2020-04-11	0.98	0.96	1.01	0.91	0.81	0.98	0.79	0.71	0.86	0.62	0.54	0.71
2020-04-12	0.95	0.93	0.98	0.87	0.79	0.96	0.77	0.68	0.84	0.58	0.48	0.66
2020-04-13	0.94	0.91	0.97	0.86	0.76	0.93	0.77	0.69	0.85	0.52	0.42	0.61
2020-04-14	0.94	0.91	0.97	0.89	0.79	0.97	0.81	0.72	0.89	0.48	0.38	0.58
2020-04-15	0.93	0.90	0.96	0.98	0.89	1.06	0.84	0.76	0.93	0.51	0.39	0.61
2020-04-16	0.90	0.86	0.92	1.07	0.98	1.15	0.86	0.78	0.96	0.55	0.42	0.66
2020-04-17	0.85	0.82	0.88	1.11	1.02	1.19	0.84	0.74	0.92	0.61	0.48	0.72
2020-04-18	0.83	0.80	0.86	1.13	1.03	1.20	0.83	0.72	0.91	0.70	0.57	0.82
2020-04-19	0.82	0.79	0.85	1.16	1.08	1.24	0.83	0.72	0.91	0.81	0.67	0.93
2020-04-20	0.80	0.77	0.84	1.23	1.14	1.30	0.83	0.73	0.92	0.90	0.76	1.02
2020-04-21	0.75	0.71	0.78	1.29	1.20	1.36	0.87	0.77	0.96	0.92	0.78	1.05
2020-04-22	0.75	0.71	0.78	1.28	1.20	1.35	0.96	0.85	1.05	0.90	0.76	1.03
2020-04-23	0.84	0.80	0.87	1.22	1.15	1.29	1.05	0.95	1.14	0.87	0.73	1.00
2020-04-24	0.98	0.94	1.01	1.15	1.08	1.22	1.08	0.98	1.17	0.84	0.70	0.98
2020-04-25	1.06	1.02	1.09	1.09	1.02	1.16	1.04	0.93	1.12	0.82	0.68	0.96
2020-04-26	1.04	1.00	1.07	1.04	0.96	1.10	0.99	0.89	1.08	0.79	0.65	0.94
2020-04-27	0.98	0.94	1.01	0.99	0.91	1.05	0.98	0.88	1.07	0.77	0.61	0.91
2020-04-28	0.92	0.89	0.96	0.93	0.86	1.00	0.98	0.88	1.07	0.81	0.65	0.96
2020-04-29	0.85	0.82	0.89	0.92	0.84	0.98	0.96	0.86	1.05	0.89	0.73	1.04
2020-04-30	0.76	0.71	0.79	0.94	0.87	1.01	0.92	0.83	1.02	1.01	0.84	1.16
2020-05-01	0.66	0.61	0.69	0.98	0.90	1.04	0.94	0.83	1.03	1.10	0.95	1.25
2020-05-02	0.61	0.57	0.65	1.01	0.94	1.08	1.00	0.89	1.09	1.12	0.96	1.26
2020-05-03	0.69	0.65	0.73	1.05	0.97	1.11	1.02	0.91	1.11	1.08	0.93	1.23