IC-MODEL1

Saier Gong

08/08/2020

IMPORT DATA

```
data<-read.csv(file="US-Coronavirus-data.csv",header = TRUE) %>%
  mutate(date=as.Date(date)) %>%
  mutate(daily.D=D-c(0,D[-length(D)])) %>% # to count the daily deaths
  filter(!is.na(C)) %>%
  filter(date>=as.Date('2020-02-29'))

D<-data %>%
  select(date,I,D,daily.D) %>%
  filter(date<=as.Date("2020-07-31"))
```

```
trans<-data$I[-1]/data$I[-nrow(data)] #from March to May
#tran.D<-data$D[-1]/data$D[-nrow(data)]
```

In our model, we define the transmission rate as: $rate_{today} = \frac{ExistingInfectious_{today}}{ExistingInfectious_{yesterday}}$ This is the transimission rate of each day in our training dataset, we will use this vector to do the following model.

MODEL

After reading the references(https://mrc-ide.github.io/covid19-short-term-forecasts/index.html#methods), we plan to write the model based on our understanding. Since in this model, we need to use the prior distriburion of R_t, we decided to use the Gamma prior distribution of R_t with $\mu = mean(TransmissionRate_{last7days}), \sigma = sd(TransmissionRate_{last7days})$. And we used the transmission rate of the last day in our training set as the start value of R_t in the MCMC process.

```
library(purrr)
Rt<- mean(tail(trans,1))
mu<-mean(tail(trans,7))
v<-var(tail(trans,7))
rate=mu/v
shape=mu*rate


daily<-D$daily.D
l<-length(daily)

###posterior of Rt,tao
R<-Rt
```

```r
A<-c()
#####MCMC
for (i in 1:2000) {
  #1st given Rt
  daily[l-6]<-rpois(1,lambda = Rt * mean(head(daily,l-7)) )
  daily[l-5]<-rpois(1,lambda = Rt * mean(head(daily,l-6)) )
  daily[l-4]<-rpois(1,lambda = Rt * mean(head(daily,l-5)) )
  daily[l-3]<-rpois(1,lambda = Rt * mean(head(daily,l-4)) )
  daily[l-2]<-rpois(1,lambda = Rt * mean(head(daily,l-3)) )
  daily[l-1]<-rpois(1,lambda = Rt * mean(head(daily,l-2)) )
  daily[l-0]<-rpois(1,lambda = Rt * mean(head(daily,l-1)) )

  shape<-shape+sum(tail(daily,7))
  rate<-rate+mean(head(daily,l-6-1))+mean(head(daily,l-5-1))+mean(head(daily,l-4-1))+mean(head(daily,l-
  Rt<-rgamma(1,shape=shape,rate=rate)

  R<-append(R,Rt)
  A<-rbind(A,tail(daily,7))


}



R.post<-mean(tail(R,500))
R.post
```

```
## [1] 1.004606
```

```r
w.exam.post<-colMeans(A)
w.exam.post
```

```
## [1] 999.0550 998.1255 998.3105 997.9435 998.8655 998.2090 998.7100
```

```r
daily[(l-6):l]<-w.exam.post
```

So $R_{t.post}$ is the estimated value of R_t that wil remain constant in the shorterm forecasting, the vector $w.exam.post$ is the estimated value of daily deaths in part of the training group (from 7/25 to 7/31). We will count the estimated total deaths in these 7 days and calculate the MSE and RMSE.

MSE & RMSE

```r
library(purrr)
total<-map_dbl(1:length(daily),function(x) sum(head(daily,x)))
tail(total,7)
```

```
## [1] 147073.1 148071.2 149069.5 150067.4 151066.3 152064.5 153063.2
```

```r
mse<-mean((tail(total,7)-tail(D$D,7))^2)
mse
```

```
## [1] 226894
```

```r
rmse<-sqrt(mse)
rmse
```

```
## [1] 476.3339
```

```r
#(l1<-mean(abs(tail(total,7)-tail(D$D,7))))
```

SHORTERM FORECASTING (8/1-8/7)

```r
####forcast of the following 7 days of daily deaths
t<-7
w<-c()

for (iter in 1:2000) {
  daily<-D$daily.D
  l<-length(daily)

  for (i in (l+1):(l+t)) {
    d<-rpois(1,R.post* (mean(head(daily,i-1))) )
    daily[i]<-d
  }

#daily

  total.new<-map_dbl(1:length(daily),function(x) sum(head(daily,x)))

  w.fore<-tail(total.new,t)
  w<-rbind(w,w.fore)

}

answer=colMeans(tail(w,500))

temp<-seq(from=as.Date('2020-08-01'),by='day',length.out = t)
(pred=tibble(date=temp,`prediction in total deaths`=answer))
```

```
## # A tibble: 7 x 2
##   date       `prediction in total deaths`
##   <date>                            <dbl>
## 1 2020-08-01                      154817.
## 2 2020-08-02                      155821.
## 3 2020-08-03                      156825.
## 4 2020-08-04                      157826.
## 5 2020-08-05                      158828.
## 6 2020-08-06                      159832.
## 7 2020-08-07                      160837.
```