Oct19

2022-10-19

PLS closed-form solution

1. Deriving the closed form solution for PLS with MSE loss and L2 penalty on consecutive smoothing R_t. Starting from the objectives in the matrix form.

$\|I - RW\|_2^2 + \lambda \|R_{t+1} - R_t\|_2^2$

=> $\|I - RW\|_2^2 + \lambda \|DR\|_2^2$

where $D \in R^{(d-1)*d}$ is the difference matrix, with all 1's on the diagonal, and -1 on the off-diagonal above.

where W is a square matrix with diagonol being the vector $w_t = \sum_{a=1}^{t} I_{t-a} s_a$

=> $W^T R^T RW - 2W^T R^T I + I^T I + \lambda R^T D^T DR$

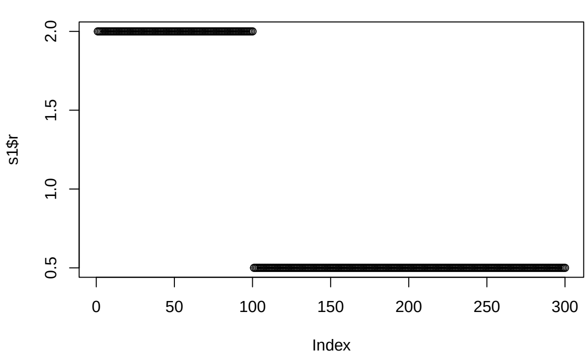
Taking derivative and equate to 0

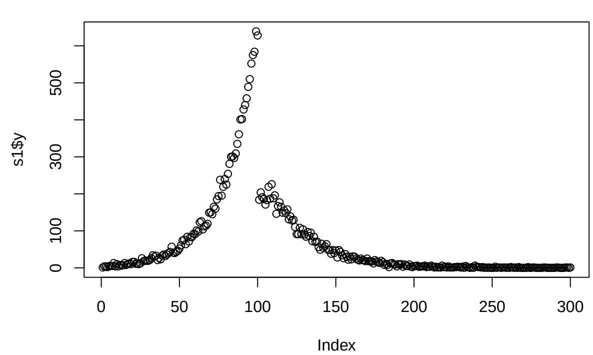
=> $2W^T R^T W - 2W^T I + 2\lambda R^T D^T D = 0$

=> $R^T (W^T W + \lambda D^T D) = W^T I$

=> $R^T = (W^T W + \lambda D^T D)^{-1} W^T I$

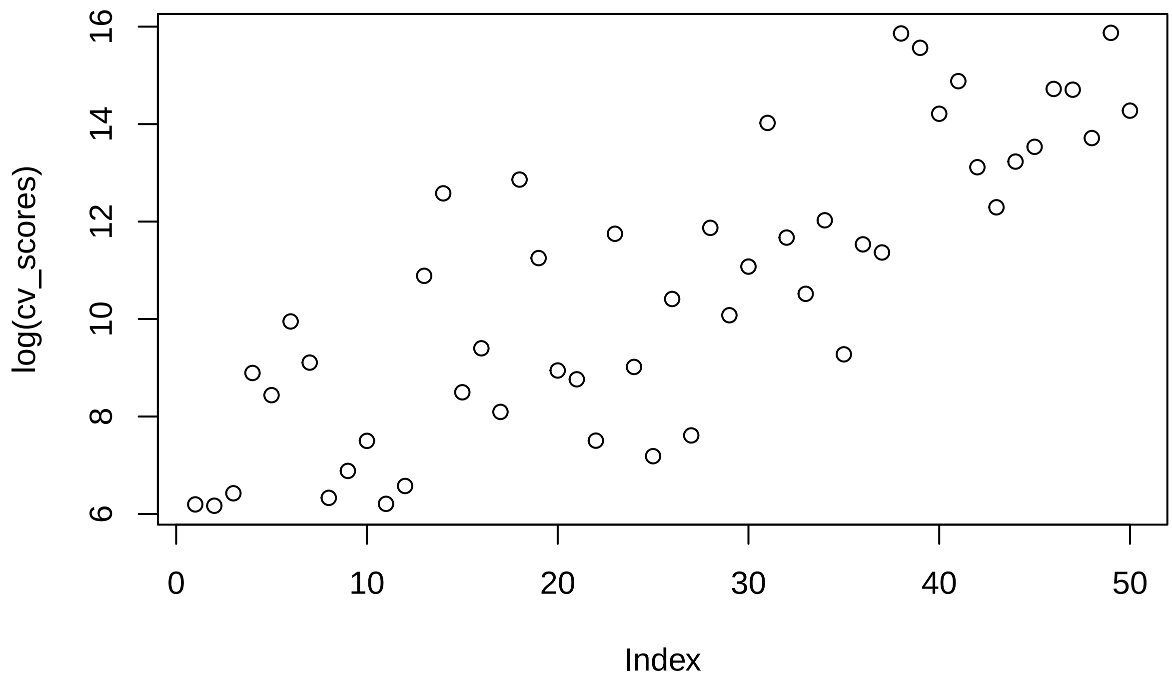
```
### Import s1
s1 <- read.csv("../data/processed/a.csv")

### Plotting
plot(s1$r)
plot(s1$y)
```



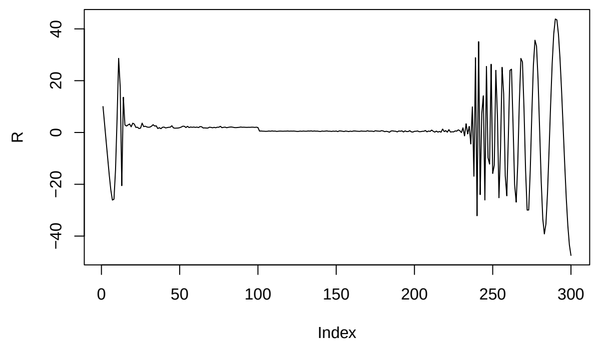


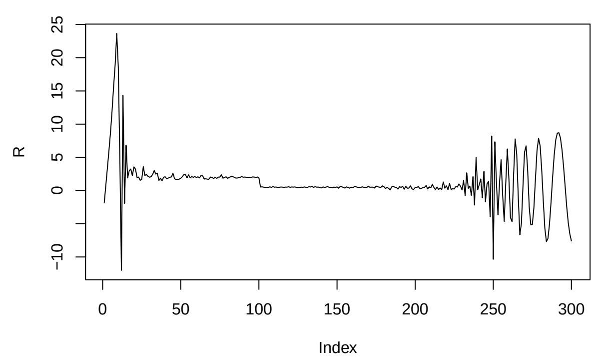
```r
r1_length <- nrow(s1)

### Construct difference matrix
D = diag(r1_length)
D[row(D) == col(D)-1] = -1
D = D[1:(nrow(D)-1),]


### Construct the Iwt vector
source("../function/get_iwt.R")
source("../function/disc_gamma.R")
source("../constant/constant.R")

s1_iwt <- get_iwt(s1$y, disc_gamma(1:nrow(s1), sid_ebola_shape, sid_ebola_scale))
W <- diag(s1_iwt)
Y <- s1$y

# R = solve((t(W)%*%W-lambda*t(D)%*%D))%*%t(W)%*%Y

get_r = function(Y,W,D, lambda=10) solve((t(W)%*%W-lambda*t(D)%*%D))%*%t(W)%*%Y

get_obj = function(Y,W,R,D) sum(Y-W%*%R)^2 + sum(lambda*(D%*%R)^2)
```

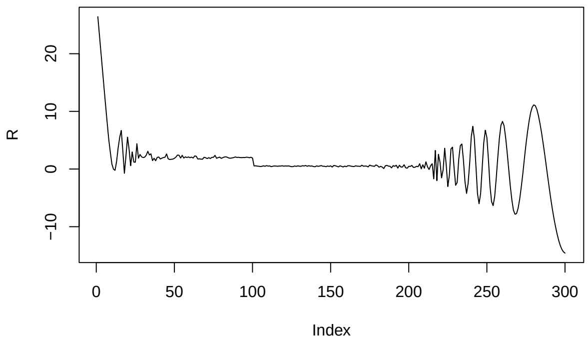
```r
### Function: Cross validation on ridge regression
CV <- function(W, Y, D, lambdas = exp(seq(0.1,10,0.2))){
  ### Length of data
  dat_length = length(Y)

  ### Get grid of lambda
  cv_scores = c()
  best = 0

  I = diag(rep(1, dat_length))

  for (lambda in lambdas){
    L = solve((t(W)%*%W-lambda*t(D)%*%D))%*%t(W)
    H = I-t(L)%*%W
    H_tilde = diag(diag(H))
    HHY = H%*%solve(H_tilde)%*%Y
    E_cv = 1/(dat_length)*sum(HHY^2)
    cv_scores = c(cv_scores, E_cv)
  }

  return(cv_scores)

}

lambdas = exp(seq(0.1,10,0.2))

cv_scores = CV(W, Y, D, lambdas = lambdas)
plot(log(cv_scores))
```

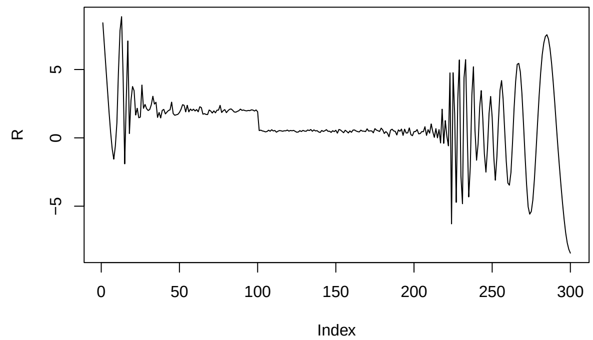


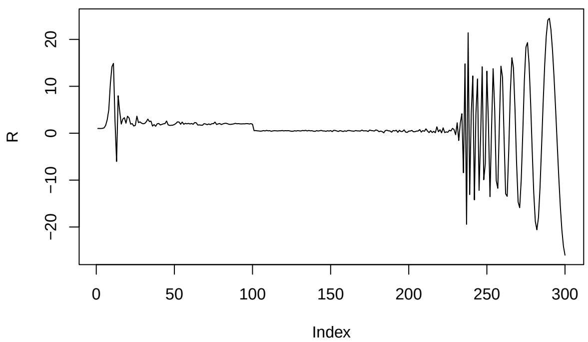

```
ordered = order(cv_scores)
lambdas = lambdas[ordered]

### Plot the best 5
for (optim_lambda in lambdas[1:5]){
  R = get_r(Y, W, D, optim_lambda)
  plot(R, type="l")
}
```











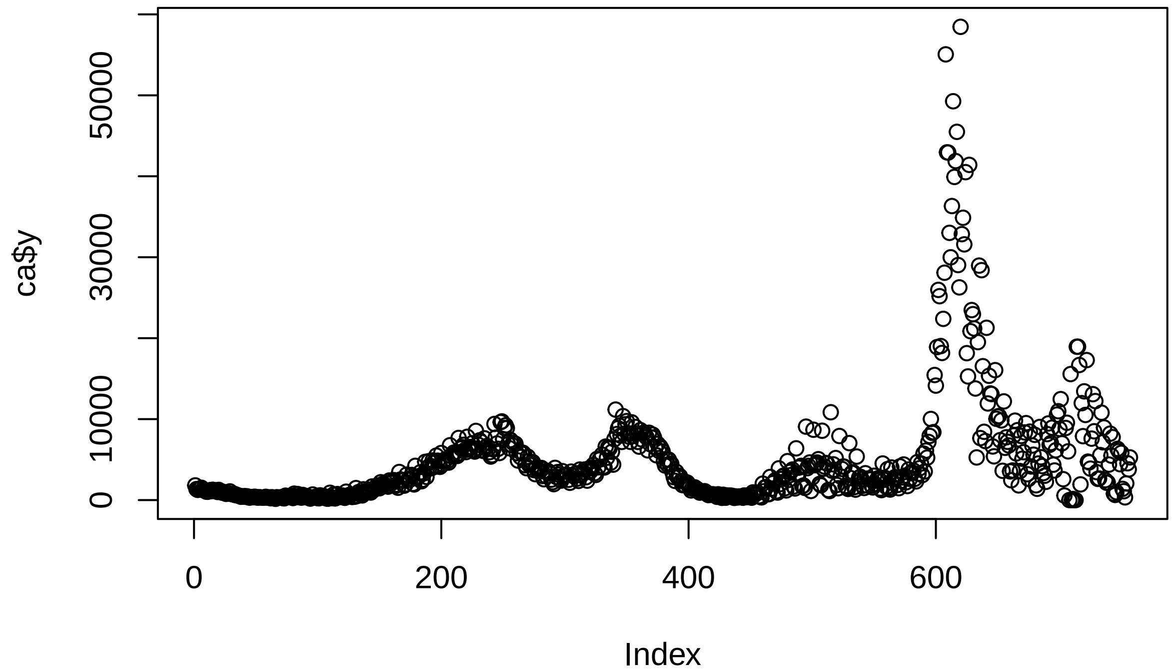

```r
source("../function/process_data.R")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::rdunif() masks extraDistr::rdunif()
```

```r
library("zoo")
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
ca <- get_owid_data(data_loc = "../data/raw/owid_Sep5.csv")
ca$y[which.max(ca$y)] = NA
ca$y <- na.locf(ca$y)
ca <- ca[100:(nrow(ca)-100),]
plot(ca$y)
```




```r
ca_iwt <- get_iwt(ca$y, disc_gamma(1:nrow(ca), sid_covid_shape, sid_covid_scale))
# ca_iwt[1] <- ca_iwt[2]
ca_iwt <- na.locf(ca_iwt)

length(ca_iwt)
```

```
## [1] 757
```

```r
W <- diag(ca_iwt)
Y <- ca$y
ca_length <- nrow(ca)

D = diag(ca_length)
D[row(D) == col(D)-1] = -1
D = D[1:(nrow(D)-1),]

print(dim(W))
```

```
## [1] 757 757
```

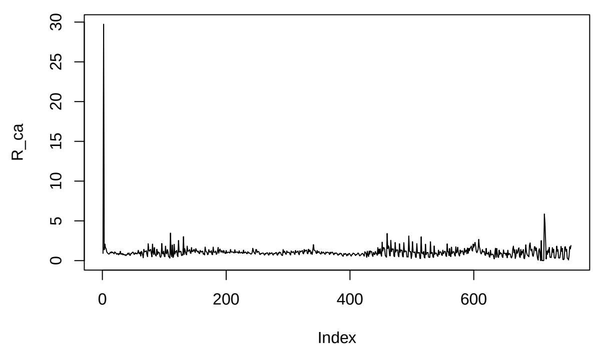
```r
print(length(Y))
```

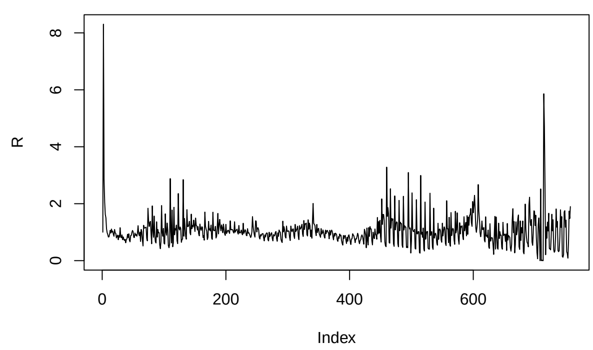
```
## [1] 757
```

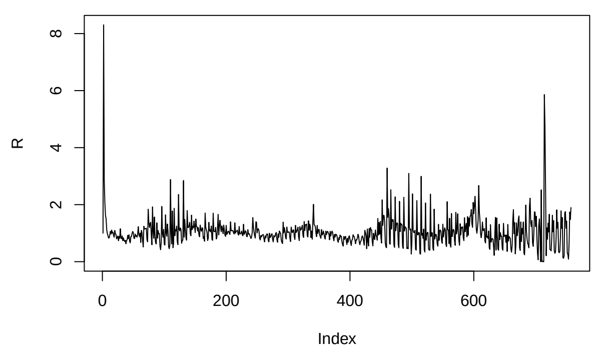
```r
print(dim(D))
```

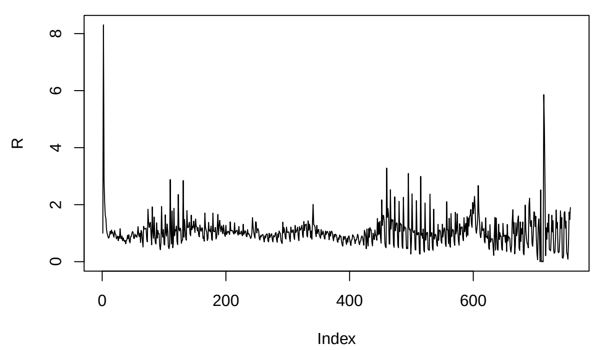
```
## [1] 756 757
```

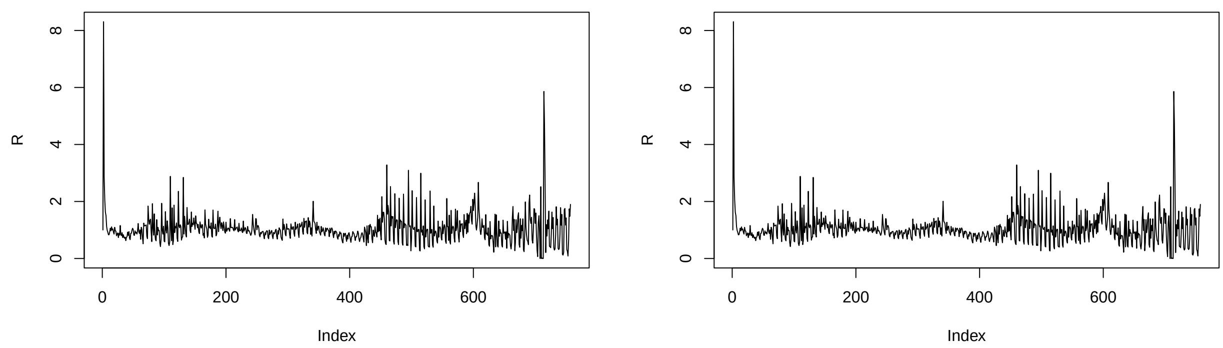
```r
R_ca <- get_r(Y, W, D, lambda=10000)
plot(R_ca, type="l")

lambdas = exp(seq(2,10,0.2))
CV_ca <- CV(W, Y, D, lambdas)


### Plot the best 5
for (optim_lambda in lambdas[1:5]){
  optim_lambda
  R = get_r(Y, W, D, optim_lambda)
  plot(R, type="l")
}
```







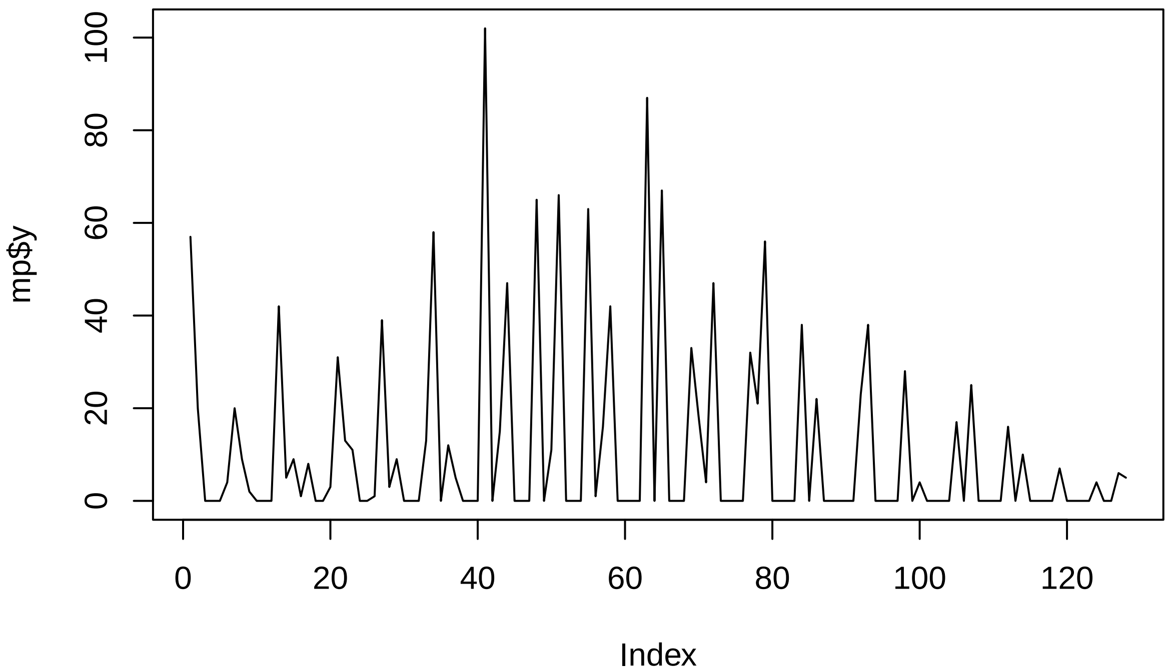





```r
source("../function/process_data.R")
source("../function/gen_data.R")

mp <- get_owid_mp(data_loc = "../data/raw/owid-monkeypox-data.csv")
head(mp)
```

```
##         date  y
## 1 2022-06-03 57
## 2 2022-06-04 20
## 3 2022-06-05  0
## 4 2022-06-06  0
## 5 2022-06-07  0
## 6 2022-06-08  4
```

```r
plot(mp$y, type="l")
```




```
mp_length <- nrow(mp)

sid_mp_mean
```

```
## [1] 9.8
```

```
sid_mp_sd
```

```
## [1] 3.875
```

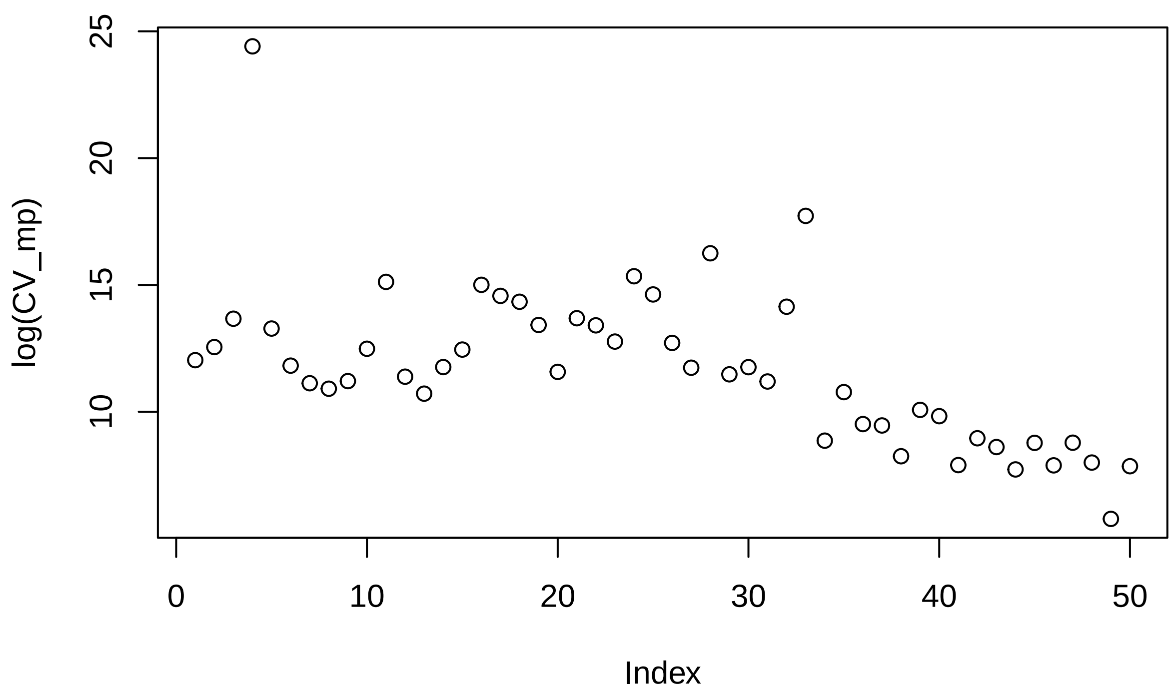
```
sid_mp_shape = gamma_reparam(sid_mp_mean, sid_mp_sd)[1]
sid_mp_scale = gamma_reparam(sid_mp_mean, sid_mp_sd)[2]

sid_mp_shape
```

```
## [1] 6.396004
```

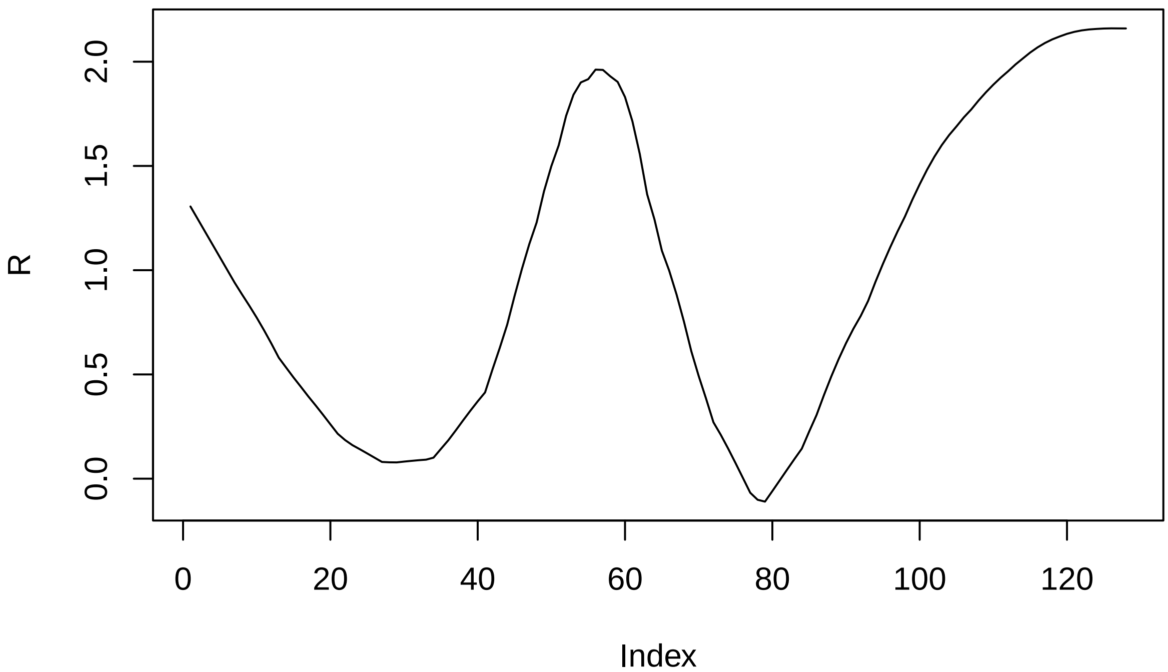
```
sid_mp_scale
```

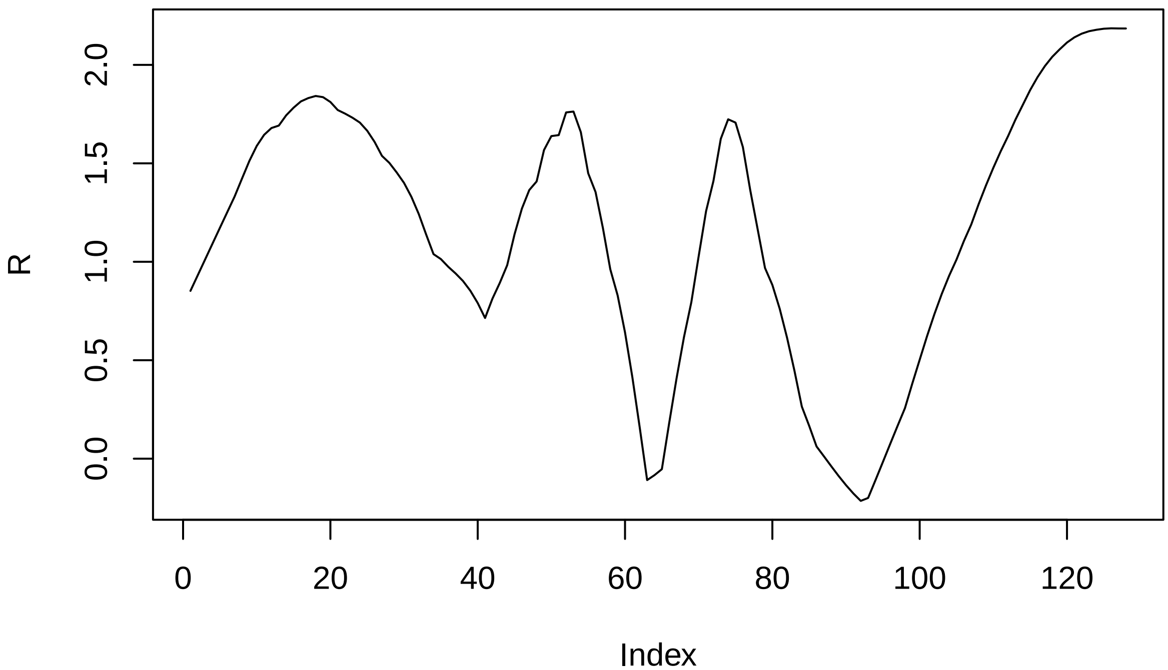
```
## [1] 1.532207
```

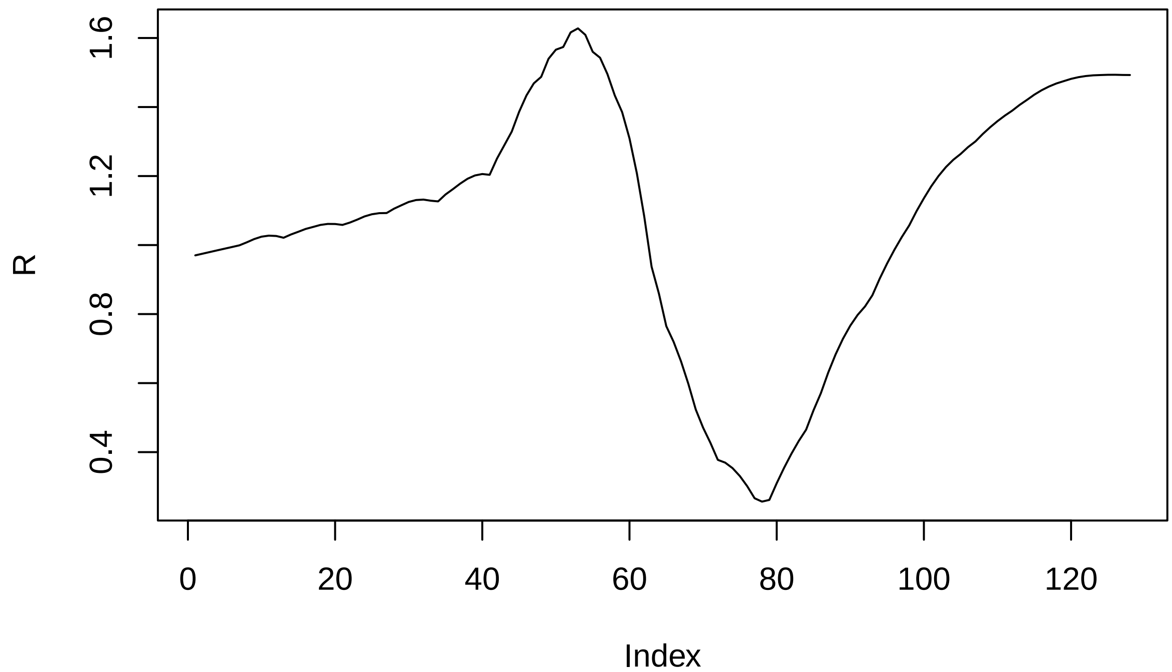
```r
mp_iwt <- get_iwt(mp$y, disc_gamma(1:nrow(mp), sid_mp_shape, sid_mp_scale))


D = diag(mp_length)
D[row(D) == col(D)-1] = -1
D = D[1:(nrow(D)-1),]

Y = mp$y

W = diag(mp_iwt)


CV_mp = CV(W, Y, D)

plot(log(CV_mp))
```

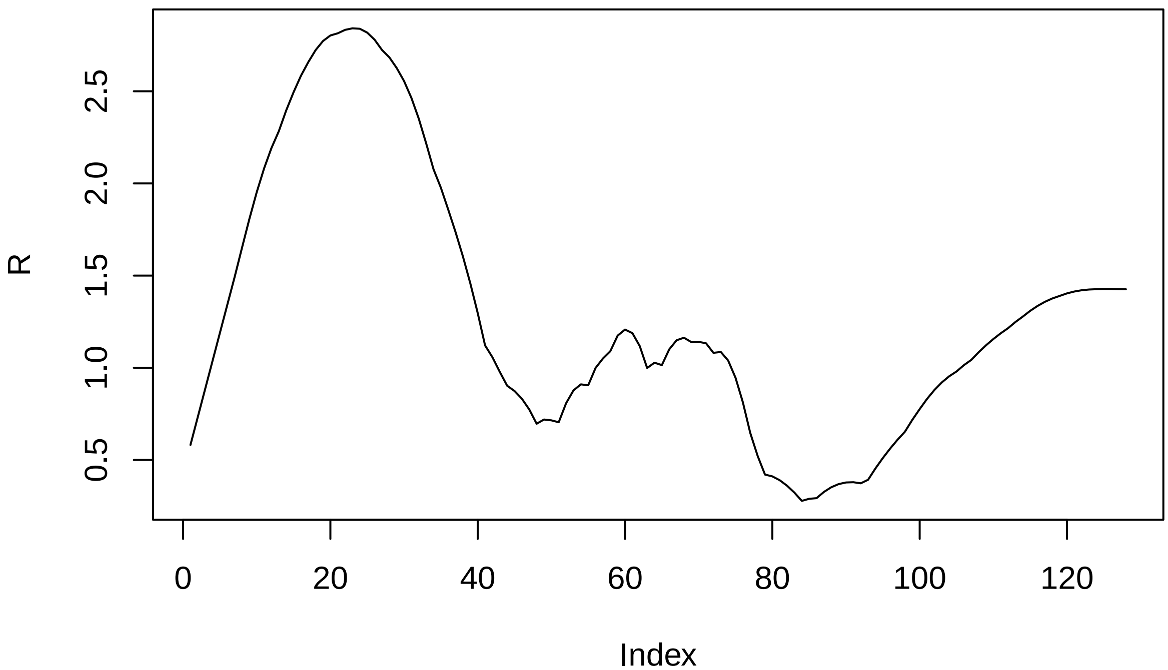



```r
lambdas = exp(seq(0.1,13,0.2))

ordered = order(CV_mp)
lambdas = lambdas[ordered]

for (optim_lambda in lambdas[1:5]){
  R = get_r(Y, W, D, optim_lambda)
  plot(R, type="l")
}
```

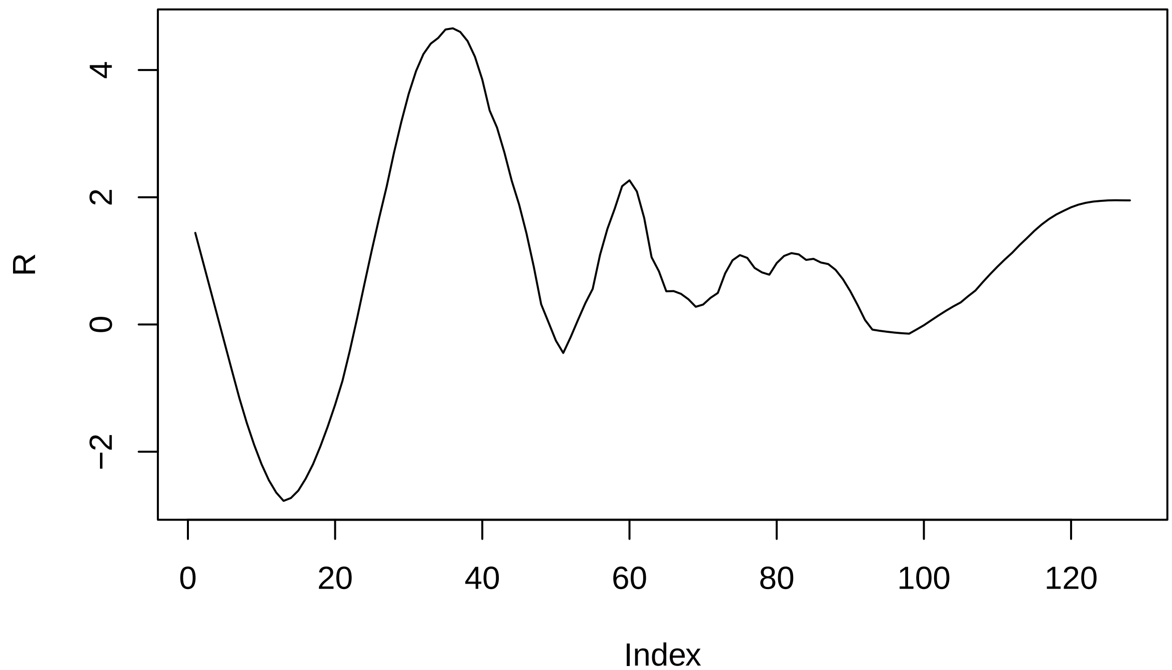

R
0.0
0.5
1.0
1.5
2.0
0
20
40
60
80
100
120
Index


R
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
120
Index


R
0.4
0.8
1.2
1.6
0
20
40
60
80
100
120
Index



R
0.5
1.0
1.5
2.0
2.5
0
20
40
60
80
100
120
Index




```r
#
# R_mp <- get_r(Y, W, D, lambda=50)
#
# plot(R_mp, type="l")
```

Smoothness at head and tail

1. Start and end of the synthetic datasets have low case counts, therefore prediction (using renewal equation) fluctuate. Model should take this into consideration and should not put too much effort in trying to smoothing those regions.
2. Penalty change with size of case count?
3. Adding cyclic penalty term?