# data simulation

*ss5929*

*7/30/2020*

## With slow onset or decay

### Exposure

For the exposure, in this scenario, I will only use $x_{t(j)}$ and $x_{t(j-1)}$ to predict $x_{t(j+1)}$ and then the data generation process(DGP) is:

$$x_{t(j+1)} = g^X(x_{t(j)}, x_{t(j-1)})$$

for each time point, $x_{t(j)}$ will follow a Bernoulli distribution and the data generation model is(DGM):

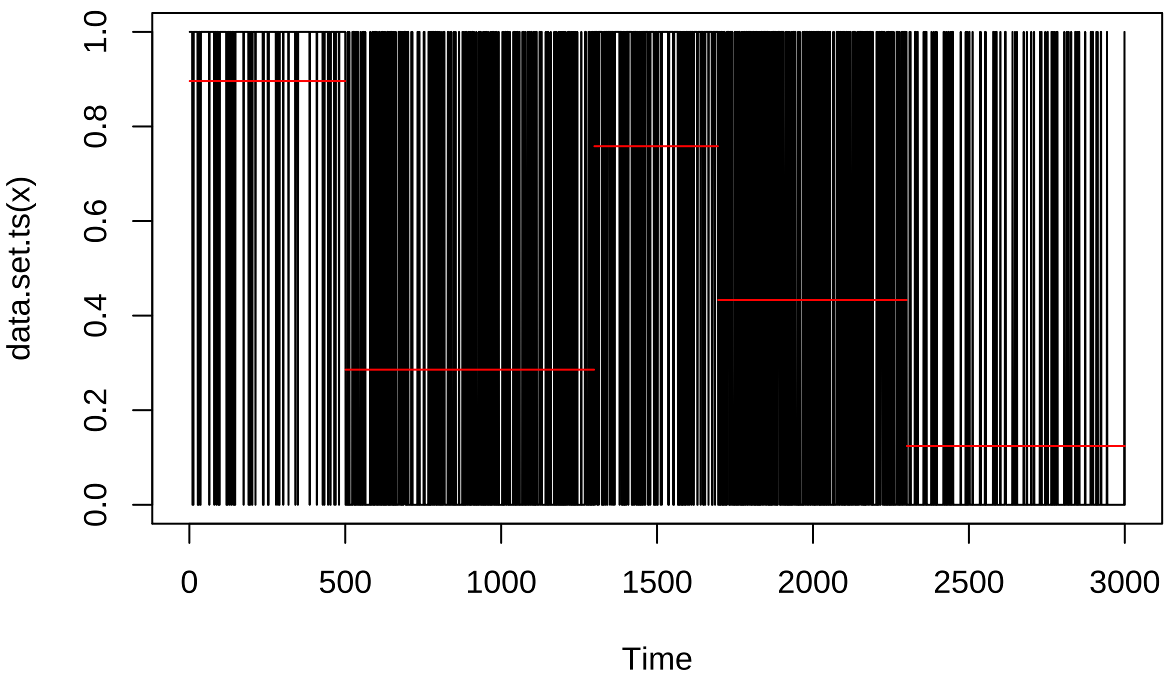
$$X_{t(j+1)} \sim bernoulli(logit(p_{t(j+1)}))$$
$$logit(p_{t(j+1)}) = \nu_t + \phi_{1,t} X_{t(j)} + \phi_{2,t} X_{t(j-1)}$$

Here, $\nu_t$ means fixed effect for x at different time periods which is decided by the probability of having high count/texts in each time period. To make it stationary and close to the fact the first lag will have more effect to the current day than the second lag, $\phi_1$ and $\phi_2$ are restricted as:

$$|\phi_1 + \phi_2| < 1$$
$$|\phi_2 - \phi_1| < 1$$
$$|\phi_1| < 1$$
$$|\phi_2| < 1$$
$$|\phi_2| < |\phi_1|$$

```
#x = c(rbinom(500,1,6/7),rbinom(800,1,2/7),rbinom(400,1,5/7),rbinom(600,1,3/7),rbinom(700,1,1/7))
seed = 1234
# simulate x
x = NULL
phi1 = c(1/7,1/20,1/8,1/10,1/30) # set up phi1
phi2 = c(1/8,1/30,1/9,1/12,1/40) # set up phi2
gama = c(1.8,-0.9,0.9,-0.3,-1.8) # define gama(fixed value)
num = c(500,800,400,600,700)  # define the number of time points in every period
x_test_result = NULL


for(j in 1:5)
{
  set.seed(3399)
  p.treat = NULL
  p = NULL
  a = NULL
  a[1] = ifelse(j==1,1,ifelse(j==2,0,ifelse(j==3,0,ifelse(j==4,1,ifelse(j==5,0,NULL)))))
  a[2] = ifelse(j==1,1,ifelse(j==2,0,ifelse(j==3,1,ifelse(j==4,0,ifelse(j==5,0,NULL))))) # set up the f
  for(i in 3:num[j])
  {
```

```
    p[i] = gama[j]+phi1[j]*a[i-1]+phi2[j]*a[i-2]
    p.treat= exp(p[i])/(1+exp(p[i]))
    a[i] =sample(rbinom(1000,1,p.treat),size=1)

  }
  adf = adf.test(a)
  x_test_result[j] = adf$p.value
  x=c(x,a)

}

plot(cpt.mean(x,penalty='Manual',pen.value = 2,method='PELT'))
```




The parameters for $\nu$, probability and number of time points at different time period are as below:

| | | Parameters of Exposure | |
|---|---|---|---|
| time.period | num | gama | probability |
| 1 | 500 | 1.8 | 0.8571429 |
| 2 | 800 | -0.9 | 0.2857143 |
| 3 | 400 | 0.9 | 0.7142857 |
| 4 | 600 | -0.3 | 0.4285714 |
| 5 | 700 | -1.8 | 0.1428571 |

## Outcome

The DGP is:

$$Y_{t(j)} = g^Y(X_{t(j)}, Y_{t(j-1)}, U_{t(j)})$$

The DGM is:

$$Y_{t(j)} = \zeta_t + \alpha_t X_{t(j)} + \phi_t Y_{t(j-1)} + \eta_{t(j)} U_{t(j)} + \varepsilon_{t(j)}$$

$\zeta_t$: fixed effect for Y at different time period

$U_{t(j)}$: other nonexposure covariates

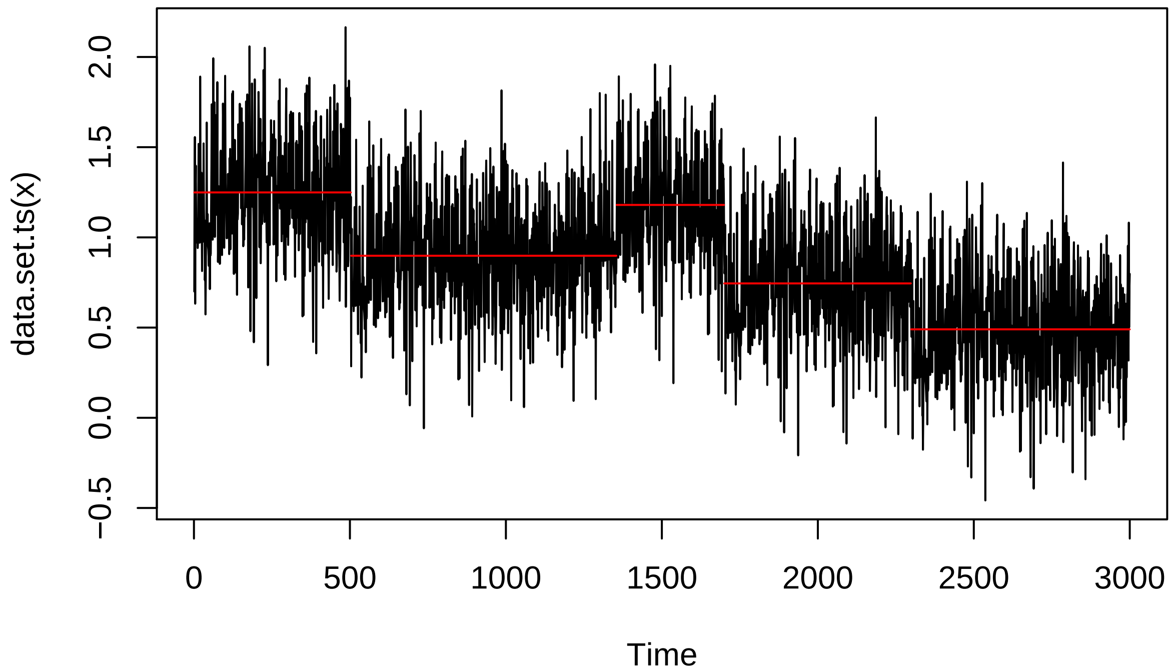
$\varepsilon_{t(j+1)}$ $N(0, \sigma_t^2)$: error term

Other covariates and $\eta_{t(j)}$ will be updated evey time when a new observation is added:

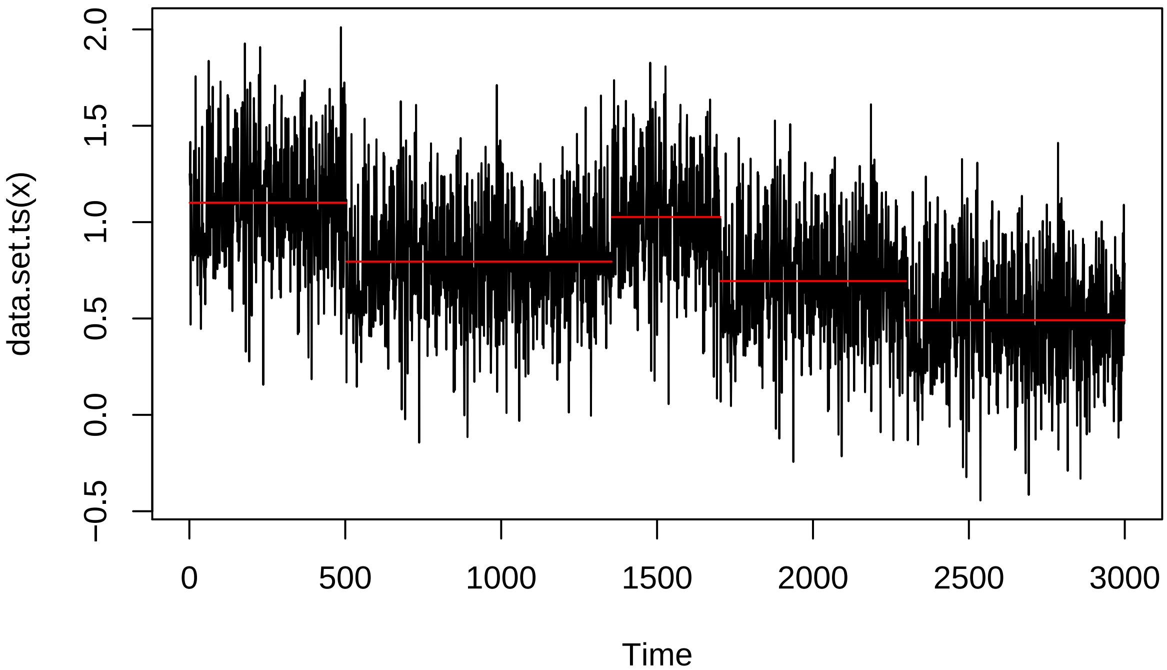
$U_{t(j+1)} = U_t^{\triangle} + \rho_{1,t}(U_{t(j)} - U_t^{\triangle}) + w_{1,t(j)}$ $w_{1,t(j)} \sim N(0, \sigma_{1,w}^2)$

$\eta_{t(j+1)} = \eta_t^{\triangle} + \rho_{2,t}(\eta_{t(j)} - \eta_t^{\triangle}) + w_{2,t(j)}$ $w_{1,t(j)} \sim N(0, \sigma_{2,w}^2)$

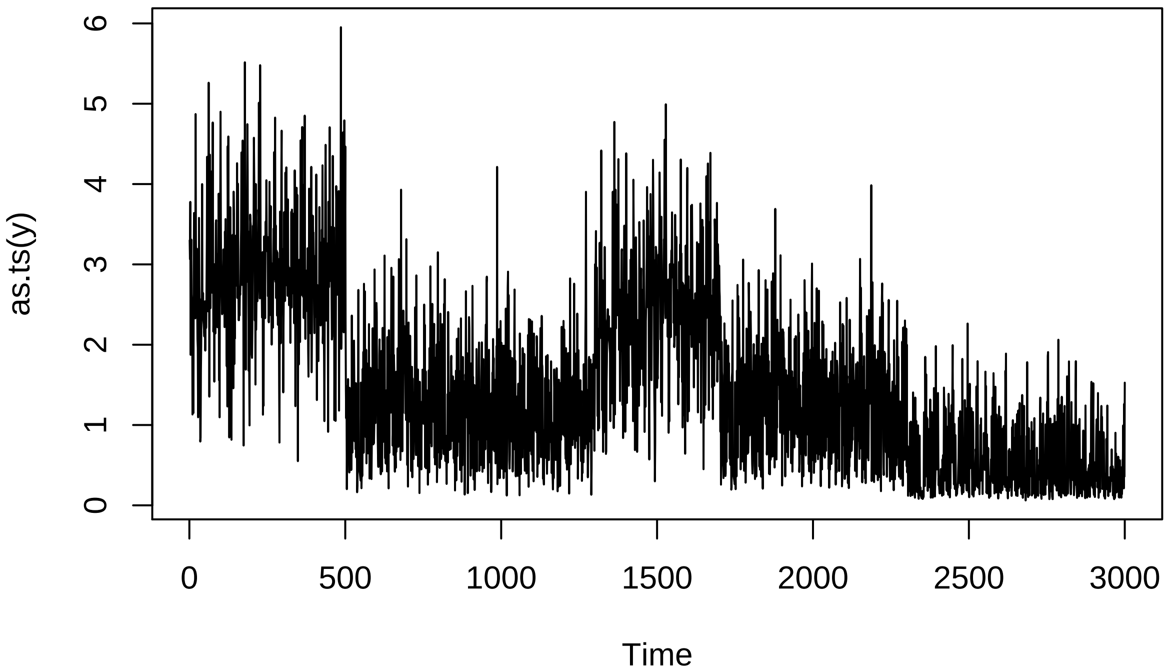
```r
# simulate nonexposure covariates U
u = NULL
var_u = 0.08
u0 = c(2.5,1.8,2.3,1.5,1)/2
rho1 = c(0.3,0.2,0.3,0.2,0.1)/2


first_c = c(0.7,1.2,1.8,0.8,0.5)
for(j in 1:5)
{
  set.seed(seed)
  c=NULL
  c[1]=first_c[j]
  error_u = rnorm(num[j],0,sqrt(var_u))
  for(i in 2:num[j])
  {
    c[i] = u0[j] + rho1*(c[i-1]-u0[j])+error_u[i]
  }
  u = c(u,c)
  u_error = NULL
}


plot(cpt.mean(u,penalty='Manual',pen.value = 2,method='PELT'))
```



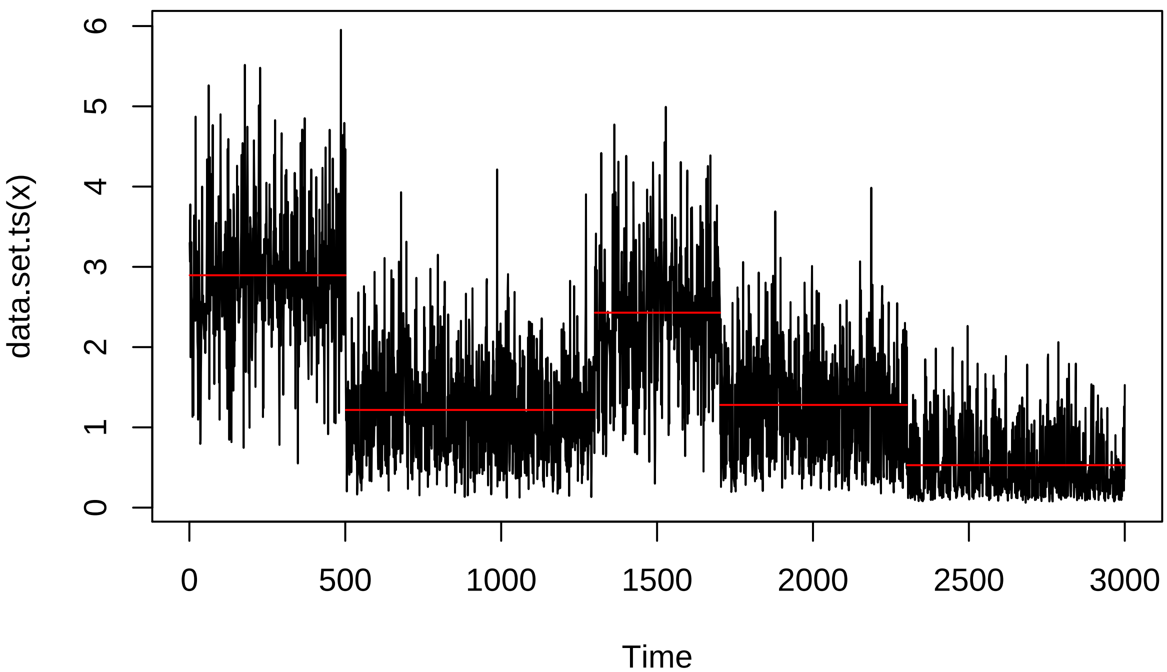

```r
# simulate eta
eta = NULL
var_eta = 0.08
eta0 = c(2.2,1.6,2,1.4,1)/2
rho2 = c(0.2,0.1,0.2,0.14,0.08)/2

first_eta = c(1.25,1.05,1.15,0.95,0.55)
for(j in 1:5)
{
  set.seed(seed)
  d=NULL
  d[1]= first_eta[j]
  error_eta = rnorm(num[j],0,sqrt(var_eta))
  for(i in 2:num[j])
  {
    d[i] = eta0[j] + rho2*(d[i-1]-eta0[j])+error_eta[i]
  }
  eta = c(eta,d)
  error_eta = NULL
}

plot(cpt.mean(eta,penalty='Manual',pen.value = 2,method='PELT'))
```




```
#simulate Y
zeta = c(2,1,2,1.5,0.8)/10 #fixed effect for y at different time period
phi = c(0.8,0.5,0.7,0.6,0.4)/15
alpha = c(2.4,2,2.3,2.2,1.8)/2
var_y = 0.0001

y_test_result=NULL
y=NULL
id=0
for(j in 1:5)
{
  set.seed(seed)
  e =NULL
  e[1] = ifelse(j==1,3.3,ifelse(j==2,2.3,ifelse(j==3,3,ifelse(j==4,2.6,ifelse(j==5,2,NULL)))))
  error_y = rnorm(num[j],0,sqrt(var_y))
  for(i in 2:num[j])
  {
    e[i] = zeta[j]+alpha[j]*x[id:(id+num[j])][i]+phi[j]*e[i-1]+
      eta[id:(id+num[j])][i]*u[id:(id+num[j])][i]+error_y[i]
  }
  adf = adf.test(e)
  y_test_result[j] = adf$p.value
  y = c(y,e)
  error_y =NULL
  id=id+num[j]

}
plot(as.ts(y))
```




```
plot(cpt.mean(y,penalty='Manual',pen.value = 12,method='PELT'))
```




Parameters for the outcome are shown as below:

| | | other covariates | | | eta | | | final outcome | | |
|---|---|---|---|---|---|---|---|---|---|---|
| time.period | num | var_u | u0 | rho1 | var_eta | eta0 | rho2 | alpha | zeta | phi |
| 1 | 500 | 0.08 | 1.25 | 0.15 | 0.08 | 1.1 | 0.10 | 1.20 | 0.20 | 0.05 |
| 2 | 800 | 0.08 | 0.90 | 0.10 | 0.08 | 0.8 | 0.05 | 1.00 | 0.10 | 0.03 |
| 3 | 400 | 0.08 | 1.15 | 0.15 | 0.08 | 1.0 | 0.10 | 1.15 | 0.20 | 0.05 |
| 4 | 600 | 0.08 | 0.75 | 0.10 | 0.08 | 0.7 | 0.07 | 1.10 | 0.15 | 0.04 |
| 5 | 700 | 0.08 | 0.50 | 0.05 | 0.08 | 0.5 | 0.04 | 0.90 | 0.08 | 0.03 |