

# Changepoint analysis of transients

Jody Reimer

2019-10-06

```
library(tidyverse)
library(nimble)
library(ecp)
```

```
## Warning: package 'ecp' was built under R version 3.6.1
```

```
#a = 0.023
r = 0.05
p <- list(r = .05, K = 2, Q = 5, H = .38, sigma = .02, a=0.023, N = 1e3)
growth <- function(x, p) x * p$r * (1 - x / p$K)
consumption <- function(x,p) p$a * x ^ p$Q / (x^p$Q + p$H^p$Q)
```

```
theory <-
  tibble(x= seq(0,2, length.out = 100)) %>%
  mutate(g = growth(x,p),
         c = consumption(x,p)) %>%
  mutate(potential = - cumsum(g - c)) %>%
  gather(curve, y, -x, -potential)
```

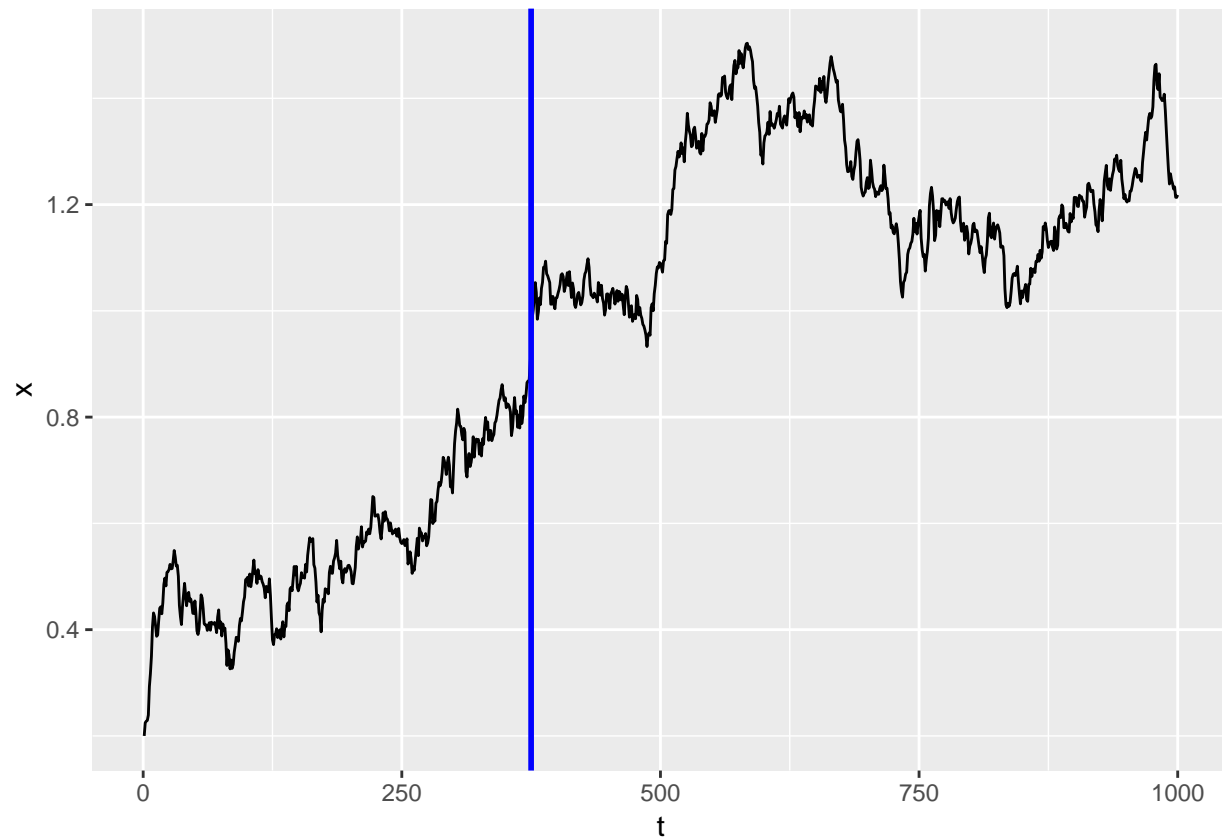
```
# Define stochastic model in BUGS notation
```

```
may <- nimble::nimbleCode({
```

```
  x[1] <- x0
  for(t in 1:(N-1)){
    # Deterministic mean looks like standard R
    mu[t] <- x[t] + x[t] * r * (1 - x[t] / K) - a * x[t] ^ Q / (x[t] ^ Q + H ^ Q)
    # Note the use of ~ in BUGS to show 'distributed as normal'
    y[t+1] ~ dnorm(mu[t], sd = sigma)
    x[t+1] <- max(y[t+1],0)
  }
}
```

```
})
model <- nimbleModel(may,constants = p, inits = list(x0 = 0.2))
cmodel <- model #compileNimble(model)
set.seed(123456)
simulate(cmodel)
df <- tibble(t = seq_along(cmodel$x), x = cmodel$x)
df %>% write_csv("data.single.csv")
```

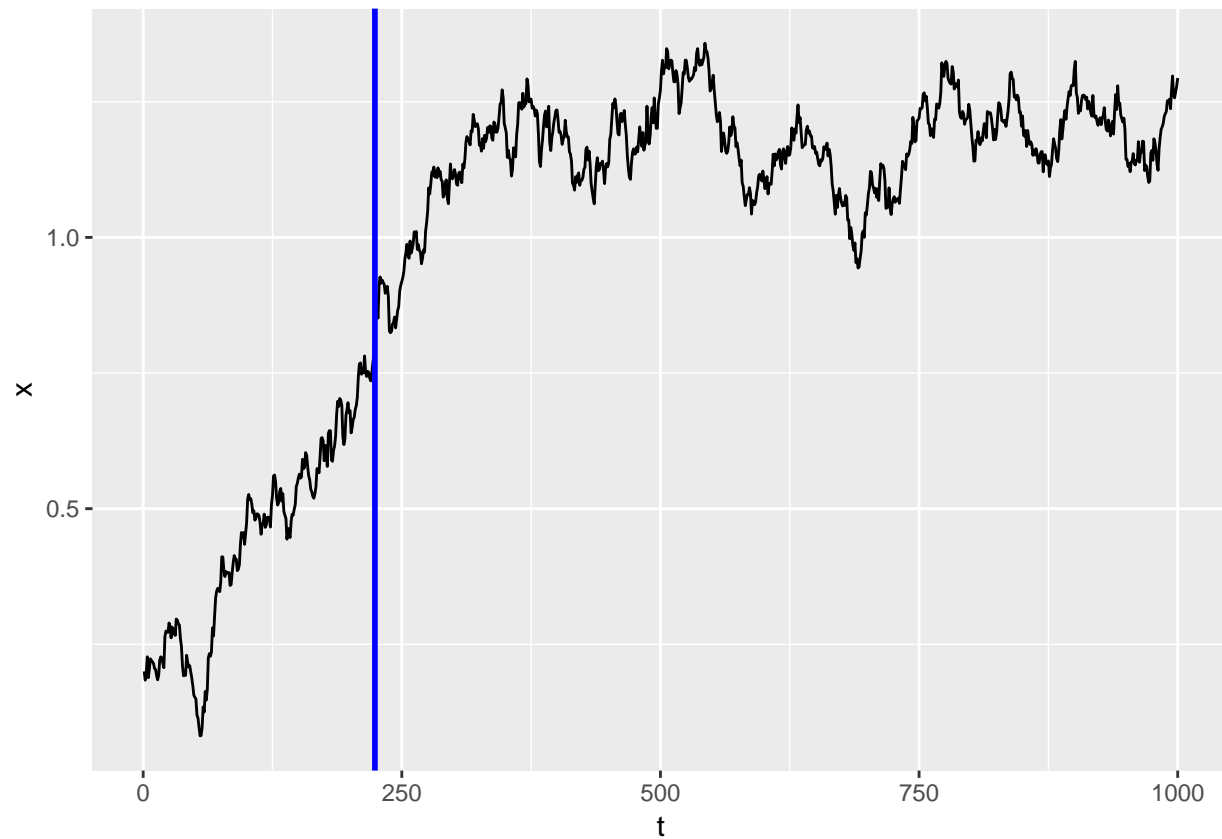
```
ECP.res <- e.divisive(df[,2],k=1)
df %>% ggplot(aes(t, x)) + geom_line() + geom_vline(xintercept=ECP.res$estimates[2], color="blue", size=
```



Changepoint is estimated as 375

Now let's change the seed...

```
set.seed(1234)
simulate(cmodel)
df <- tibble(t = seq_along(cmodel$x), x = cmodel$x)
ECP.res <- e.divisive(df[,2],k=1)
df %>% ggplot(aes(t, x)) + geom_line() + geom_vline(xintercept=ECP.res$estimates[2], color="blue", size=2)
```



The new changepoint is estimated as 224.

Now let's do 100 replicates

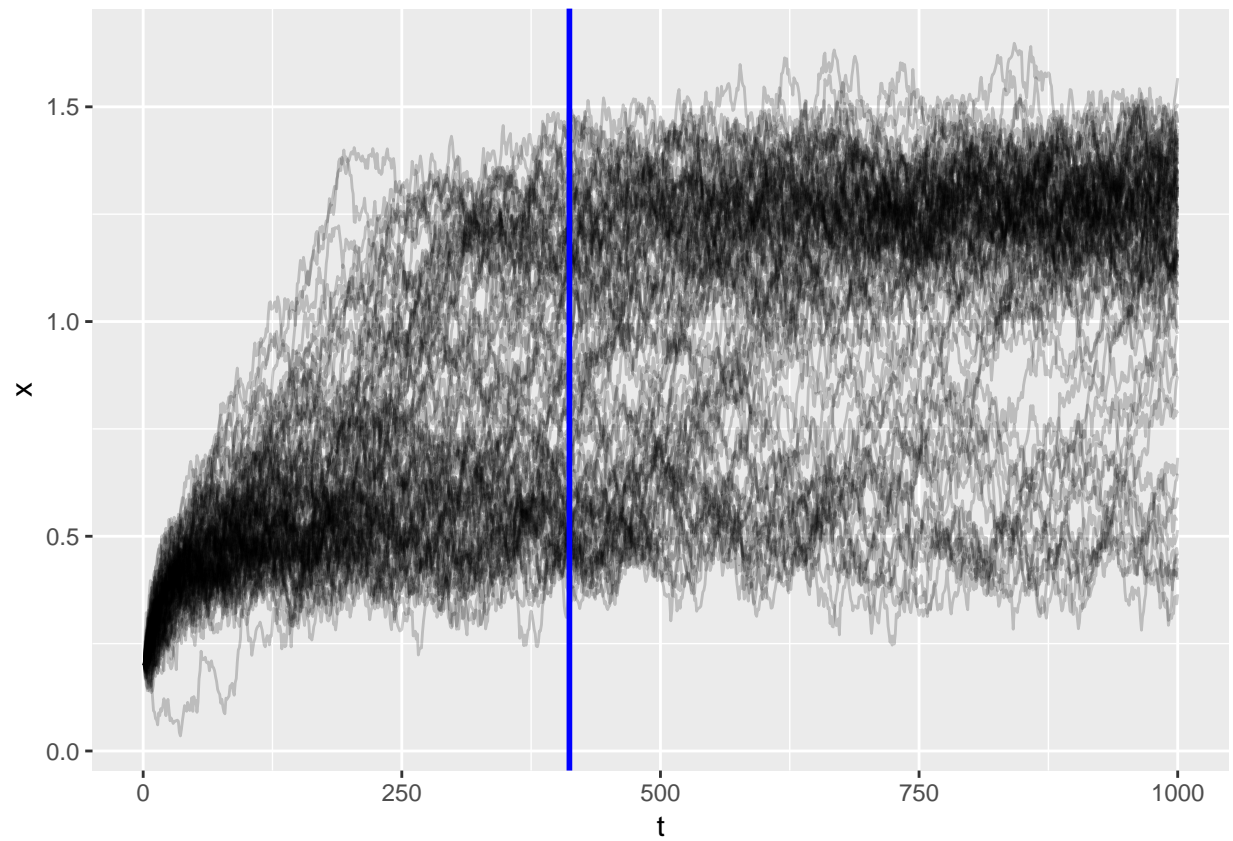
```
set.seed(123)
```

```
df <- map_dfr(1:100,
  function(i){
    simulate(cmodel)
    tibble(t = seq_along(cmodel$x), x = cmodel$x, reps = i)
  })
df %>% write_csv("data.reps.csv")
```

```
df.reps <- df %>% spread(reps, x)
```

```
ECP.res.reps = e.divisive(df.reps[, -1], k=1)
```

```
df %>% ggplot(aes(t, x, group=reps)) + geom_line(alpha=.2) + geom_vline(xintercept=ECP.res.reps$estimate)
```



```
#matplot(rep.dat, type = "l", col = scales::alpha(1, 0.01), lty = 1)
```

Changepoint is estimated as 412