Data generation

Jody Reimer

2019-10-10

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

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

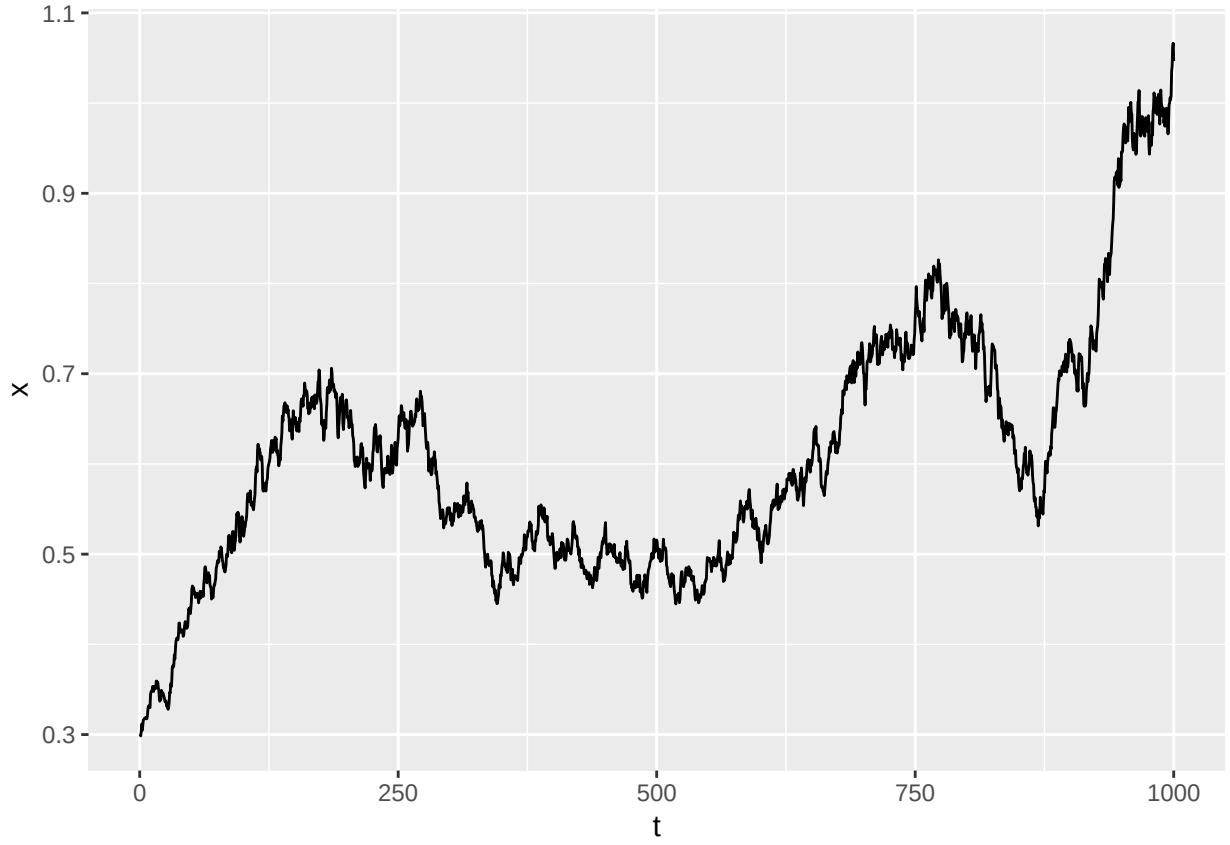
```
p <- list(r = .05, K = 2, Q = 5, H = .38, sigma = 0.02, a=0.023, N = 1e3, t.step = 1/2)
# note: you may need to make N larger for small values of sigma (e.g., N = 3e3 is good for sigma=0)
growth <- function(x, p) x * p$r * (1 - x / p$K)
consumption <- function(x,p) p$a * x ^ p$Q / (x^p$Q + p$H^p$Q)
```

```
# Define stochastic model in BUGS notation
may  <- nimble::nimbleCode({

  x[1] <- x0
  for(t in 1:((1/t.step)*N-1)){
    # Determinstic mean looks like standard R
    mu[t] <- x[t] + t.step*(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*mu[t]*sqrt(t.step)) # or should this be lognormal?
    # note: since variance scales linearly with time, sd scales with the square root of time
    x[t+1] <- max(y[t+1],0)
  }
})
model <- nimbleModel(may,constants = p, inits = list(x0 = 0.3))
cmodel <- model #compileNimble(model)
```

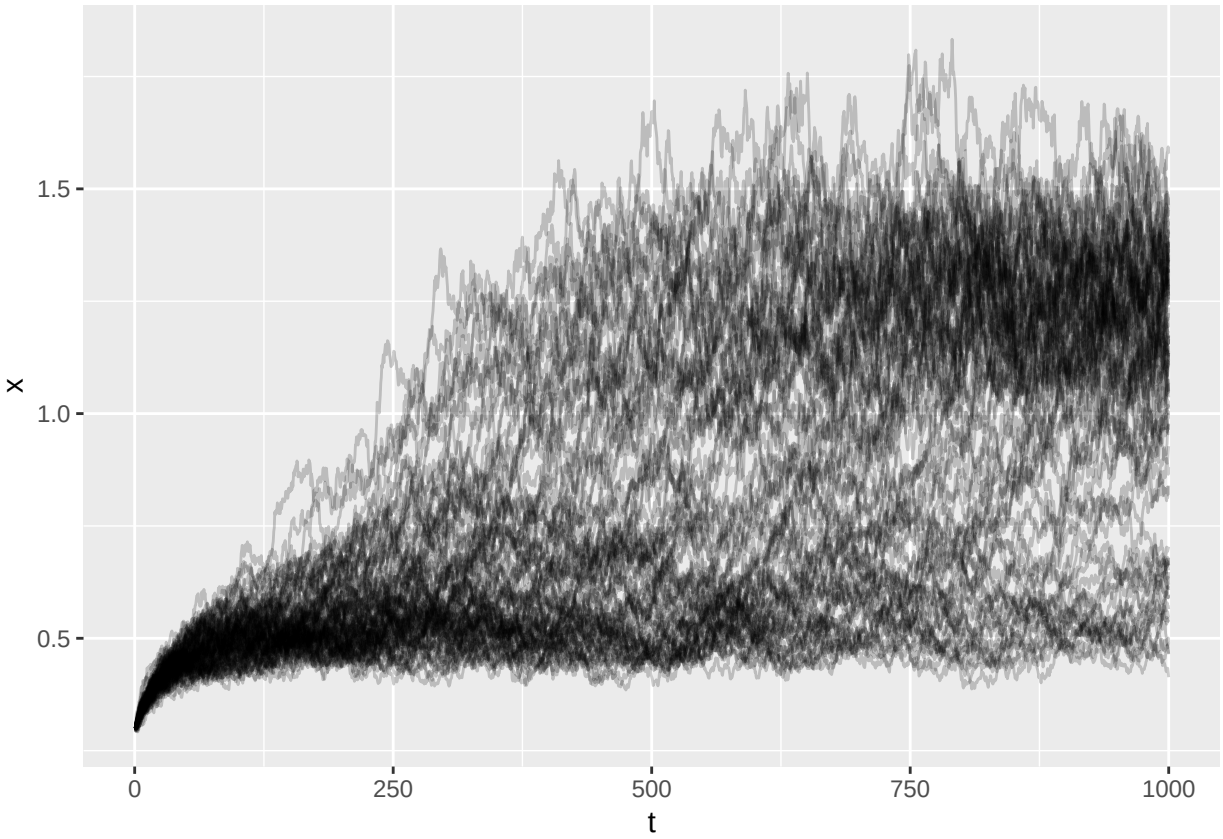
Let's simulate 1 replicate

```
set.seed(1234)
simulate(cmodel)
df <- tibble(t = p$t.step*seq_along(cmodel$x), x = cmodel$x)
df %>% write_csv(paste0("data.single.tstep.",p$t.step,".sigma.",p$sigma,".N.",p$N,".csv"))
df %>% ggplot(aes(t, x)) + geom_line()
```



Now let's simulate 100 replicates

```
df <- map_dfr(1:100,
  function(i){
    simulate(cmodel)
    tibble(t = p$t.step*seq_along(cmodel$x), x = cmodel$x, reps = i)
   })
df %>% write_csv(paste0("data.reps.tstep.",p$t.step,".sigma.",p$sigma,".N.",p$N,".csv"))
df %>% ggplot(aes(t, x, group=reps)) + geom_line(alpha=.2)
```


1.5
1.0
0.5
x
0
250
500
750
1000
t