HW2: Dynamic ODE

AUTHOR
Brooke Grazda

PUBLISHED
May 12, 2025

Consider the following model of forest growth (where forest size in measured ni units of carbon (C))

• dC/dt = r * C′ for forests where C is below a threshold canopy closure • dC/dt = g* 1(- C/K) for forests where carbon is at or above the threshold canopy closure • K is a carrying capacity in units of carbon

The size of the forest (C), Canopy closure threshold and carrying capacity are al in = units of carbon You could think of the canopy closure threshold as the size of the forest at which growth rates change from exponential to linear You can think of r, as early exponential growth rate and gas the linear growth rate once canopy closure has been reached

```
library(tidyverse)
library(deSolve)
library(sensitivity)
library(here)

source(here("hw5/dforestgrowth.R"))
source(here("hw5/maxforest.R"))
source(here("hw5/maxforestwrapper.R"))
```

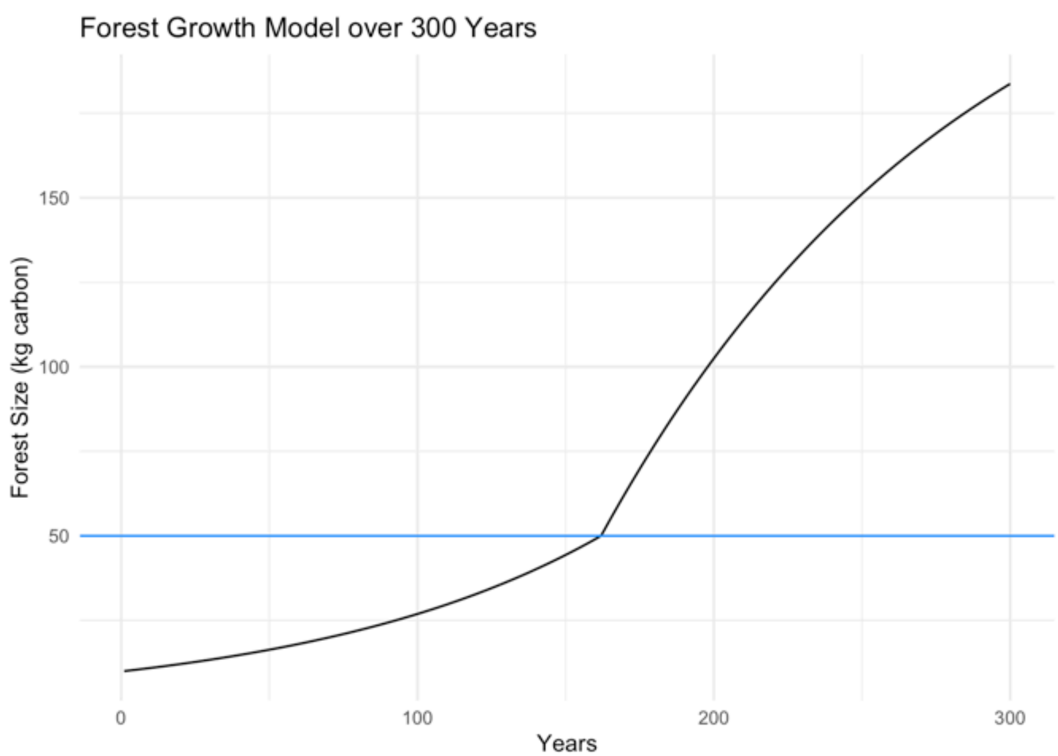
Forest Growth Model

```
# initial population list
Cinitial <- c(C=10)

# time period over 300 yr sequence
simtimes <- seq(from = 1, to = 300)

# list of parameters
parms <- list(r = 0.01,
              g = 2,
              K = 250,
              thresh = 50)

# run ODE model
```

```
result <- ode(y = Cinitial, times = simtimes, func = df
```

Visualize the Forest Growth Model

```
# Graph forest growth over time
ggplot(result, aes(time, C)) +
  geom_line() +
  geom_hline(yintercept = 50, col = "dodgerblue ") +
  labs(y = "Forest Size (kg carbon)",
       title = "Forest Growth Model over 300 Years",
       x = "Years") +
  theme_minimal()
```



Sobol Global (Vary all parameters)

Sensitivity Analysis

```
# Set initial population
Cinitial <- 10
```

```r
# Create parameter distributions
np <- 2000
K <- rnorm(mean = 250, sd = 25, n = np)
r <- rnorm(mean = 0.01, sd = 0.001, n = np)
g <- rnorm(mean = 0.2, sd = 0.02, n = np)
X1 <- cbind.data.frame(r = r, K = K, g = g)
# We don't vary thresh so add it here statically
X1 <- X1 |>
  mutate(thresh = 50)

# Repeat for second set of samples
np <- 2000
K <- rnorm(mean = 250, sd = 25, n = np) # carrying capa
r <- rnorm(mean = 0.01, sd = 0.001, n = np)
g <- rnorm(mean = 0.2, sd = 0.02, n = np) # vary post
X2 <- cbind.data.frame(r = r, K = K, g = g)
# We don't vary thresh so add it here statically
X2 <- X2 |>
  mutate(thresh = 50)  |>
  map_df(pmax, 0.0) # fix negatives

# Fix any negative values and the=y are not meaningful
X1 <- X1 %>% map_df(pmax, 0.0)
X2 <- X2 %>% map_df(pmax, 0.0)

# Run the sobol sensitivity analysis
sens_C <- sobolSalt(model = NULL, X1, X2, nboot = 300)

# Add parameter names
colnames(sens_C$X) <- c("r", "K", "g", "thresh")
head(sens_C$X)
```

```
               r        K         g thresh
[1,] 0.010161614 261.3606 0.1519468     50
[2,] 0.009199783 200.9397 0.2060443     50
[3,] 0.008804727 231.3919 0.2319310     50
[4,] 0.011294539 225.6362 0.2142589     50
[5,] 0.007911615 241.0317 0.2382082     50
[6,] 0.008237328 243.9668 0.1944732     50
```

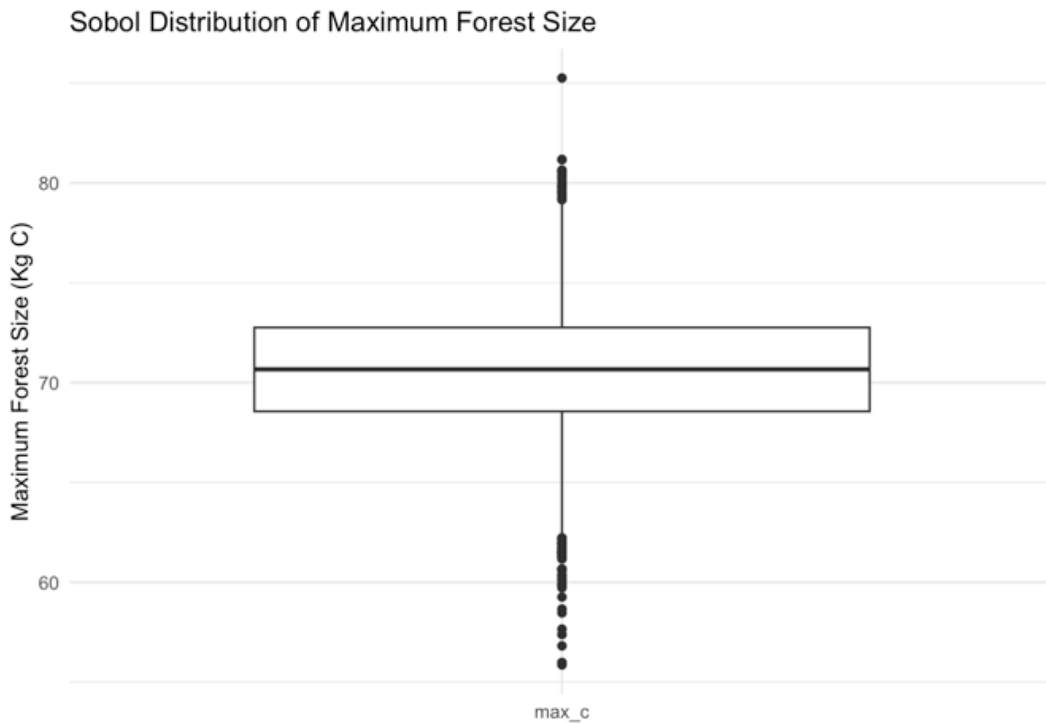
Use Wrapper Function

The metric we're looking for here is max forest growth. We'll use the functionized version of the `max` and a wrapper function to calculate this value for our model.

```
# List parameters and format for pmap
parms <- list(r = sens_C$X[, "r"],
              g = sens_C$X[, "g"],
              K = sens_C$X[, "K"],
              thresh = sens_C$X[, "thresh"])

# Map parameters onto ODE forest growth function
allresults <- parms %>% pmap(maxforestwrapper, Cinitial

# Extract results and store in a data frame
allres <- allresults %>%
  map_dfr(~ tibble(max_c = .x[[1]]))
```

Plotting Sobol Results

```
# Retrieve maximum forest size from the sobol output
max_forest <- sens_C$y

# Pivot longer for plotting
allres_long <- allres %>%
  pivot_longer(cols = everything(),
               names_to = "max_forest",
               values_to = "value")

# Visualize results
ggplot(allres_long) +
  geom_boxplot(aes(x = max_forest, y = value)) +
  labs(title = "Sobol Distribution of Maximum Forest Si
       y = "Maximum Forest Size (Kg C)",
       x = "") +
  theme_minimal()
```



Sobol Indices

```
# Get indices for max forest
sens_Cmax <- sensitivity::tell(sens_C, allres$max_c)

# First-order indices
rownames(sens_Cmax$S) <- c("r", "g", "K", "thresh")
print(sens_Cmax$S)
```

```
          original          bias std. error   min. c.i.   max.
c.i.
r       0.57985602  0.0014139825 0.01623792  0.54864074
0.60865686
g       0.03298538  0.0006258659 0.02344214 -0.01482866
0.08347101
K       0.34476616 -0.0012688411 0.02111638  0.30741066
0.38597247
thresh -0.01921714  0.0004440735 0.02350707 -0.07121068
0.03084294
```

```
# Total sensitivity index
rownames(sens_Cmax$T) <- c("r", "g", "K", "thresh")
```

```
print(sens_Cmax$T)
```

```
           original          bias   std. error     min. c.i.
max. c.i.
r      5.905620e-01  1.523578e-03 1.982989e-02  5.521163e-01
6.297426e-01
g      4.737845e-02  2.468061e-04 2.268461e-03  4.327350e-02
5.158944e-02
K      3.761603e-01 -1.000161e-03 1.445563e-02  3.442972e-01
4.043634e-01
thresh 3.368417e-13 -2.495966e-13 9.894241e-13 -1.293500e-12
2.635858e-12
```