

EDS 230/ESM 232 Growth Model Assignment

Genevieve Chiong, Halina Do-Linh, Mia Forsline

2022-05-19

Set up

Introduction

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

- $dC/dt = r \times C$ for forests
 - where C is below a threshold canopy closure
 - and r is the early exponential growth rate
- $dC/dt = g \times (1-C/K)$ for forests
 - where C is at or above the threshold canopy closure
 - and g is the linear growth rate
- and K is a carrying capacity in units of carbon.

The size of the forest (C), canopy closure threshold, and carrying capacity (K) are all 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 g as the linear growth rate once canopy closure has been reached

1. Implement this model in R (as a differential equation)

```
source(here("R", "forest_growth.R"))
source(here("R", "compute_metrics.R"))
```

```
forest_growth
```

```
## function (time, C, parms)
## {
##   if (C < parms$threshold) {
##     dC_dt = parms$r * C
```

```
##     }
##     else {
##         dC_dt = parms$g * (1 - (C/parms$K))
##     }
##     return(list(dC_dt))
## }
```

```
compute_metrics
```

```
## function (result)
## {
##     maxpop = max(result$P)
##     idx = which.max(result$P)
##     meanpop = mean(result$P)
##     return(list(maxpop = maxpop, meanpop = meanpop))
## }
```

2. Run the model for 300 years (using the ODE solver) starting with an initial forest size of 10 kg/C, and using the following parameters:

- canopy closure threshold of 50 kgC
- $K = 250$ kg C (carrying capacity)
- $r = 0.01$ (exponential growth rate before canopy closure)
- $g = 2$ kg/year (linear growth rate after canopy closure)

```
#set up parameters
C = 10 #forest size
time = seq(from = 1, to = 300) #run the model for 300 years
threshold = 50 #canopy closure
K = 250
r = 0.01 #exponential growth rate
g = 2 #linear growth rate

#create parameters list
parms = list(time = time,
            threshold = threshold,
            K = K,
            r = r,
            g = g)

#apply solver - we input the differential equation and the ODE solver integrates it
results = ode(C, time, forest_growth, parms)

head(results)
```

```
##      time      1
## [1,]    1 10.00000
## [2,]    2 10.10050
## [3,]    3 10.20202
```

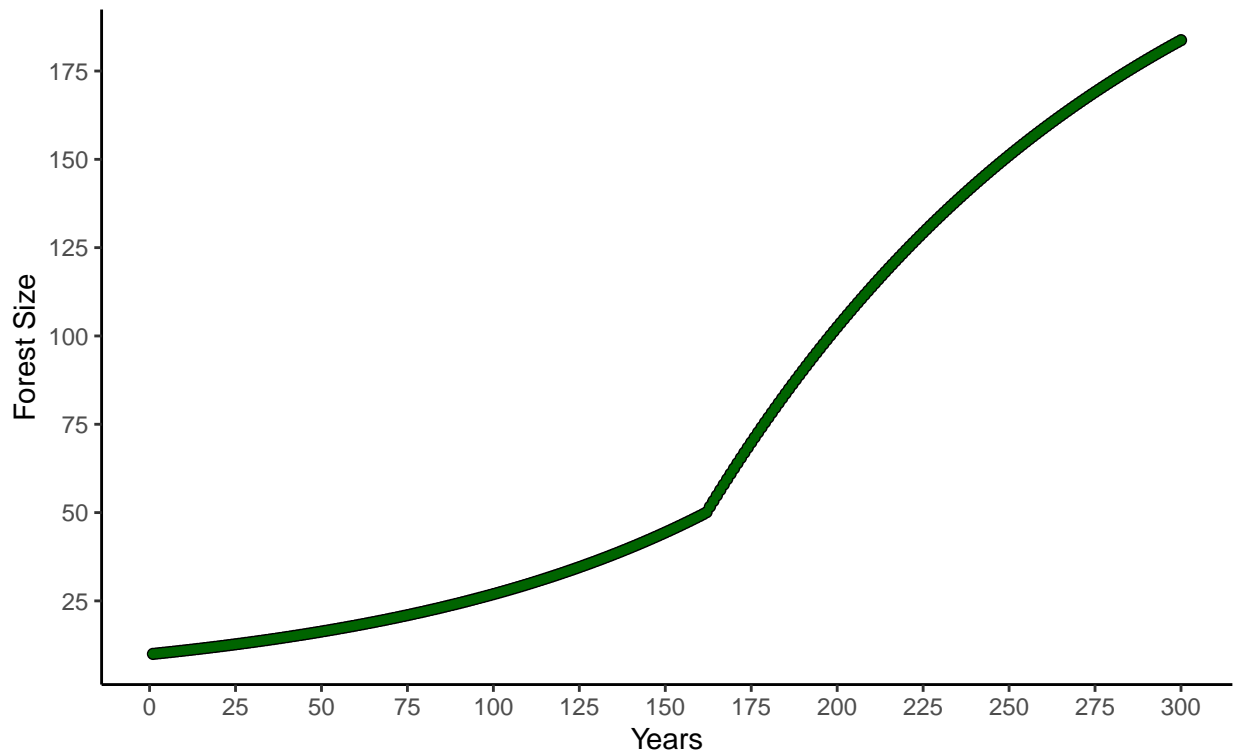
```
## [4,] 4 10.30455
## [5,] 5 10.40811
## [6,] 6 10.51271
```

```
# add more meaningful names
colnames(results) = c("year", "P")
```

Graph the results

```
ggplot(as.data.frame(results), aes(year, P)) +
  geom_point() +
  labs(y="Forest Population",
       x = "Years") +
  theme_classic() +
  geom_point(col = "darkgreen",
            size = 1) +
  ggtitle(label = "Forest Growth after 300 Years of Growth",
         subtitle = "") +
  labs(y = "Forest Size",
       x = "Years") +
  theme_classic() +
  scale_x_continuous(breaks = seq(0, 300, by = 25)) +
  scale_y_continuous(breaks = seq(0, 200, by = 25))
```

Forest Growth after 300 Years of Growth



3. Run a sobol sensitivity analysis that explores how the estimated maximum and mean forest size (e.g maximum and mean values of C over the 300 years) varies with the pre canopy closure growth rate (r) and post-canopy closure growth rate (g) and canopy closure threshold and carrying capacity(K)

Assume that parameters are all normally distributed with means as given above and standard deviation of 10% of mean value.

```
#set parameters
C = 10 #forest size

#number of samples
np = 100

#create first sample parameters from normal distributions
r = rnorm(mean = 0.01, sd = r*.10, n = np)
g = rnorm(mean = 2, sd = g*.10, n = np)
threshold = rnorm(mean = 50, sd = threshold*.10, n = np)
K = rnorm(mean = 250, sd = K*.10, n = np)

#create the first dataframe
X1 = cbind.data.frame(r = r, g = g, threshold = threshold, K = K)

#create second sample parameters from normal distributions (this is just how sobol works)
r = rnorm(mean = 0.01, sd = r*.10, n = np)
g = rnorm(mean = 2, sd = g*.10, n = np)
threshold = rnorm(mean = 50, sd = threshold*.10, n = np)
K = rnorm(mean = 250, sd = K*.10, n = np)

#create the second dataframe
X2 = cbind.data.frame(r = r, g = g, threshold = threshold, K = K)

#create our sobol object and get sets of parameters for running the model
sens_P = sobolSalt(model = NULL, X1, X2, nboot = 300)

colnames(sens_P$X) = c("r",
                      "g",
                      "threshold",
                      "K")

#our parameter sets are
head(sens_P$X)
```

```
##           r           g threshold      K
## [1,] 0.011129649 2.003516  44.68301 238.9361
## [2,] 0.009317186 1.921311  60.92302 240.9093
## [3,] 0.010283425 2.079286  53.09713 203.3721
## [4,] 0.008838717 1.970633  53.68775 219.2354
## [5,] 0.012148239 1.678658  51.77647 265.3529
## [6,] 0.011072621 1.961243  37.86202 251.4618
```

```
# define a wrapper function to do everything we need
# run solver and compute metrics - and send back results for each parameter
```

```
p_wrapper = function(r, g, threshold, K, Pinitial, simtimes, func) {
  parms = list(r = r,
              g = g,
              threshold = threshold,
              K = K)
  result = ode(y = Pinitial,
              times = simtimes,
              func = func,
              parms = parms)
  colnames(result) = c("time", "P")
  # get metrics
  metrics = compute_metrics(as.data.frame(result))
  return(metrics)
}
```

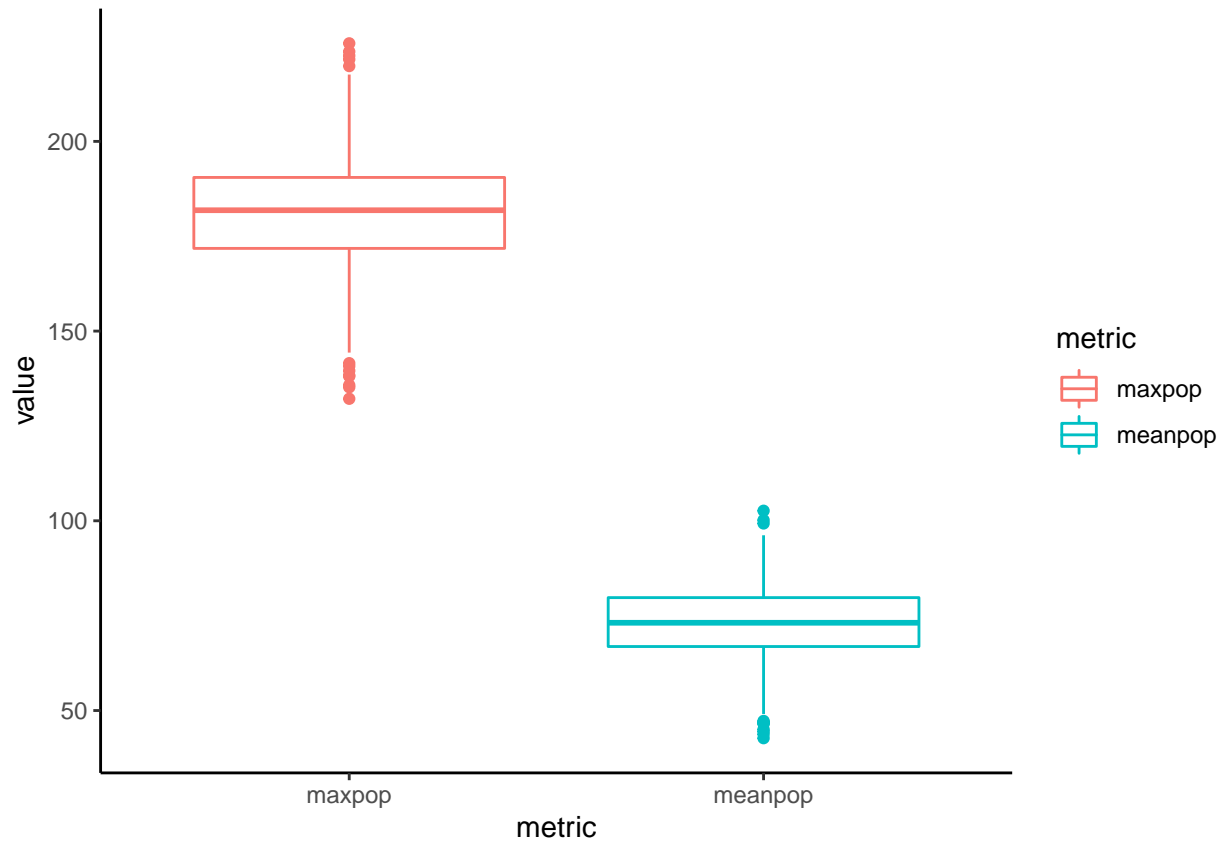
```
# reiterate parameters
Pinitial = 10 #forest size (initial population in units of kg C)
simtimes = seq(from = 1, to = 300) #run the model for 300 years
func = forest_growth
```

```
# now use pmap as we did before
allresults = as.data.frame(sens_P$X) %>%
  pmap(p_wrapper,
      Pinitial = Pinitial,
      simtimes = simtimes,
      func = func)
```

```
# extract out results from pmap into a data frame
allres = allresults %>% map_dfr(`[,c("maxpop", "meanpop")])
```

Graph the results

```
# create boxplots
tmp = allres %>% gather(key = "metric", value = "value")
ggplot(tmp, aes(metric, value, col = metric)) +
  geom_boxplot() +
  theme_classic()
```



```
# compute the sobol indices for maxpop
sens_P_maxpop = sensitivity::tell(sens_P, allres$maxpop)
# compute the sobol indices for meanpop
sens_P_meanpop = sensitivity::tell(sens_P, allres$meanpop)
```

```
# creating df of max T
max_T <- as.data.frame(sens_P_maxpop$T)
```

```
max_T <- max_T %>%
  rowid_to_column(var = "parameter")
```

```
max_T[1,1] <- "r"
max_T[2,1] <- "g"
max_T[3,1] <- "threshold"
max_T[4,1] <- "K"
```

```
# creating df of max S
max_S <- as.data.frame(sens_P_maxpop$S)
```

```
max_S <- max_S %>%
  rowid_to_column(var = "parameter")
```

```
max_S[1,1] <- "r"
max_S[2,1] <- "g"
max_S[3,1] <- "threshold"
max_S[4,1] <- "K"
```

```

# creating df of mean T
mean_T <- as.data.frame(sens_P_meanpop$T)

mean_T <- mean_T %>%
  rowid_to_column(var = "parameter")

mean_T[1,1] <- "r"
mean_T[2,1] <- "g"
mean_T[3,1] <- "threshold"
mean_T[4,1] <- "K"

# creating df of mean S
mean_S <- as.data.frame(sens_P_meanpop$S)

mean_S <- mean_S %>%
  rowid_to_column(var = "parameter")

mean_S[1,1] <- "r"
mean_S[2,1] <- "g"
mean_S[3,1] <- "threshold"
mean_S[4,1] <- "K"

```

```

# create S and T plots
max_T_plot <- ggplot(max_T, aes(x = original, y = parameter)) +
  geom_col(fill = "darkgreen") +
  theme_classic() +
  labs(title = "Max Total Sensitivity Index",
       x = "Sobol Score",
       y = "Parameter")

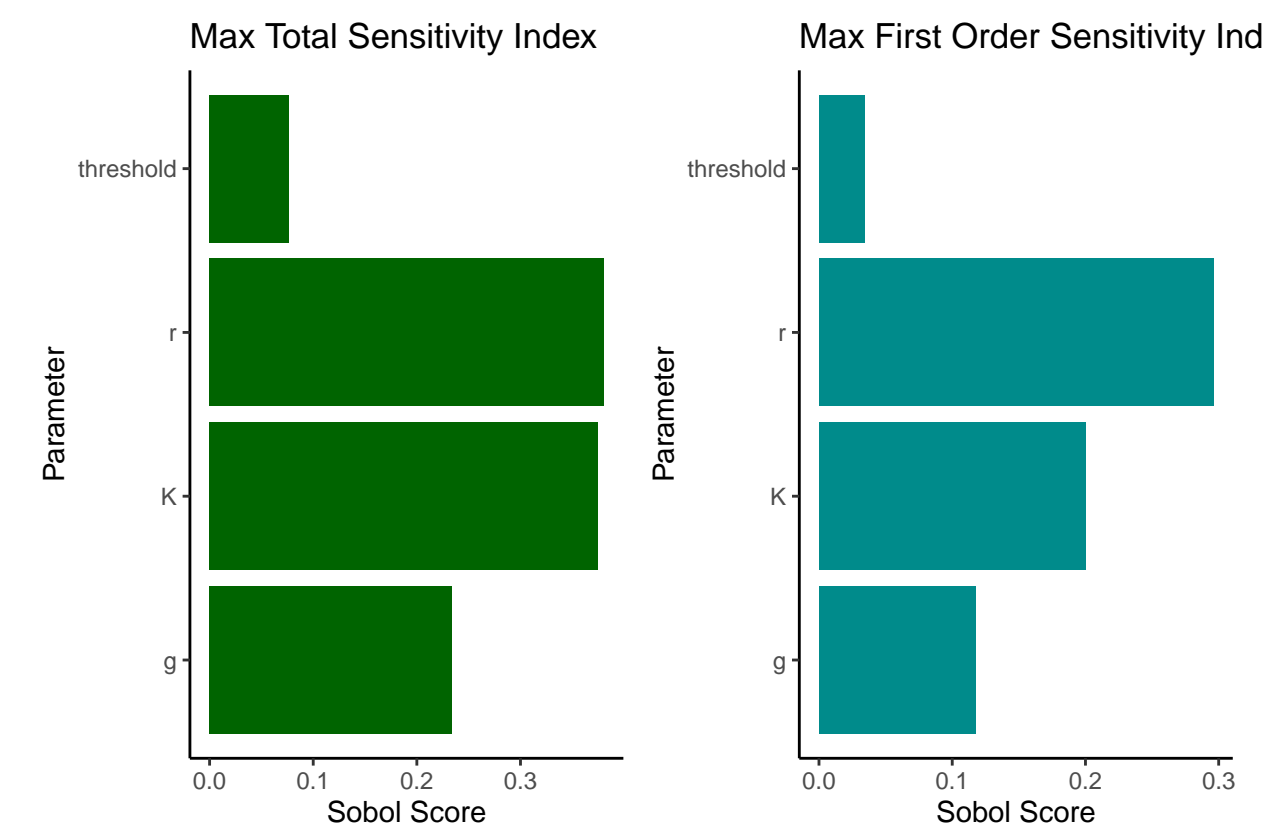
max_S_plot <- ggplot(max_S, aes(x = original, y = parameter)) +
  geom_col(fill = "cyan4") +
  theme_classic() +
  labs(title = "Max First Order Sensitivity Index",
       x = "Sobol Score",
       y = "Parameter")

mean_T_plot <- ggplot(mean_T, aes(x = original, y = parameter)) +
  geom_col(fill = "darkgreen") +
  theme_classic() +
  labs(title = "Mean Total Sensitivity Index",
       x = "Sobol Score",
       y = "Parameter")

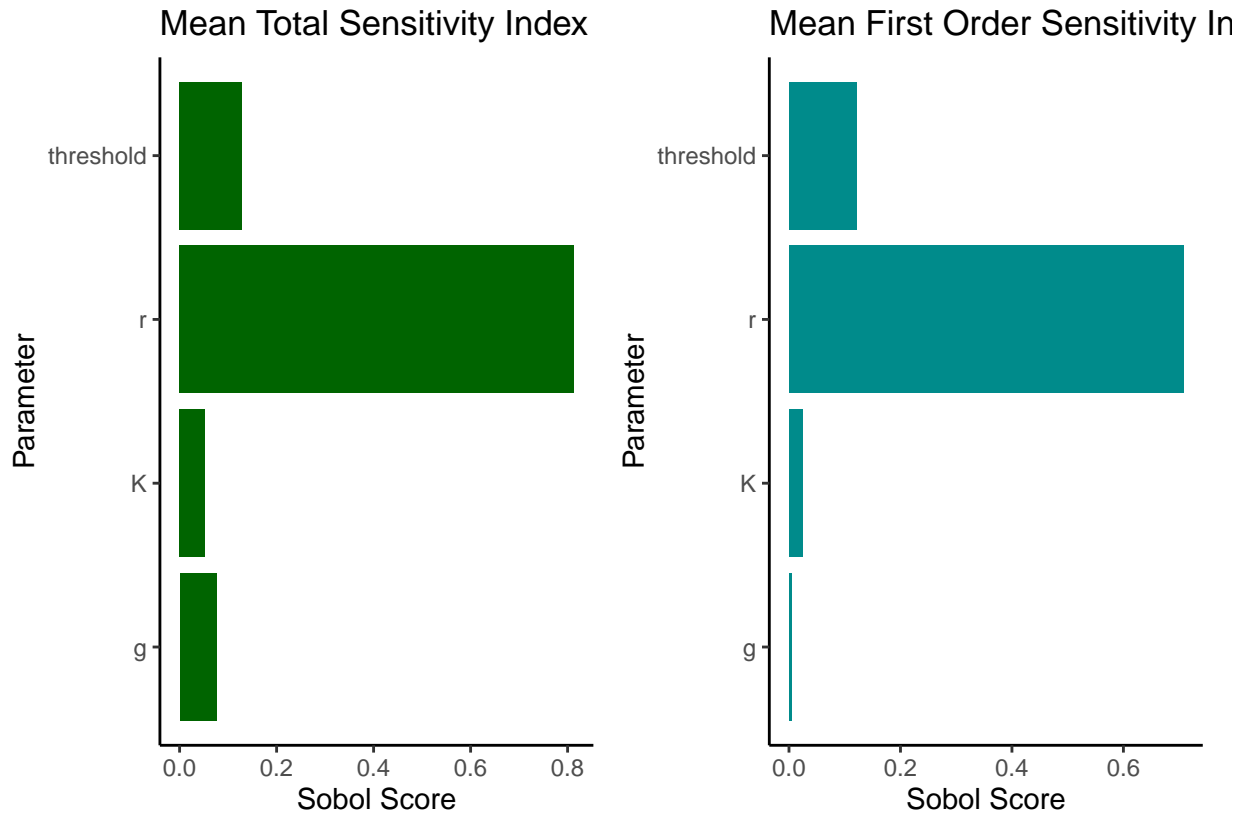
mean_S_plot <- ggplot(mean_S, aes(x = original, y = parameter)) +
  geom_col(fill = "cyan4") +
  theme_classic() +
  labs(title = "Mean First Order Sensitivity Index",
       x = "Sobol Score",
       y = "Parameter")

# patchwork plots
max_T_plot + max_S_plot

```



```
mean_T_plot + mean_S_plot
```

In 2-3 sentences, discuss what the results of your simulation might mean for climate change impacts on forest growth (e.g think about what parameters climate change might influence).

The results of our simulation show that the exponential growth rate (r) and the carrying capacity (K) are the most influential parameters for max forest population. Whereas the exponential growth rate (r) was the single most significant parameter for mean forest population. However, climate change might influence certain parameters. Due to warmer temperatures lasting longer periods of time, trees may have a longer growing season, with more time to photosynthesize. This can lead to increased forest growth and carbon sequestration. However, recent research has also found that trees are growing faster but dying earlier, which regresses their capacity to store carbon emissions.