EDS 230 Assignment 6: Using Sobol with an ODE

Erika Egg

2023-05-24

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

```
# Source the function
source("calc_forest_growth.R")
```

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 before canopy closure)
- g = 2 kg/year (linear growth rate after canopy closure)

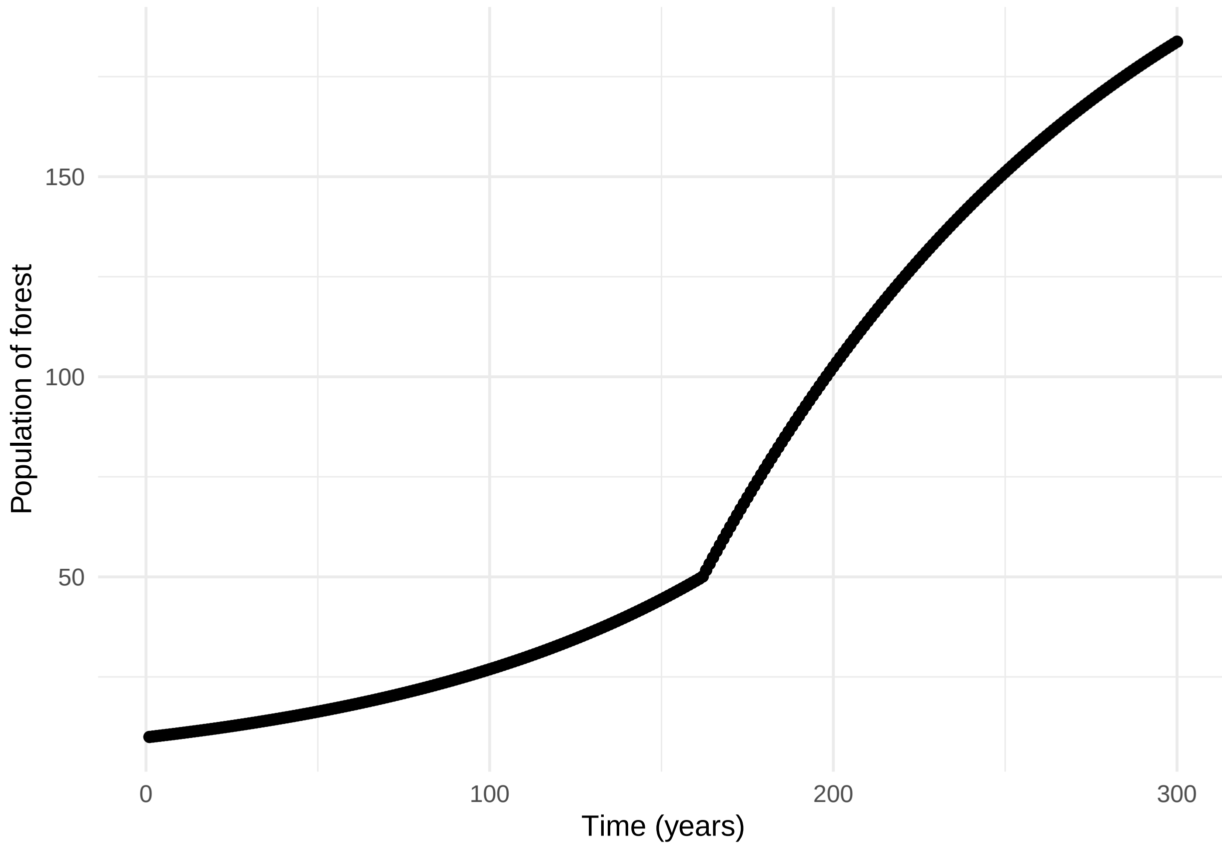
```
# Run the model for the specified values
# Set values
time <- seq(from = 1, to = 300) # 300 years
C <- 10
threshold <- 50
K <- 250
r <- 0.01
g <- 2

# Create obj to pass to func
parms <- list(time = time, threshold = threshold, K = K, r = r, g = g)

# Run our differential equation and keep the output
result <- ode(y = C, times = time, func = calc_forest_growth, parms = parms)

head(result)
```

```
##      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
```

```
colnames(result) = c("time","P")
```

3. Graph the results. Here you are graphing the trajectory with the parameters as given (e.g no uncertainty)

```
# Create graph
result <- as.data.frame(result)
ggplot(result, aes(time, P)) +
  geom_point() +
  labs(x = "Time (years)",
       y = "Population of forest") +
  theme_minimal()
```



4. Run a sobol global (vary all parameters at the same time) sensitivity analysis that explores how the estimated maximum forest size (e.g maximum of 300 years, varies with these parameters

- pre canopy closure growth rate (r)
- post-canopy closure growth rate (g)
- 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

```
# Come up with first set of sample parameters
# We will assume that we know C and threshold

C <- 10 # same as before

# Want to learn about sensitivity to threshold, r, g, and K
```

```
# Set the number of parameters
np <- 100

# Create first sample parameters from normal distributions
r <- rnorm(mean = 0.01, sd = r * 0.10, n = np)
g <- rnorm(mean = 2, sd = g * 0.10, n = np)
threshold <- rnorm(mean = 50, sd = threshold * 0.10, n = np)
K <- rnorm(mean = 250, sd = K * 0.10, n = np)

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

# Repeat to get our second set of samples
r <- rnorm(mean = 0.01, sd = r * 0.10, n = np)
g <- rnorm(mean = 2, sd = g * 0.10, n = np)
threshold <- rnorm(mean = 50, sd = threshold * 0.10, n = np)
K <- rnorm(mean = 250, sd = K * 0.10, n = np)

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

# Create our sobel object and get sets ofparameters for running the model
sens_P <- sobolSalt(model = NULL, X1, X2, nboot = 300)

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

```
##              [,1]     [,2]     [,3]     [,4]
## [1,] 0.010038930 2.118294 42.48301 259.3097
## [2,] 0.012079812 1.979853 50.00363 235.0244
## [3,] 0.010503247 2.058683 45.50773 245.9921
## [4,] 0.009118422 2.354002 47.05663 251.4470
## [5,] 0.011688051 2.132589 54.77430 245.3230
## [6,] 0.009367174 2.296448 49.89001 253.2190
```

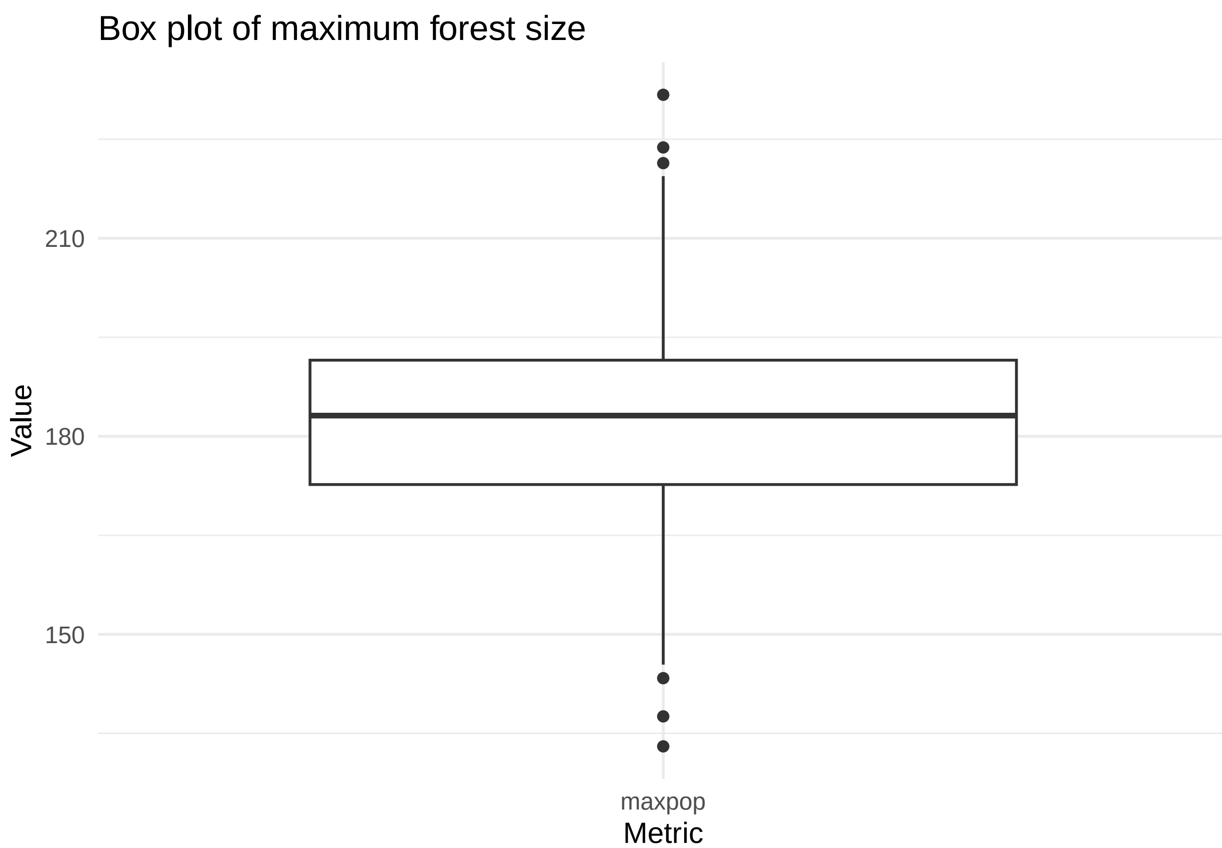
```
# Let's add names
colnames(sens_P$X) <- c("r", "g", "threshold", "K")
```

```
# Look at maximums
# Turn computing our metrics into a function (use one from class, but just max)

compute_metric <- function(result) {
  maxpop = max(result$P)
return(list(maxpop=maxpop))}

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

# Keep threshold same as before

p_wrapper <- function(threshold, r, g, K, C, time, func) {
    parms <- list(threshold = threshold, r = r, g = g, K = K)
```

```r
    result <- ode(y = C, time = time, func = func, parms = parms)
    colnames(result) <- c("time", "P")
  # Get metrics
  metrics <- compute_metric(as.data.frame(result))
  return(metrics)
}

# Now use pmap as we did before
allresults <- as.data.frame(sens_P$X) %>% pmap(p_wrapper,
                                                C = C,
                                                time = time,
                                                func = calc_forest_growth)

# Extract out results from pmap into a data frame
allres <- allresults %>% map_dfr(`[`,c("maxpop"))
```

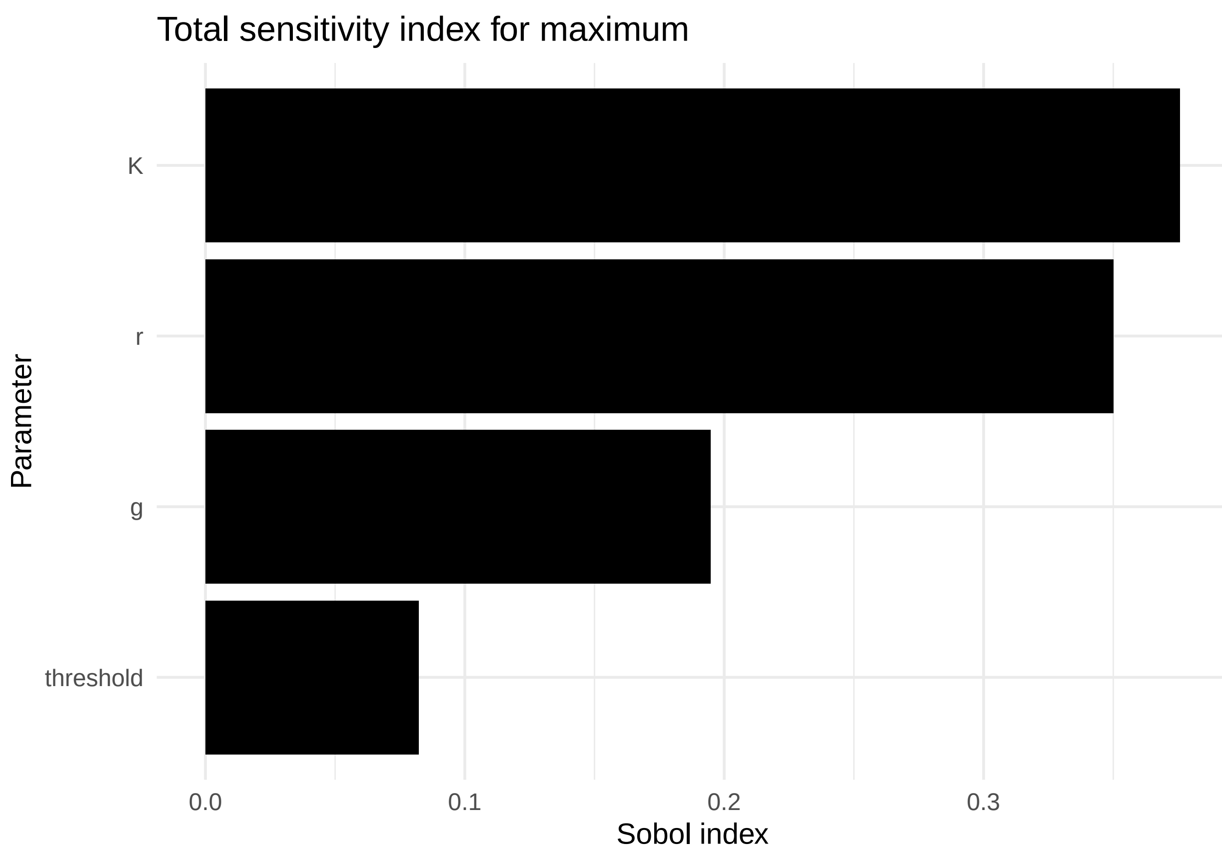
5. Graph the results of the sensitivity analysis as a box plot of maximum forest size and record the two Sobol indices (S and T).

```r
# Create metrics graph (box plot)
tmp <- allres %>% pivot_longer(cols = everything(), names_to = "metric", values_to = "value")
ggplot(tmp, aes(metric, value)) +
  geom_boxplot() +
  theme_minimal() +
  labs(x = "Metric",
       y = "Value",
       title = "Box plot of maximum forest size")
```

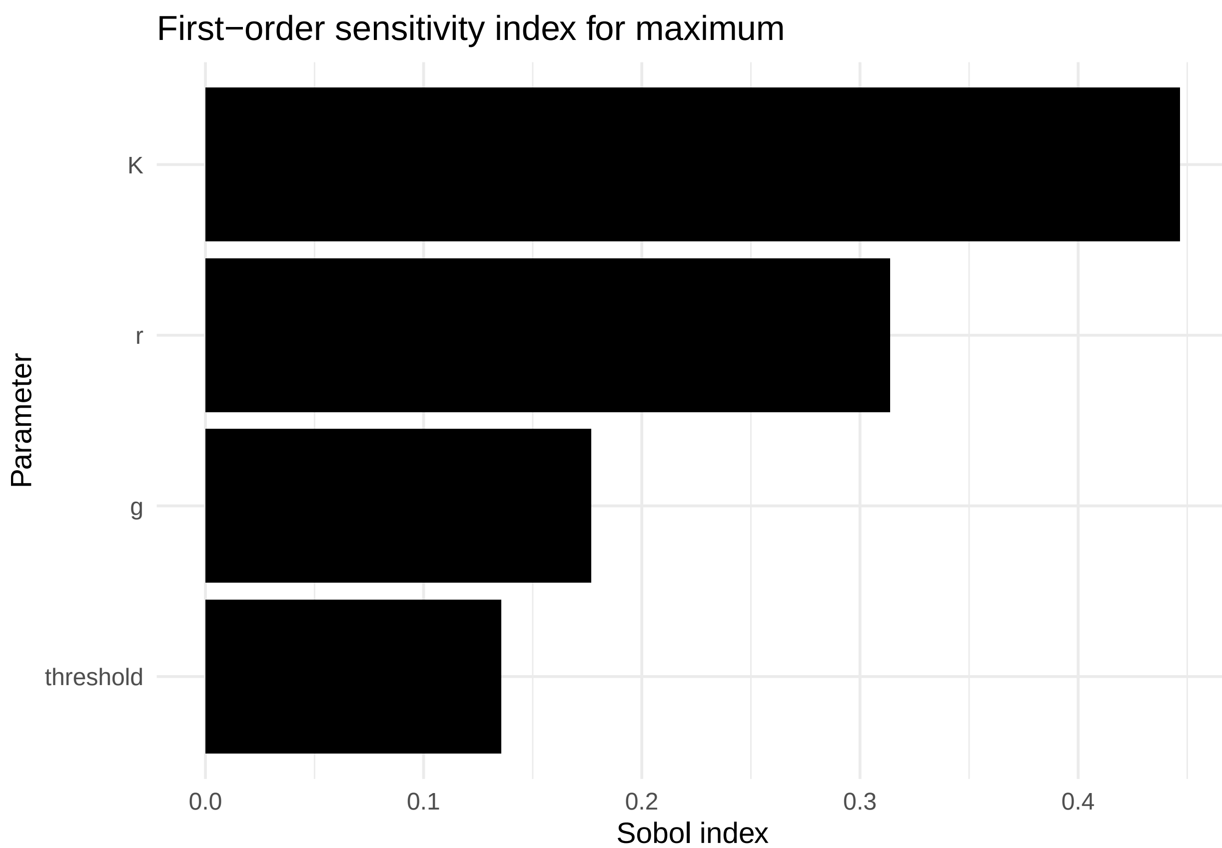



```r
# Prep to create S & T index graphs
# Sobol can only handle one output at a time  - so we will need to do them separately

sens_P_maxpop = sensitivity::tell(sens_P, allres$maxpop)

# First-order indices (main effect without co-variance)
max_S <- as.data.frame(sens_P_maxpop$S)

# Total sensitivity index -note that this partitions the output variance
max_T <- as.data.frame(sens_P_maxpop$T)

# Get ready to plot these two dfs
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"

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

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

```r
max_T[4,1] <- "K"

# PLot these
ggplot(max_T, aes(x = original, y = reorder(parameter, original))) +
  geom_col(fill = "black") +
  theme_minimal() +
  labs(title = "Total sensitivity index for maximum",
       x = "Sobol index",
       y = "Parameter")
```




```r
ggplot(max_S, aes(x = original, y = reorder(parameter, original))) +
  geom_col(fill = "black") +
  theme_minimal() +
  labs(title = "First-order sensitivity index for maximum",
       x = "Sobol index",
       y = "Parameter")
```




```
# Show in table form too
max_T %>%
  kbl(caption = "Total sensitivity index for maximum") %>%
  kable_paper(full_width = F) %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Total sensitivity index for maximum

parameter	original	bias	std. error	min. c.i.	max. c.i.
r	0.3499965	0.0037683	0.0759645	0.1957885	0.4986706
g	0.1946943	0.0076315	0.0387573	0.0993905	0.2532956
threshold	0.0821735	0.0024519	0.0180361	0.0405281	0.1148872
K	0.3755915	0.0068962	0.0663624	0.2482183	0.4925421

```
max_S %>%
  kbl(caption = "First-order sensitivity index for maximum") %>%
  kable_paper(full_width = F) %>%
   kable_styling(latex_options = "HOLD_position")
```

Table 2: First-order sensitivity index for maximum

parameter	original	bias	std. error	min. c.i.	max. c.i.
r	0.3137768	-0.0064368	0.0870404	0.1483417	0.4922370
g	0.1767939	-0.0091273	0.0893655	0.0028116	0.3464308
threshold	0.1356047	-0.0064926	0.0908629	-0.0370203	0.3106688
K	0.4465302	-0.0045952	0.0932007	0.2855649	0.6409438

6. In 2-3 sentences, discuss what the results of your simulation might mean. (For example think about how what parameters climate change might influence).

Parameters K and r have the highest total sensitivity index and first-order sensitivity index (K is higher for both). This means that changes in these due to climate change (ie. changing temperature impacting carrying capacity and exponential growth rate) could change forest growth more significantly. We should keep this under consideration when using a model such as this.