Forest Growth Modeling

Paloma Cartwright Joe DeCesaro Connor Flynn

2022-05-18

Consider the following model of forest growth (where forest size in measured in 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 initial size of the forest, **C**, canopy closure threshold, **threshold**, and carrying capacity, **K**, are all in units of kgC. 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 the model in R

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

```
## function (time, C, params)
## {
##     if (C < params$threshold) {
##         dC_dt <- params$r * C
##         return(list(dC_dt))
##     }
##     else {
##         dC_dt <- params$g * (1 - C/params$K)
##         return(list(dC_dt))
##     }
## }
```

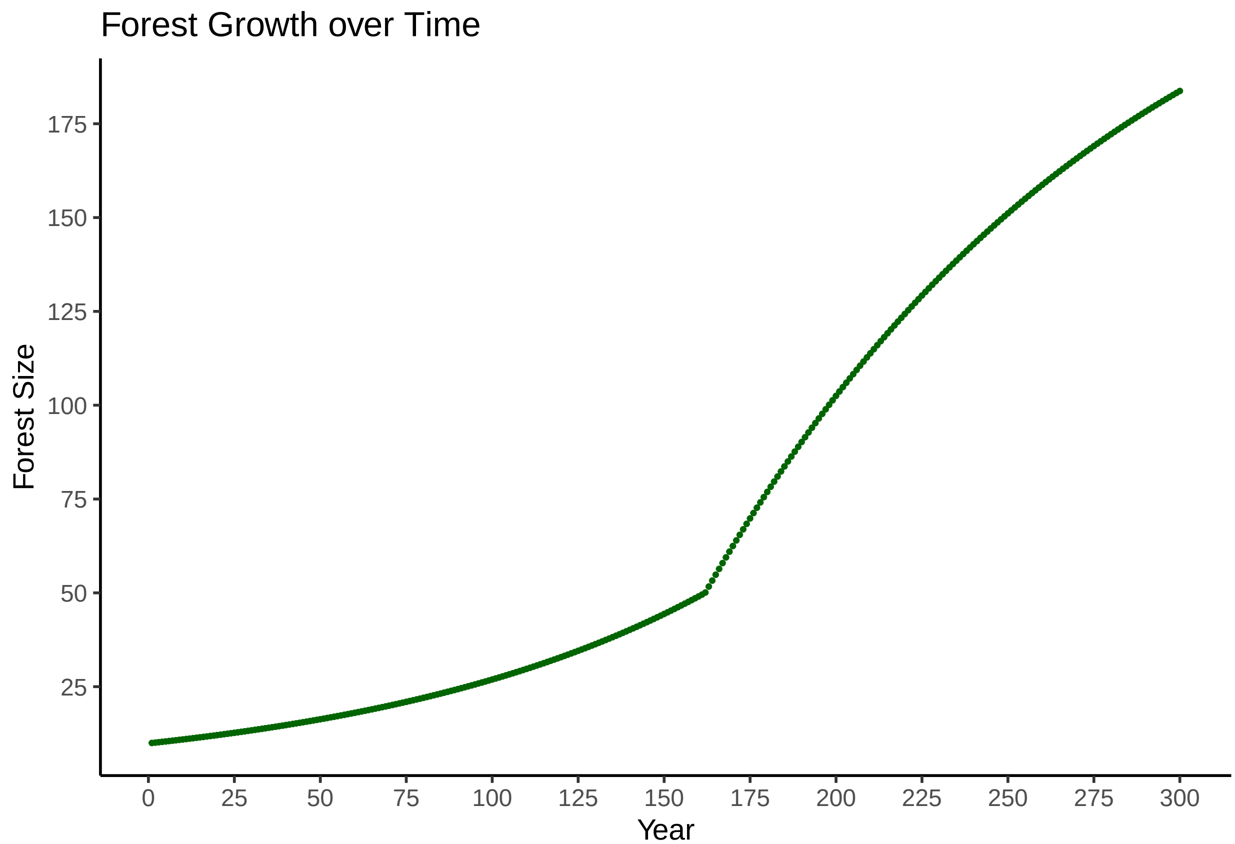
2. Run the model for 300 years

- Start with an initial forest size of 10kgC
- Parameters:
  - Canopy Closure Threshold, threshold: 50kgC
  - Carrying Capacity, K: 250kgC
  - Early Exponential Growth rate, r: 0.01
  - Linear Growth rate, g: 2kg/year

```r
time = seq(from = 1, to = 300, by = 1) # 300 years
C = 10 # forest size
params = list(threshold = 50,  # canopy closure
              r = 0.01, # exponential growth rate
              g = 2, # linear growth rate
              K = 250)

forest_growth_300 <- ode(y = C,
                         times = time,
                         func = forest_growth,
                         parms = params) %>%
  as.data.frame()

colnames(forest_growth_300) = c("year", "forest_size")
```

Graph the Results

```r
ggplot(forest_growth_300, aes(x = year, y = forest_size)) +
  geom_point(color = "darkgreen",
             size = 0.5) +
  labs(title = "Forest Growth over Time",
       x = "Year",
       y = "Forest Size") +
  theme_classic() +
  scale_x_continuous(breaks = seq(0, 300, by = 25)) +
  scale_y_continuous(breaks = seq(0, 200, by = 25))
```


Forest Growth over Time
Forest Size
175
150
125
100
75
50
25
0
25
50
75
100
125
150
175
200
225
250
275
300
Year

3. Run a Sobol Sensitivity Analysis

Explore 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.

```r
# conditions
threshold = 50  # canopy closure
r = 0.01 # exponential growth rate
g = 2 # linear growth rate
K = 250

C = 10 # forest size
np = 100 # number of samples

threshold = rnorm(mean = threshold, sd = threshold * 0.1, n = np)
r = rnorm(mean = r, sd = r * 0.1, n = np)
g = rnorm(mean = g, sd = g * 0.1, n = np)
K = rnorm(mean = K, sd = K * 0.1, n = np)

X1 = cbind.data.frame(threshold = threshold,
                      r = r,
                      g = g,
                      K = K)

np = 100
threshold = rnorm(mean = threshold, sd = threshold * 0.1, n = np)
r = rnorm(mean = r, sd = r * 0.1, n = np)
g = rnorm(mean = g, sd = g * 0.1, n = np)
K = rnorm(mean = K, sd = K * 0.1, n = np)
X2 = cbind.data.frame(threshold = threshold,
                      r = r,
                      g = g,
                      K = K)

sens_forestSize <- sobolSalt(model = NULL, X1, X2, nboot = 300)

sens_forestSize_df <- sens_forestSize$X %>%
  as.data.frame()
sens_forestSize_df <- sens_forestSize_df %>%
  rename(threshold = V1,
         r = "V2",
         g = "V3",
         K = "V4")

head(sens_forestSize_df)
```

```
##   threshold           r        g        K
## 1  53.54775 0.009808867 1.896362 251.0471
## 2  54.90168 0.011433368 2.053482 336.9698
## 3  49.16321 0.009316993 2.233171 275.7631
```

```
## 4  54.96426 0.008236300 2.193489 289.3946
## 5  53.42038 0.008947900 2.162877 248.8390
## 6  53.69156 0.009254309 2.010428 226.8552
```

Read in the Compute Metrics Function for Sobol Analysis

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

```
## function (result)
## {
##     maxsize = max(result$forest_size)
##     meansize = mean(result$forest_size)
##     return(list(maxsize = maxsize, meansize = meansize))
## }
```

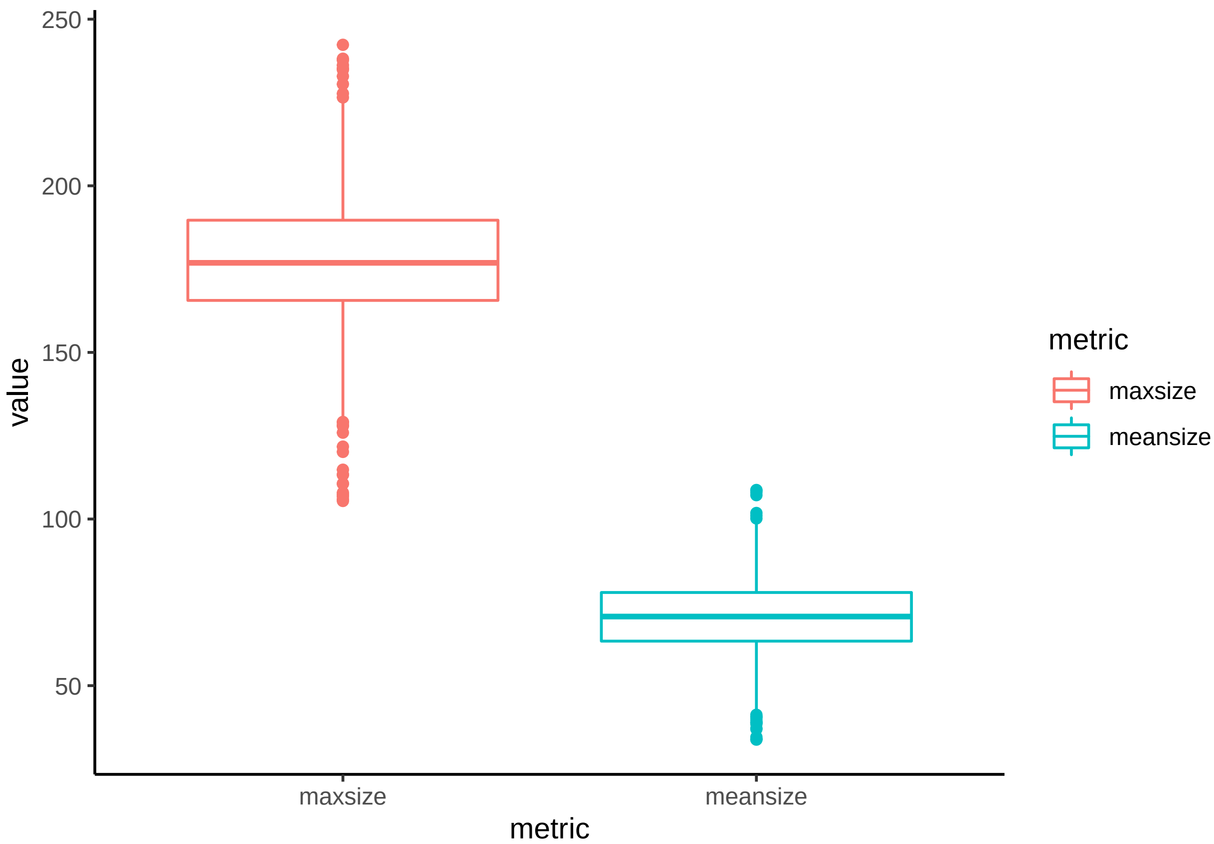
Create the wrapper function for running compute metrics and the ode solver

```r
p_wrapper = function(threshold, r, g, K, Cinitial, times, func) {
  params = list(threshold = threshold,
                r = r,
                g = g,
                K = K)
  result = ode(y = Cinitial,
               times = time,
               func = func,
               parms = params,
               method = "daspk")
  colnames(result) = c("year", "forest_size")
  # get metrics
  metrics = compute_metrics(as.data.frame(result))
  return(metrics)
}

allresults = sens_forestSize_df %>%
  pmap(p_wrapper,
       Cinitial = C,
       times = time,
       func = forest_growth)

allres = allresults %>%
  map_dfr(`[`,c("maxsize","meansize"))
```

Graph the results of the sensitivity analysis

as a box plot of maximum forest size and a plot of the two Sobol indices (S and T).

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

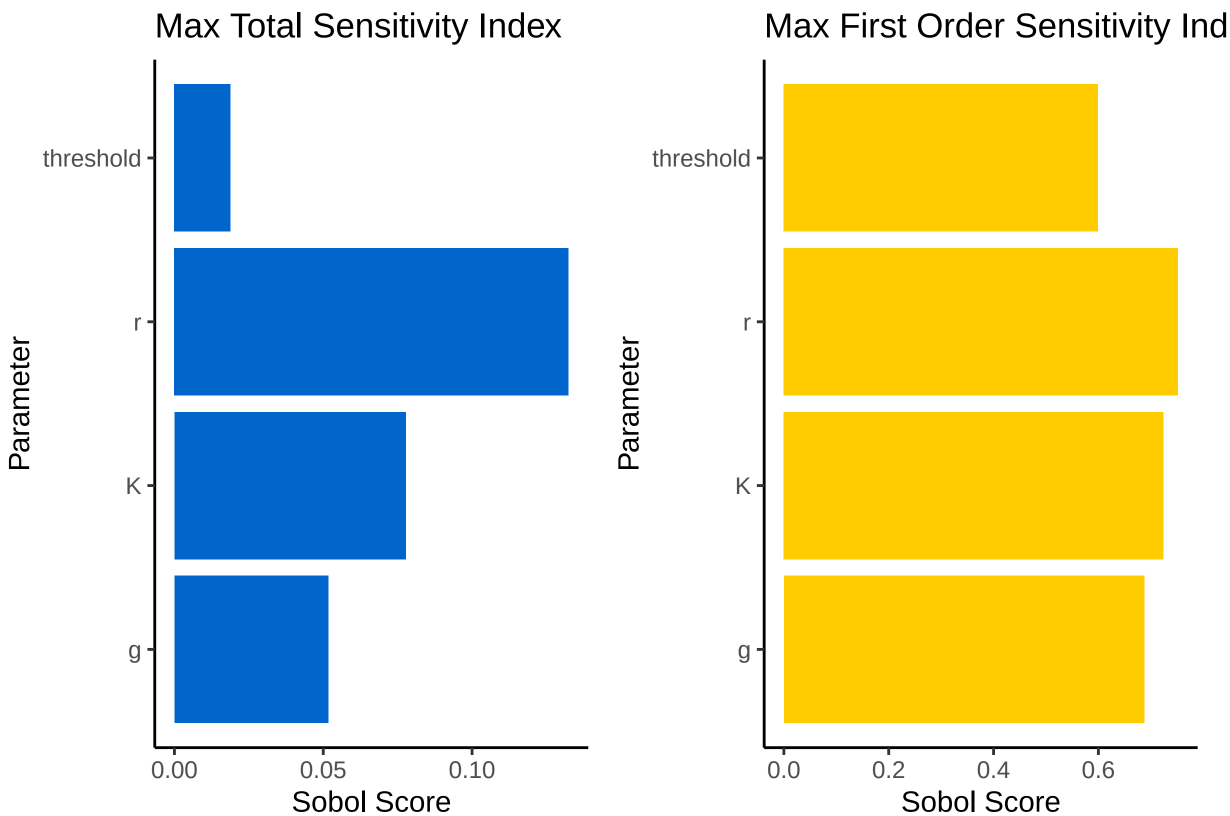



```
# sobol can only handle one output at a time  - so we will need to do them separately, let's start with
sens_forest_maxSize = sensitivity::tell(sens_forestSize, allres$maxsize)

# first-order indices (main effect without co-variance)
max_S <- as.data.frame(sens_forest_maxSize$S) %>% # make it a dataframe
  rowid_to_column(var = "parameter") # make rowids into parameters so graphing is easier

# replace rowids with variables names that matches order input above
max_S[1,1] <- "threshold"
max_S[2,1] <- "r"
max_S[3,1] <- "g"
max_S[4,1] <- "K"

# total sensitivity index - note that this partitions the output variance - so values sum to 1
max_T <- as.data.frame(sens_forest_maxSize$T) %>% # make it a dataframe
  rowid_to_column(var = "parameter") # make rowids into parameters so graphing is easier

# replace rowids with variables names that matches order input above
max_T[1,1] <- "threshold"
max_T[2,1] <- "r"
```

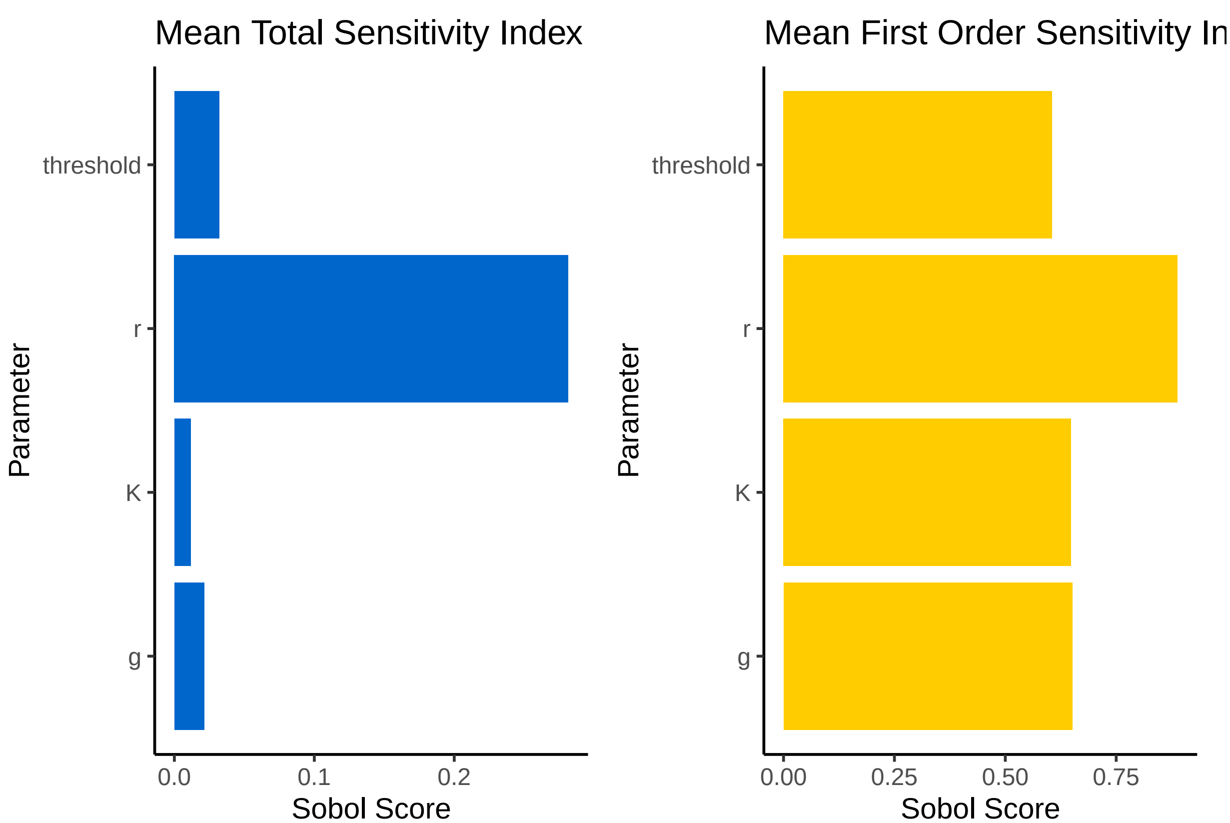
```
max_T[3,1] <- "g"
max_T[4,1] <- "K"
```

```
# create another one for mean size
sens_forest_meanSize = sensitivity::tell(sens_forestSize, allres$meansize)

# first-order indices (main effect without co-variance)
mean_S <- as.data.frame(sens_forest_meanSize$S) %>% # make it a dataframe
  rowid_to_column(var = "parameter") # make rowids into parameters so graphing is easier

# replace rowids with variables names that matches order input above
mean_S[1,1] <- "threshold"
mean_S[2,1] <- "r"
mean_S[3,1] <- "g"
mean_S[4,1] <- "K"

# total sensitivity index - note that this partitions the output variance - so values sum to 1
mean_T <- as.data.frame(sens_forest_meanSize$T) %>% # make it a dataframe
  rowid_to_column(var = "parameter") # make rowids into parameters so graphing is easier

# replace rowids with variables names that matches order input above
mean_T[1,1] <- "threshold"
mean_T[2,1] <- "r"
mean_T[3,1] <- "g"
mean_T[4,1] <- "K"
```

```
# create S and T plots
max_T_plot <- ggplot(max_T, aes(x = original, y = parameter)) +
  geom_col(fill = "#0066cc") +
  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 = "#ffcc00") +
  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 = "#0066cc") +
  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 = "#ffcc00") +
  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
```



Simulation results: what this may mean regarding the impact of climate change on forest growth

For the Maximum Total Sensitivity Index, the parameter with the highest sensitivity is carrying capacity. For the Mean Total Sensitivity Index, the parameter with the highest sensitivity is the exponential growth rate. Forest carbon offsets have emerged as one of the fastest growing finance tools to incentivize forest conservation, so it is very interesting to see that the carbon carying capacity of forests is so highly sensitive. With climate change we are seeing extended droughts, more severe forest fires, increase pests, and replacement of forests with grasslands and shrubs. All of these factors will strongly influence the carbon carrying capacity of forests, and the carbon offset market should recognize this.