EDS230/ESM232: Modeling Growth

Juliet Cohen, Dani Sclafani, Nikole Vannest

2022-05-13

Modeling Forest Growth in Carbon

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 and `K` is a carrying capacity in units of carbon

- The size of the forest (`C`), Canopy closure threshold and carrying capacity are all in units of `kg 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**) (Grading 25% - we will look for correct implementation but also good programming style - meaningful variable names, comments/documentation throughout)

```
# Load in function
source(here("R", "C_growth.R"))

## C_growth <- function(time, C, parms) { #need time as a parameter because we are running the function
##   if (C < parms$canopy_thresh){
##     c_list = parms$r*C
##   }
##   else {
##     c_list = parms$g * (1 - C/parms$K)
##   }
##   return(list(c_list)) # the ODE solver needs the output to be a list
## }

# Keep samples the same
set.seed(123)
```

2. Run the model for 300 years (using the ODE solver)and using the following parameters:

- C = Initial forest size of 10 kgC
- canopy_thresh = Canopy closure threshold of 50 kgC
- K = 250 kgC (carrying capacity)
- r = 0.01 (exponential growth rate before before canopy closure)
- g = 2 kg/year (linear growth rate after canopy closure)

Graph the results. (15% - we look for an appropriate graphs and good visualization practice - labels, etc)

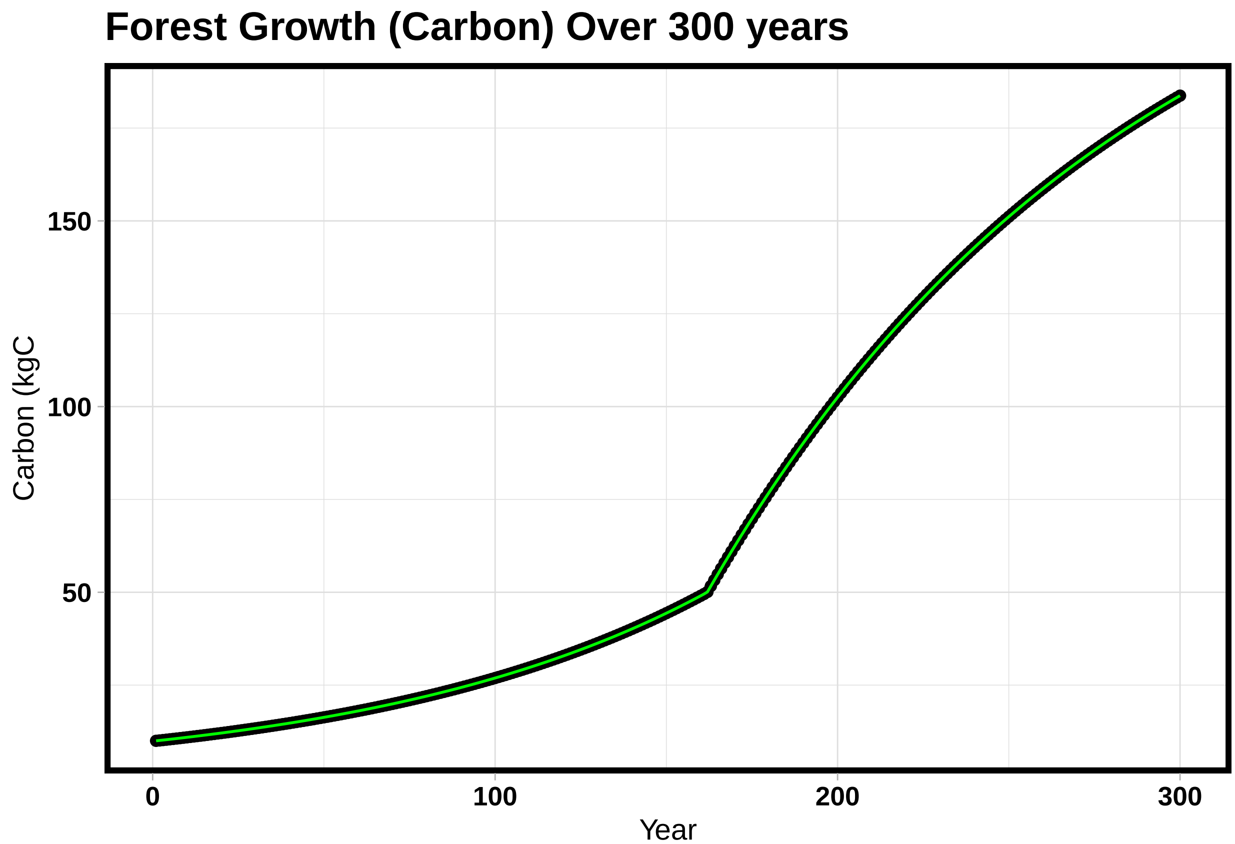
ODE Solver

```r
# define time range in years
simtimes = seq(from = 1, to = 300, by = 1)

# for now the parameter(s) can be static values, but when we feed these paramaters into sobol sensitivi
r = 0.01 #exponential rate of growth
K = 250 #carrying capacity
g = 2 #linear rate of growth
canopy_thresh = 50
C = 10 #starting carbon point
parms <- list(r = r, K = K, g = g, canopy_thresh = canopy_thresh)

# run the ODE solver
C_over_time = ode(y = C, #initial condition, then the next time stamp, ODE uses the initial condition a
            times = simtimes,
            func = C_growth,
            parms = parms)

# rename columns for clarity: change "time" to "Year" and "1" to "Carbon"
colnames(C_over_time) = c("Year", "Carbon")

head(C_over_time)
```

```
##      Year   Carbon
## [1,]    1 10.00000
## [2,]    2 10.10050
## [3,]    3 10.20202
## [4,]    4 10.30455
## [5,]    5 10.40811
## [6,]    6 10.51271
```

Plot Results of ODE Solver

```r
# Plot results of the ODE Solver
ggplot(as.data.frame(C_over_time), aes(Year, Carbon)) +
  geom_point() +
  labs(y = "Carbon",
       x = "Years") +
  geom_line(aes(Year, Carbon), color = 'green') +
  theme_light() +
  ggtitle("Forest Growth (Carbon) Over 300 years") +
   xlab("Year") +
   ylab("Carbon (kgC") +
   theme(axis.title.x = element_text(color = "black", size = 11),
         axis.text.x = element_text(face = "bold", color = "black", size = 10),
         axis.title.y = element_text(color = "black", size = 11),
         axis.text.y = element_text(face = "bold", color = "black", size = 10),
         plot.title = element_text(color="black", size = 15, face = "bold"),
         panel.border = element_rect(colour = "black", fill = NA, size = 2))
```



Sobol Sensitivity Analysis

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

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). (25% - correct implementation of Sobol and graph We also look for good graphing style -labels etc)

```
# We want to learn about sensitivity to pre-canopy growth rate (r), post-canopy closure growth rate (g)

## CREATE FIRST SET OF SAMPLES
# define the number of parameters (called "observations" in the documentation for the rnorm() function)
np = 100

# define K's standard deviation, which is 10% of the mean of 250
K_sd <-  0.1*250

# define K, using the default mean provided, the calculated standard deviation, and the number of obser
K <-  rnorm(mean = 250, sd = K_sd, n = np)

# define r's standard deviation, which is 10% of the mean of 0.01
r_sd <- 0.1*0.01
```

```r
# define r, pre-canopy closure growth rate, using the default mean provided, the calculated standard de
r = rnorm(mean = 0.01, sd = r_sd, n = np)

# define g's standard deviation, which is 10% of the mean of 2
g_sd <- 0.1*2

# define g, the post-canopy closure growth rate, using the default mean provided, the calculated standa
g = rnorm(mean = 2, sd = g_sd, n = np)

# define canopy_thresh's standard deviation, which is 10% of the mean of 50
canopy_thresh_sd = 0.1*50

# define canopy_thresh, the size of the forest at which growth rates change from exponential to linear
canopy_thresh = rnorm(mean = 50, sd = canopy_thresh_sd, n = np)

# bind all 3 parameters together
X1 = cbind.data.frame(r = r, K = K, g = g, canopy_thresh = canopy_thresh)

## CREATE SECOND SET OF SAMPLES (for sensitivity analysis)
# these are the same arguments that defined K and r for the first set sample set
K <-  rnorm(mean = 250, sd = K_sd, n = np)
r <-  rnorm(mean = 0.01, sd = r_sd, n = np)
g <-  rnorm(mean = 2, sd = g_sd, n = np)
canopy_thresh <- rnorm(mean = 50, sd = canopy_thresh_sd, n = np)
X2 <- cbind.data.frame(r = r, K = K, g = g, canopy_thresh = canopy_thresh)

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

# lets add names to the columns
colnames(sens_C$X) = c("r", "K", "g", "canopy_thresh")

# check that the columns and names look correct
head(sens_C$X)
```

```
##                r        K        g canopy_thresh
## [1,] 0.009289593 235.9881 2.439762      46.42379
## [2,] 0.010256884 244.2456 2.262483      46.23656
## [3,] 0.009753308 288.9677 1.946971      45.30731
## [4,] 0.009652457 251.7627 2.108639      44.73743
## [5,] 0.009048381 253.2322 1.917132      47.81420
## [6,] 0.009954972 292.8766 1.904751      51.65590
```

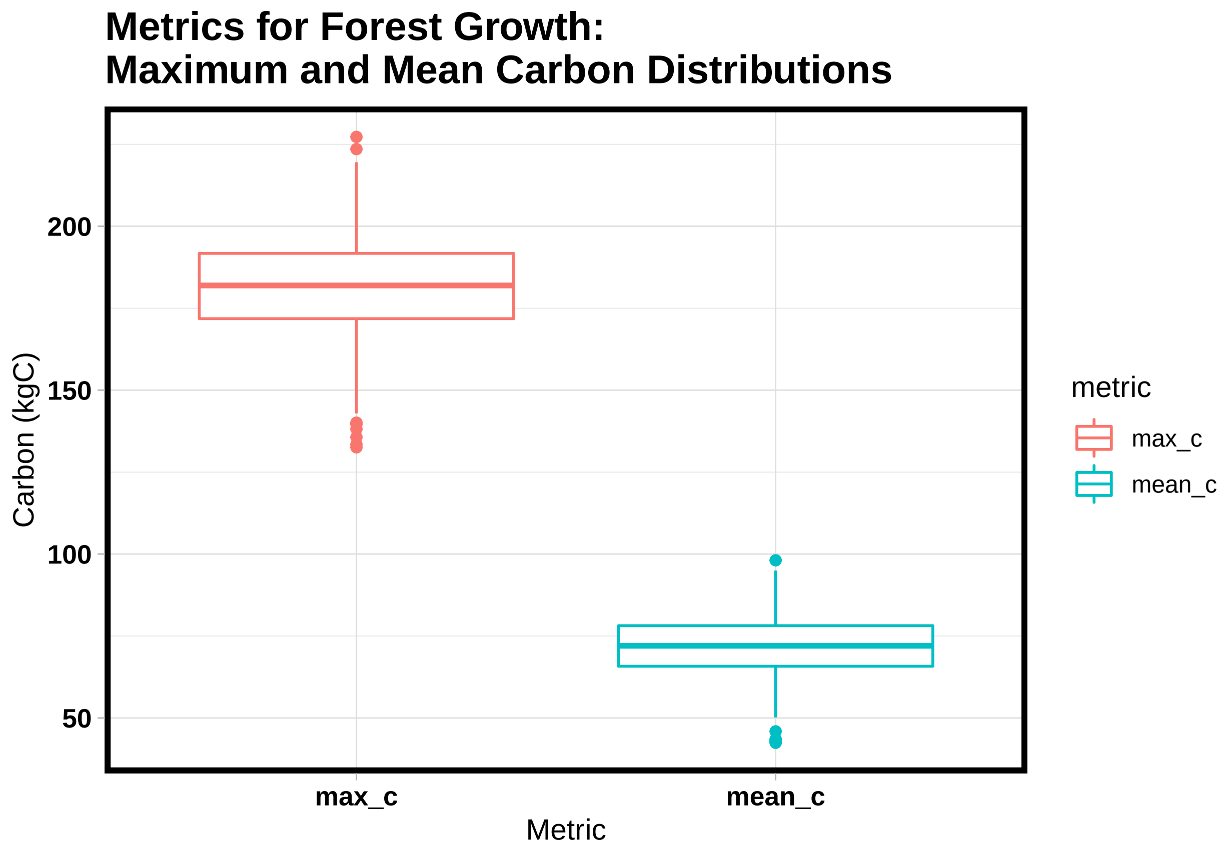
Calculate Metrics and Results

```r
# Function to caculate max and mean forest C

compute_metrics <- function(result){
  max_c <- max(result$C)
  mean_c <- mean(result$C)
  return(list(max_c = max_c, mean_c = mean_c))
}

# note: `simtimes` and `C` objects were already defined earlier in this Rmd when we tested the `C_growt
```

```r
c_wrapper = function(r, g, K, canopy_thresh, C, simtimes, func) {
    parms = list(r = r,
                 g = g,
                 K = K,
                 canopy_thresh = canopy_thresh)
    result = ode(y = C, times = simtimes, func = C_growth, parms = as.data.frame(parms))
    result <- as.data.frame(result)
    colnames(result) <- c('time', 'C')
  # get metrics
    metrics = compute_metrics(result)
  return(metrics)
}

# Pull results from wrapper function into a data frame
allresults <- as.data.frame(sens_C$X) %>%
  pmap(c_wrapper, C = C, simtimes = simtimes, func = C_growth)
```

Plot Results of Sobol Analysis

```r
# extract out results from pmap into a data frame
allres = allresults %>%
  map_dfr(`[`,c("max_c", "mean_c"))

# create boxplots
tmp = allres %>%
  gather(key="metric", value="value")

ggplot(tmp, aes(metric, value, col = metric)) +
  geom_boxplot() +
  theme_light() +
  ggtitle("Metrics for Forest Growth:\nMaximum and Mean Carbon Distributions") +
   xlab("Metric") +
   ylab("Carbon (kgC)") +
   theme(axis.title.x = element_text(color = "black", size = 11),
         axis.text.x = element_text(face = "bold", color = "black", size = 10),
         axis.title.y = element_text(color = "black", size = 11),
         axis.text.y = element_text(face = "bold", color = "black", size = 10),
         plot.title = element_text(color="black", size = 15, face = "bold"),
         panel.border = element_rect(colour = "black", fill = NA, size = 2))
```



Sobol Indicies

```
## compute sobol indicies for max_c
sens_C_maxpop = sensitivity::tell(sens_C, allres$max_c)

# Add row names
rownames(sens_C_maxpop$S) = c("r", "g", "K", "canopy_thresh")

# first-order indices (main effect without co-variance)
sens_C_maxpop$S
```

```
##                 original         bias std. error  min. c.i. max. c.i.
## r             0.28173035 -0.012717831  0.1081818  0.1107716 0.5319759
## g             0.34676770 -0.014699074  0.1159373  0.1348161 0.6121649
## K             0.11330838 -0.007113608  0.1048120 -0.0750912 0.3192425
## canopy_thresh 0.04869604 -0.016323623  0.1138300 -0.1755305 0.2792659
```

```
# this data represents main effect without covariance

## total sensitivity index -note that this partitions the output variance - so values sum to 1

# Add row names
rownames(sens_C_maxpop$T) = c("r", "g", "K", "canopy_thresh")

sens_C_maxpop$T
```

```
##                 original         bias std. error  min. c.i.  max. c.i.
## r             0.29343940 0.0058925679 0.05779138 0.14469391 0.39482604
```

```
## g            0.36689514 0.0076578786 0.06427576 0.22735241 0.48267640
## K            0.17779265 0.0045040632 0.03256342 0.09600037 0.22864917
## canopy_thresh 0.04917463 0.0008846092 0.01033986 0.02716536 0.06475079
```

```
# this data represents the total sensitivity index
```

```
## compute sobol indicies for mean_c
sens_C_meanpop = sensitivity::tell(sens_C, allres$mean_c)

# Add row names
rownames(sens_C_meanpop$S) = c("r", "g", "K", "canopy_thresh")

# first-order indices (main effect without co-variance)
sens_C_meanpop$S
```

```
##                original         bias std. error   min. c.i. max. c.i.
## r             0.7786227 -0.004516055 0.04360059  0.71135534 0.8977755
## g             0.1967882 -0.016929880 0.11958941 -0.05087606 0.4542029
## K             0.1521975 -0.016812252 0.11560674 -0.05538607 0.4066546
## canopy_thresh 0.2081743 -0.017075954 0.10527004  0.01408923 0.4354663
```

```
# this data represents main effect without covariance

## total sensitivity index -note that this partitions the output variance - so values sum to 1

# Add row names
rownames(sens_C_meanpop$T) = c("r", "g", "K", "canopy_thresh")

sens_C_meanpop$T
```

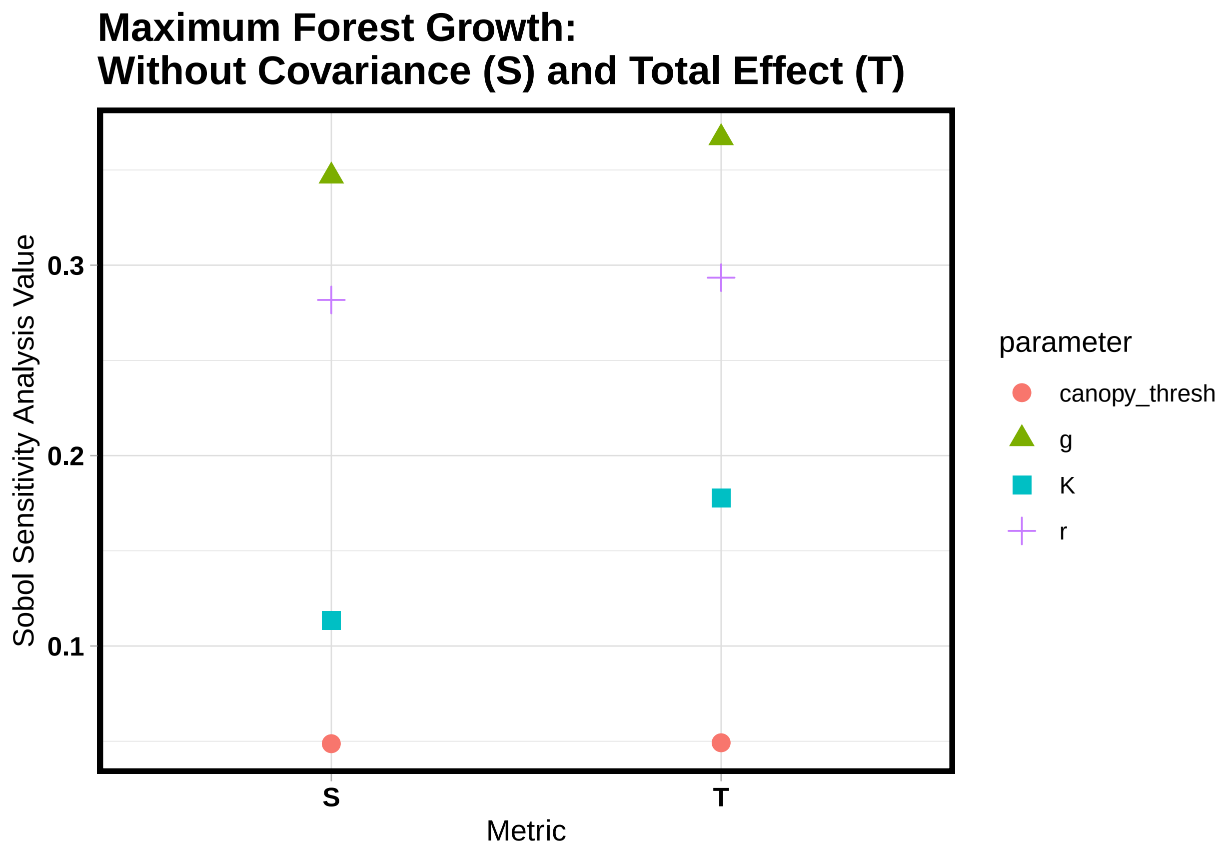
```
##                 original         bias std. error  min. c.i.  max. c.i.
## r             0.66643093 0.0141946133 0.10693039 0.44950755 0.85542128
## g             0.05032166 0.0006238934 0.01092829 0.02355755 0.06779552
## K             0.05912587 0.0014763934 0.01130086 0.03276592 0.07600101
## canopy_thresh 0.09099278 0.0014959153 0.01844440 0.04333103 0.11964053
```

```
# this data represents the total sensitivity index
```

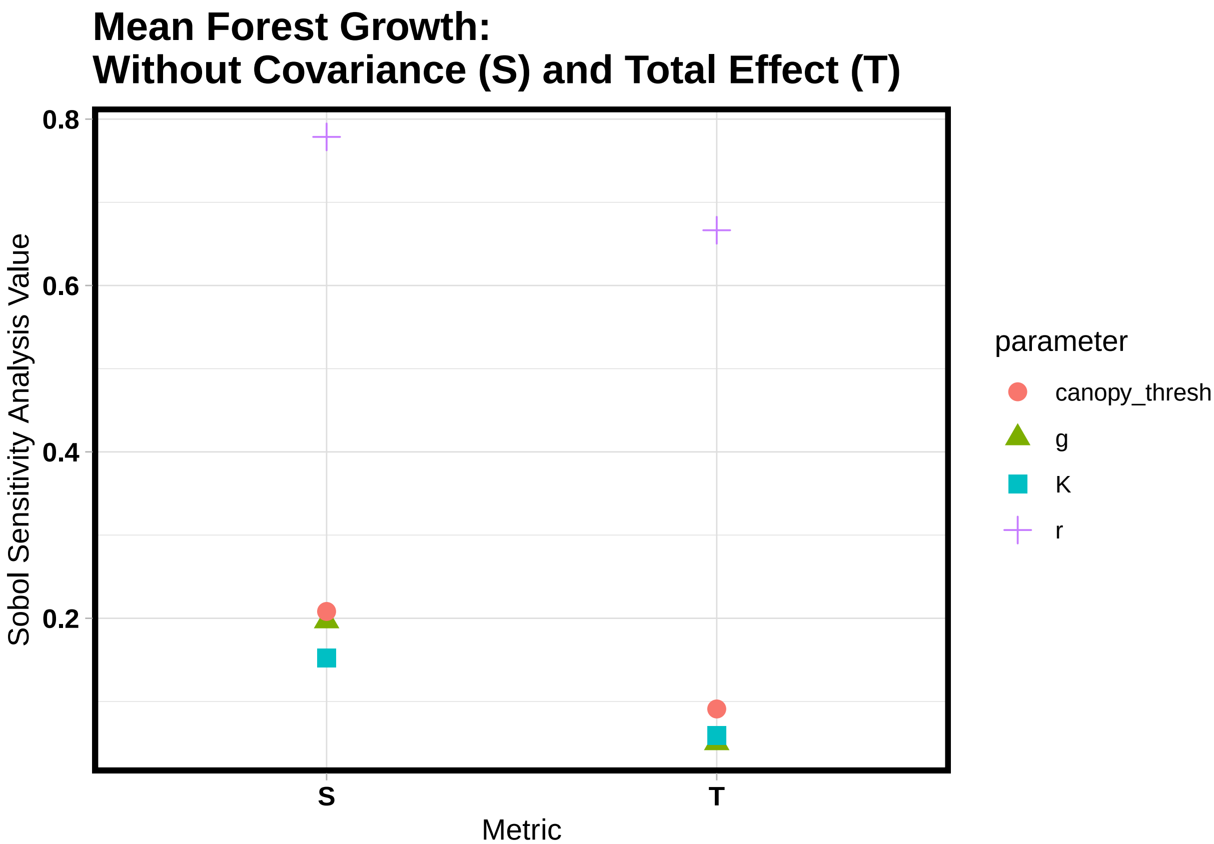
Graph the Results of the Sobol Sensitivity Analysis

```
# max_c
data <- data.frame(S = sens_C_maxpop$S$original, T = sens_C_maxpop$T$original, parameter = c("r", "g",
  pivot_longer(cols = c(S, T))

variables <- rownames(sens_C_maxpop$S)

ggplot(data, aes(x = name, color = parameter)) +
  geom_point(aes(y = value, shape = parameter), size = 3) +
  theme_light() +
  ggtitle("Maximum Forest Growth:\nWithout Covariance (S) and Total Effect (T)") +
   xlab("Metric") +
   ylab("Sobol Sensitivity Analysis Value") +
   theme(axis.title.x = element_text(color = "black", size = 11),
         axis.text.x = element_text(face = "bold", color = "black", size = 10),
         axis.title.y = element_text(color = "black", size = 11),
         axis.text.y = element_text(face = "bold", color = "black", size = 10),
```

```
        plot.title = element_text(color="black", size = 15, face = "bold"),
        panel.border = element_rect(colour = "black", fill = NA, size = 2))
```




```
# mean_c
data <- data.frame(S = sens_C_meanpop$S$original, T = sens_C_meanpop$T$original, parameter = c("r", "g"
  pivot_longer(cols = c(S, T))

variables <- rownames(sens_C_meanpop$S)

ggplot(data, aes(x = name, color = parameter)) +
  geom_point(aes(y = value, shape = parameter), size = 3) +
  theme_light() +
  ggtitle("Mean Forest Growth:\nWithout Covariance (S) and Total Effect (T)") +
   xlab("Metric") +
   ylab("Sobol Sensitivity Analysis Value") +
   theme(axis.title.x = element_text(color = "black", size = 11),
         axis.text.x = element_text(face = "bold", color = "black", size = 10),
         axis.title.y = element_text(color = "black", size = 11),
         axis.text.y = element_text(face = "bold", color = "black", size = 10),
         plot.title = element_text(color="black", size = 15, face = "bold"),
         panel.border = element_rect(colour = "black", fill = NA, size = 2))
```



Concluding Comments

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). (25% - we look for reasonable discussion that uses the results from your analysis and give extra points for discussions that offer particularly creative or insightful commentary)

Climate change might influence the parameters

Submit R markdown with model implementation, graphs and sensitivity analysis and R file with your model (Final 10% for well-organized, clearly documented R markdown)

The results of our simulation show that the most sensitive parameter for maximum forest growth is linear growth rate, and the most sensitive parameter for mean forest growth is exponential growth rate. With climate change, we expect higher temperatures globally, inconsistent precipitation (patterns of drought, massive rain events, or alternation between the two depending on the region), and an influx of carbon in the atmosphere. Since temperature and precipitation both directly influence tree growth, which were the most sensitive parameters, we might see large changes in forest size in less time than has been recorded historically, but whether forest size increases or newer forest size decreases might depend on the environmental preferences of the specific tree species.