Forest Growth Model

Steven Cognac, Charles Hendrickson, & Roshni Katrak-Adefowora

2022-05-12

Background

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

$$dC/dt = r * C$$

for forest where **C is below a threshold canopy closure** and

$$dC/dt = g * (1 = C/K)$$

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 carbon. Think of the canopy closure threshold as the size of the forest at which growth rates change from exponential to linear. Think of r as the early exponential growth rate and g as the linear growth rate once canopy closure has been reached.

C = forest size, units of carbon T = time, years r = early exponential growth rate (before canopy closure has been reached) g = the linear growth rate once canopy closure has been reached K = carrying capacity, units of carbon

canopy closure threshold = the size of the forest at which growth rates change from exponential to linear

Implement Model

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

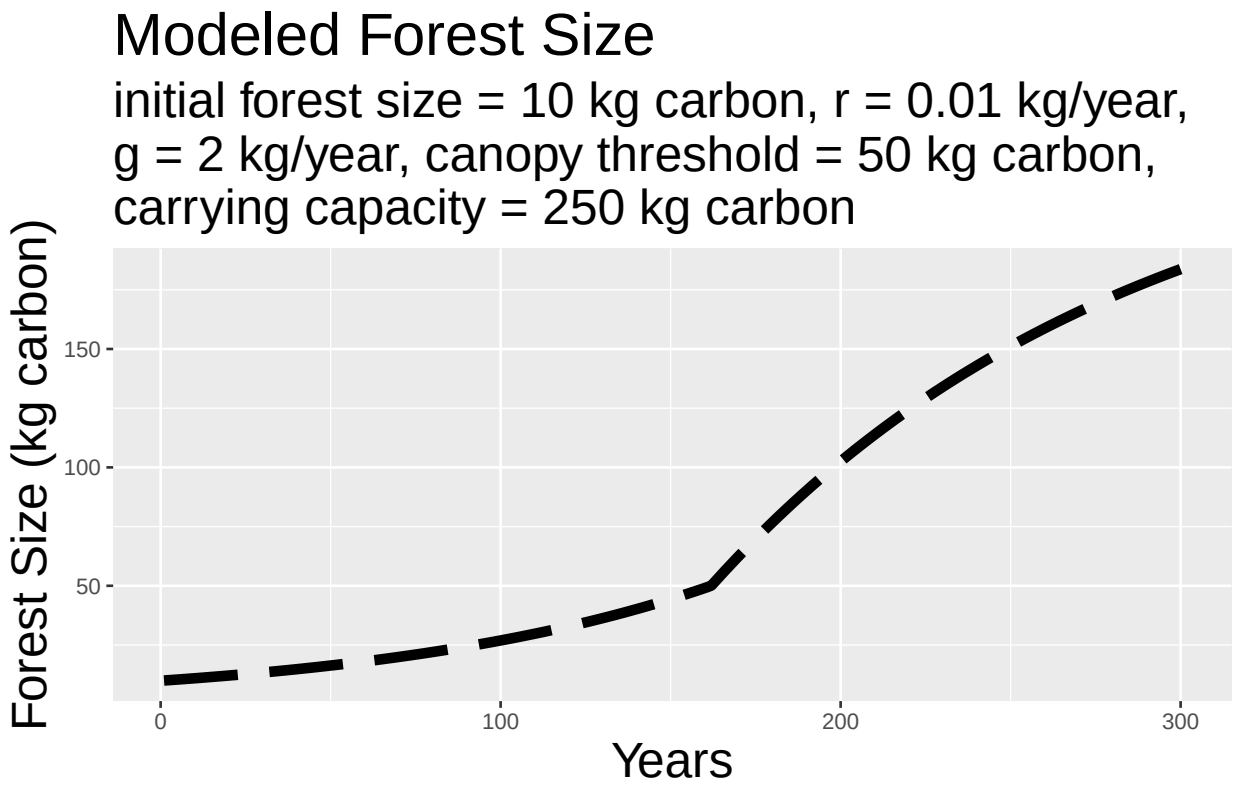
```
## function (time, C, parms)
## {
##     dC = ifelse(C < parms$canopy_threshold, parms$r * C, parms$g *
##         (1 - (C/parms$K)))
##     return(list(dC))
## }
```

Run model over 300 years using the ODE solver (`deSolve` package) starting with initial forest size of 10 kg carbon with the following parameters:

- canopy_threshold = 50 kg (canopy closure threshold)
- 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)

```
years = seq(from = 1, to = 300)

Cinitial = 10

metrics = list(K=250, r=0.01, g=2, canopy_threshold=50)

carbon_growth = ode(y = Cinitial,
                    times = years,
                    func = forest_growth,
                    parms = metrics)

colnames(carbon_growth)=c("year","carbon")
```

```
ggplot(as.data.frame(carbon_growth),aes(year, carbon))+
  geom_line(size = 2, linetype = "longdash") +
  labs(title = "Modeled Forest Size",
       subtitle = paste0("initial forest size = 10 kg carbon, r = 0.01 kg/year, \ng = 2 kg/year, canopy
       x = "Years",
       y = "Forest Size (kg carbon)") +
  theme(title = element_text(size = 20))
```



Run a Sobel sensitivity analysis and explore how the estimated max and mean forest size varies with the pre-canopy closure growth rate (r), post-canopy closure rate (g), canopy closure threshold, and carrying capacity (K).

Assumption is all parameters are normally distributed with a standard deviation of 10% of mean values.

```
# set number of parameters
n = 300

K = rnorm(mean=250, sd=25, n=n)
r = rnorm(mean=0.01, sd=0.001, n=n)
g = rnorm(mean=2, sd=0.2, n=n)
canopy_threshold = rnorm(mean=50, sd=0.1*50, n=n)

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

# repeat to get our second set of samples
K = rnorm(mean=250, sd=25, n=n)
r = rnorm(mean=0.01, sd=0.001, n=n)
g = rnorm(mean=2, sd=0.2, n=n)
canopy_threshold = rnorm(mean=50, sd=0.1*50, n=n)

X2 = cbind.data.frame(r=r, K=K, g=g, canopy_threshold=canopy_threshold)


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

# rename columns
colnames(sens_C$X) = c("r", "K", "g", "canopy_threshold")
head(sens_C$X)
```

```
##                r        K        g canopy_threshold
## [1,] 0.009627197 248.2205 2.138831         49.14991
## [2,] 0.011319123 269.8511 2.114688         55.93123
## [3,] 0.010650115 292.6456 2.014420         46.29443
## [4,] 0.009798438 277.9796 1.990352         55.31325
## [5,] 0.009542424 198.4993 2.174514         55.63171
## [6,] 0.010622571 221.1477 1.836718         49.03989
```

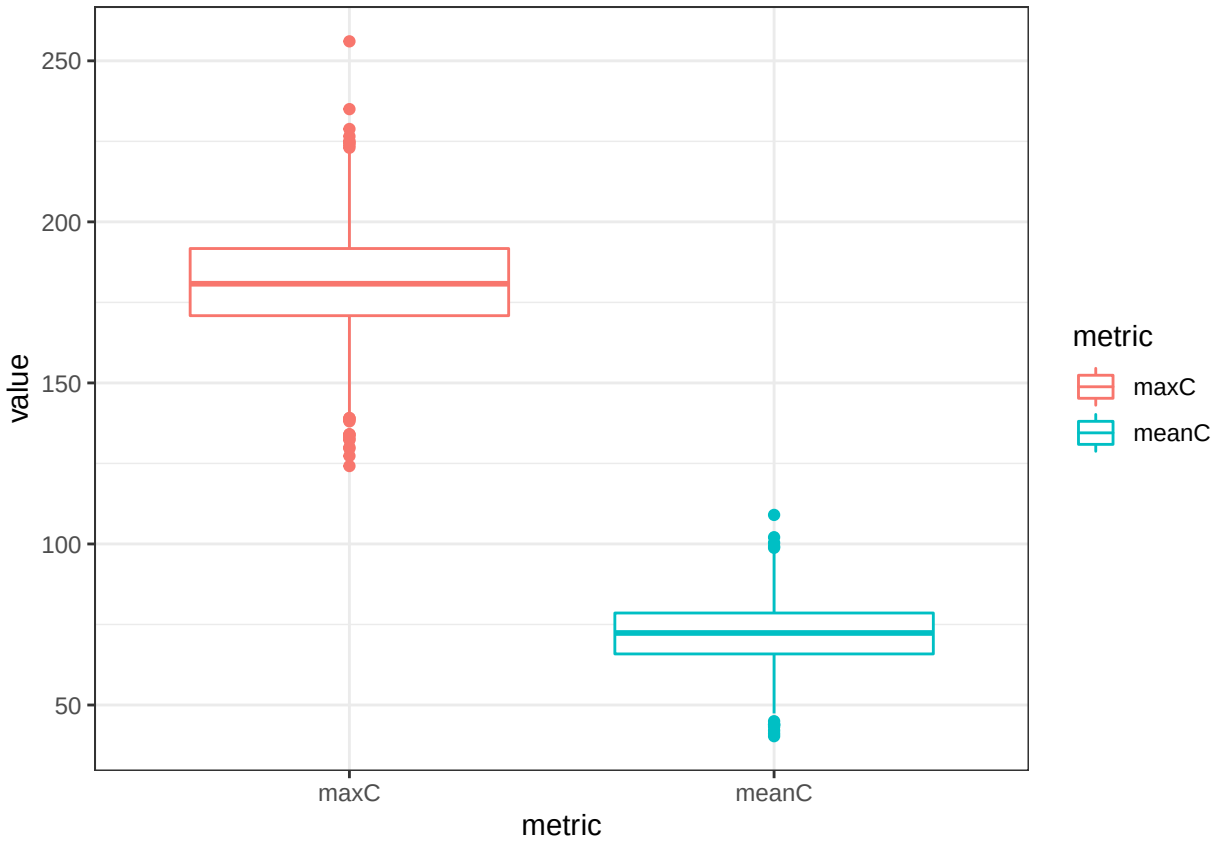
Create two functions to compute metrics. Then, run the ODE solver to compute metrics with wrapper function.

```
# turn computing our metrics into a function
compute_metrics = function(result) {
  maxC = max(result$carbon)
  idx1 = which.max(result$carbon)
  maxyear = result$year[idx1]

  meanC = mean(result$carbon)
```

```
return(list(maxC=maxC,
            maxyear=maxyear,
            meanC=meanC))}

# wrapper function
p_wrapper <- function(r, g, canopy_threshold, K, Cinitial, years, func) {

  parms <- list(r = r, g = g, canopy_threshold = canopy_threshold, K = K)
  result <- ode(y = Cinitial,
                times = years,
                func = func,
                parms = parms,
                method = "lsode")
  colnames(result) <- c("year", "carbon")

  # get metrics
  metrics <- compute_metrics(as.data.frame(result))

  return(metrics)

}
```

```
# now use pmap as we did before
allresults <- as.data.frame(sens_C$X) %>%
  pmap(p_wrapper, Cinitial=Cinitial, years=years, func=forest_growth)

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

```
# create boxplots
tmp <- allres %>%
  pivot_longer(cols = 1:2,
               names_to = "metric",
               values_to = "value")


ggplot(tmp, aes(metric, value, col=metric)) +
  geom_boxplot()+
  theme_bw()
```



Compute the sobol indicies for each metric

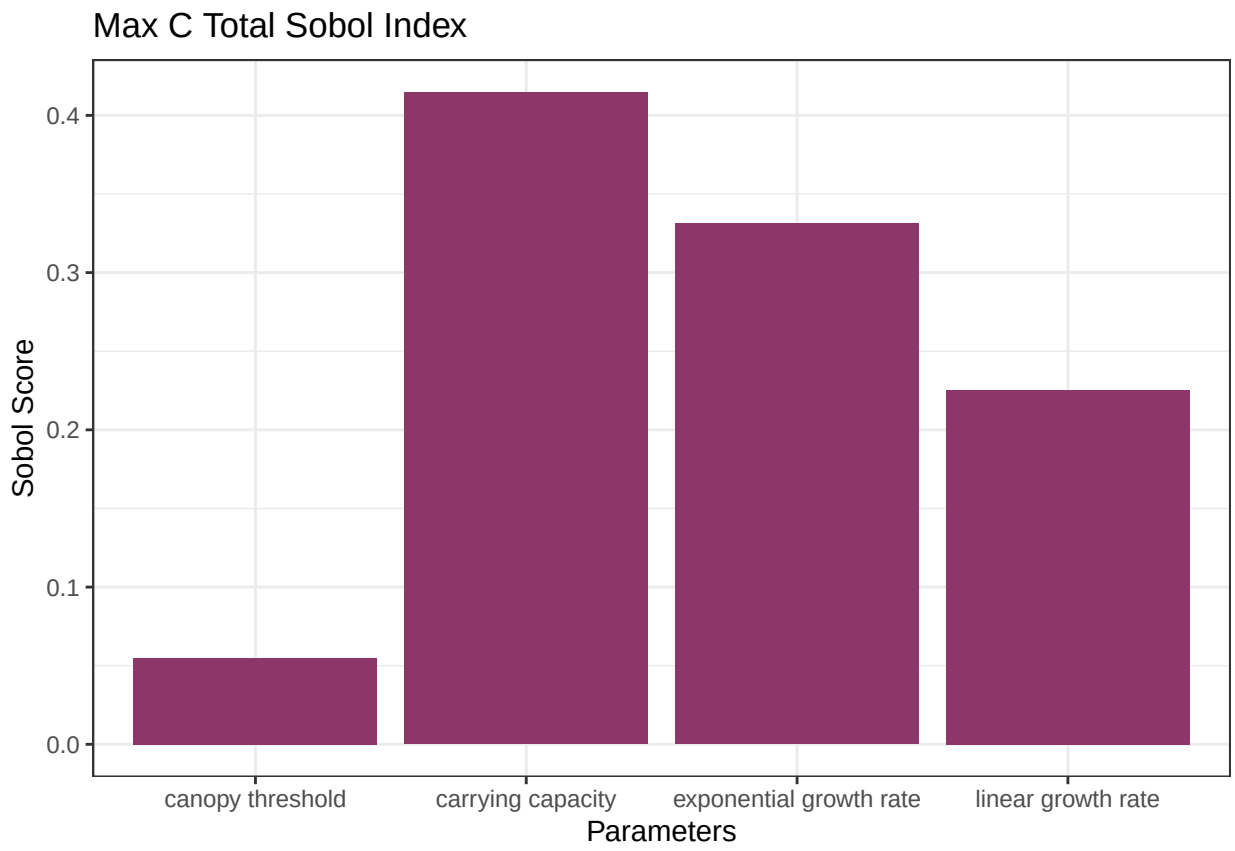
```
# sobol can only handle one output at a time  - so we will need to do them separately

# compute sobol indicies for maxC
sens_C_maxC = sensitivity::tell(sens_C, allres$maxC)

# prepare data frames for plotting maxC first order indices and total indices
# first order indices
sens_C_maxC_S <- sens_C_maxC$S %>%
  rowid_to_column("parameters") %>%
  mutate(parameter = case_when(parameters == "1" ~ "exponential growth rate",
                               parameters == "2" ~ "carrying capacity",
                               parameters == "3" ~ "linear growth rate",
                               parameters == "4" ~ "canopy threshold"))

# total indices
sens_C_maxC_T <- sens_C_maxC$T %>%
  rowid_to_column("parameters") %>%
  mutate(parameter = case_when(parameters == "1" ~ "exponential growth rate",
                               parameters == "2" ~ "carrying capacity",
                               parameters == "3" ~ "linear growth rate",
                               parameters == "4" ~ "canopy threshold"))

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

```r
# prepare data frames for plotting meanC first order indices and total indices
# first order indices
sens_C_meanC_S <- sens_C_meanC$S %>%
  rowid_to_column("parameters") %>%
  mutate(parameter = case_when(parameters == "1" ~ "exponential growth rate",
                               parameters == "2" ~ "carrying capacity",
                               parameters == "3" ~ "linear growth rate",
                               parameters == "4" ~ "canopy threshold"))

# total indices
sens_C_meanC_T <- sens_C_meanC$T %>%
  rowid_to_column("parameters") %>%
  mutate(parameter = case_when(parameters == "1" ~ "exponential growth rate",
                               parameters == "2" ~ "carrying capacity",
                               parameters == "3" ~ "linear growth rate",
                               parameters == "4" ~ "canopy threshold"))
```

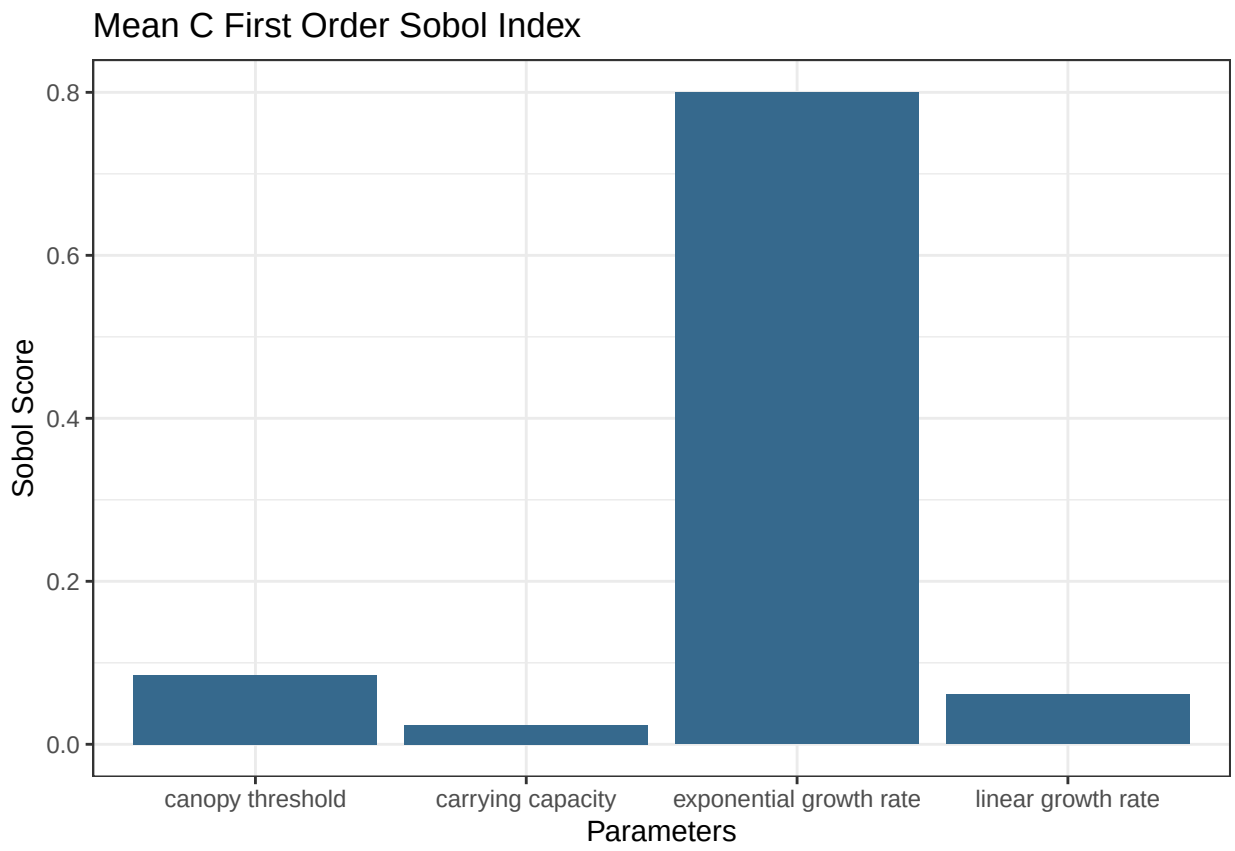
Plot sobol indices

```r
# plot maxC first order index
ggplot(data = sens_C_maxC_S, aes(x = parameter, y = original))+
  geom_col(fill = "#8D366A")+
  theme_bw()+
  labs(title = "Max C First Order Sobol Index",
       x = "Parameters",
       y = "Sobol Score")
```

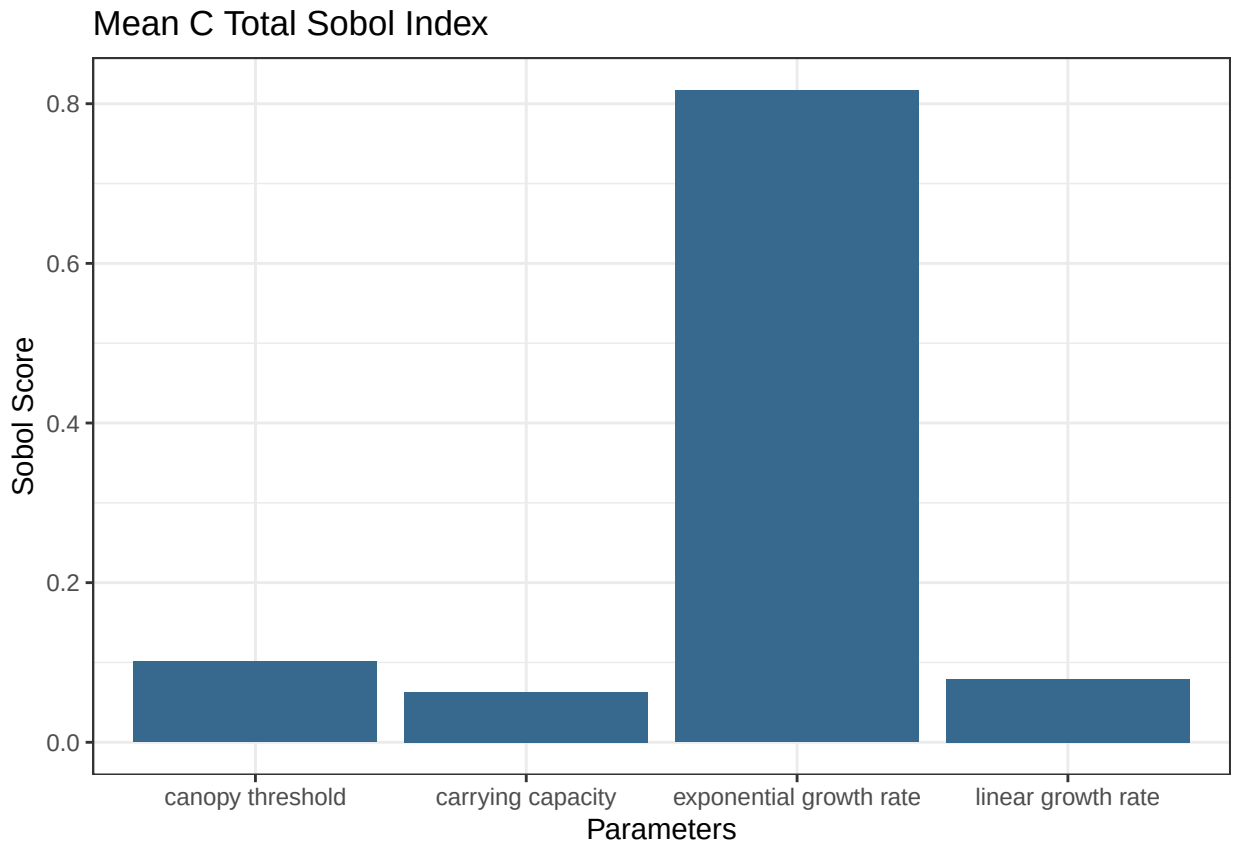

```
# plot maxC total index
ggplot(data = sens_C_maxC_T, aes(x = parameter, y = original))+
  geom_col(fill = "#8D366A")+
  theme_bw()+
  labs(title = "Max C Total Sobol Index",
       x = "Parameters",
       y = "Sobol Score")
```




```
# plot meanC first order index
ggplot(data = sens_C_meanC_S, aes(x = parameter, y = original))+
  geom_col(fill = "#36698D")+
  theme_bw()+
  labs(title = "Mean C First Order Sobol Index",
       x = "Parameters",
       y = "Sobol Score")
```




```
# plot meanC total index
ggplot(data = sens_C_meanC_T, aes(x = parameter, y = original))+
  geom_col(fill = "#36698D")+
  theme_bw()+
  labs(title = "Mean C Total Sobol Index",
       x = "Parameters",
       y = "Sobol Score")
```



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).

For the max forest size, our results indicate that the most influential parameters were carrying capacity (K) and exponential growth rate (r), and for the mean forest size, the most influential parameter was exponential growth rate. Climate change may influence these parameters; for instance, as temperatures rise and drought conditions are exacerbated, trees may experience stunted growth and additional stress, which in turn may lower their carbon carrying capacities.