Forest Growth Model

AUTHOR
Tom Gibbens-Matsuyama

Task

1. Implement the model in R as a differential equation

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

- canopy closure threshold of 50 kgC
- K = 250 kg C
- r = 0.01
- g = 2 kg/year

3. Graph the results

4. Run a sobol global sensitivity analysis that explores how the estimated maximum forest size (i.e. max of C 300 years, vaires with these parameters)

- pre canopy closure growth rate (r)
- post-canopy closure growth rate (g) canopy closure threshold and carrying capacity (K)

Assume that paramters are all normally distributed with means as given above and std deviation of 10% of mean value

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

6. In 2-3 sentences, discuss what the results mean. How do the parameters influence climate change.

Load libraries

```
library(tidyverse)
library(deSolve)
library(sensitivity)
library(here)
```

1. Implement model

Model was created in the forest_growth.R file.

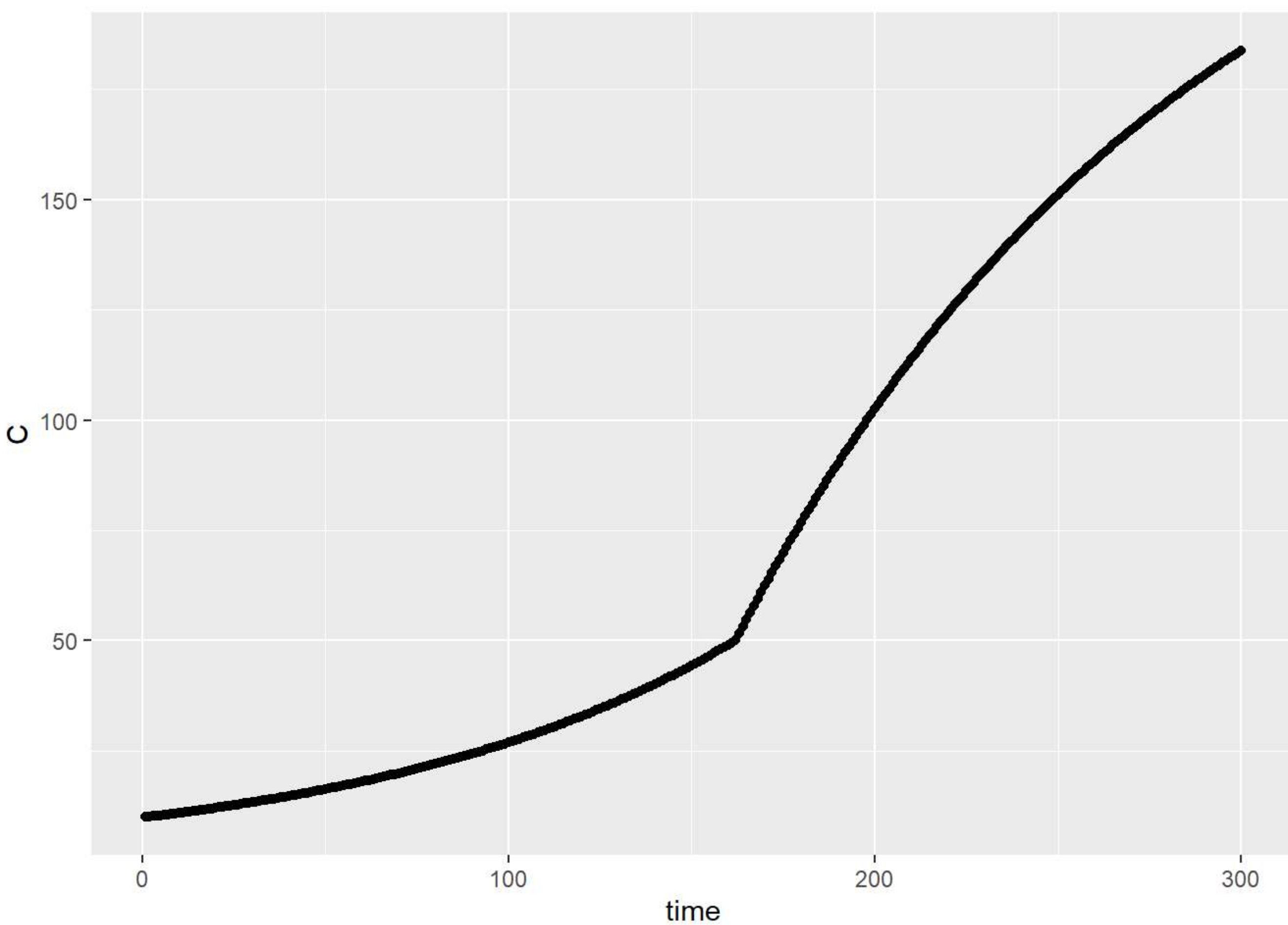
2. Run the model with the parameters above

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

```r
# We know initial Forest size C
Cinitial <- 10

simtimes <- seq(from = 1, to = 300)

parms <- list(r = 0.01, K = 250, g = 2, thresh = 50)

results <- ode(y = Cinitial, times = simtimes, func = forest_growth, parms = parms)

colnames(results) <- c("time", "C")
```

3. Graph the results

```r
results <- as.data.frame(results)
ggplot(results, aes(time, C)) +
  geom_point()
```



4. Run a sobol sensitivity analysis

We will vary the parameters, r, g, and K, assuming a normal distribution with a 10% standard deviation

```r
# Set sample size
np <- 1000

# Generate normal distributions for parameters
K <- rnorm(mean = 250, sd = 25, n = np)
r <- rnorm(mean = 0.01, sd = 0.001, n = np)
g <- rnorm(mean = 2, sd = 0.2, n = np)
thresh <- 50
X1 <- cbind.data.frame(r = r, K = K, g = g, thresh = thresh)

# Generate second set of samples
K <- rnorm(mean = 250, sd = 25, n = np)
r <- rnorm(mean = 0.01, sd = 0.001, n = np)
g <- rnorm(mean = 2, sd = 0.2, n = np)
thresh <- 50
X2 <- cbind.data.frame(r = r, K = K, g = g, thresh = thresh)

# fix any negative values and they are not meaningful
X1 <- X1 %>% map_df(pmax, 0.0)
X2 <- X2 %>% map_df(pmax, 0.0)

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

# View head
head(sens_C$X)
```

```
            [,1]     [,2]     [,3] [,4]
[1,] 0.009875211 215.6890 1.738333   50
[2,] 0.009011438 221.9325 1.647243   50
[3,] 0.012364407 267.1421 2.263124   50
[4,] 0.010845089 218.6934 1.426489   50
[5,] 0.009454891 265.3010 1.912286   50
[6,] 0.011415218 281.8210 1.737262   50
```

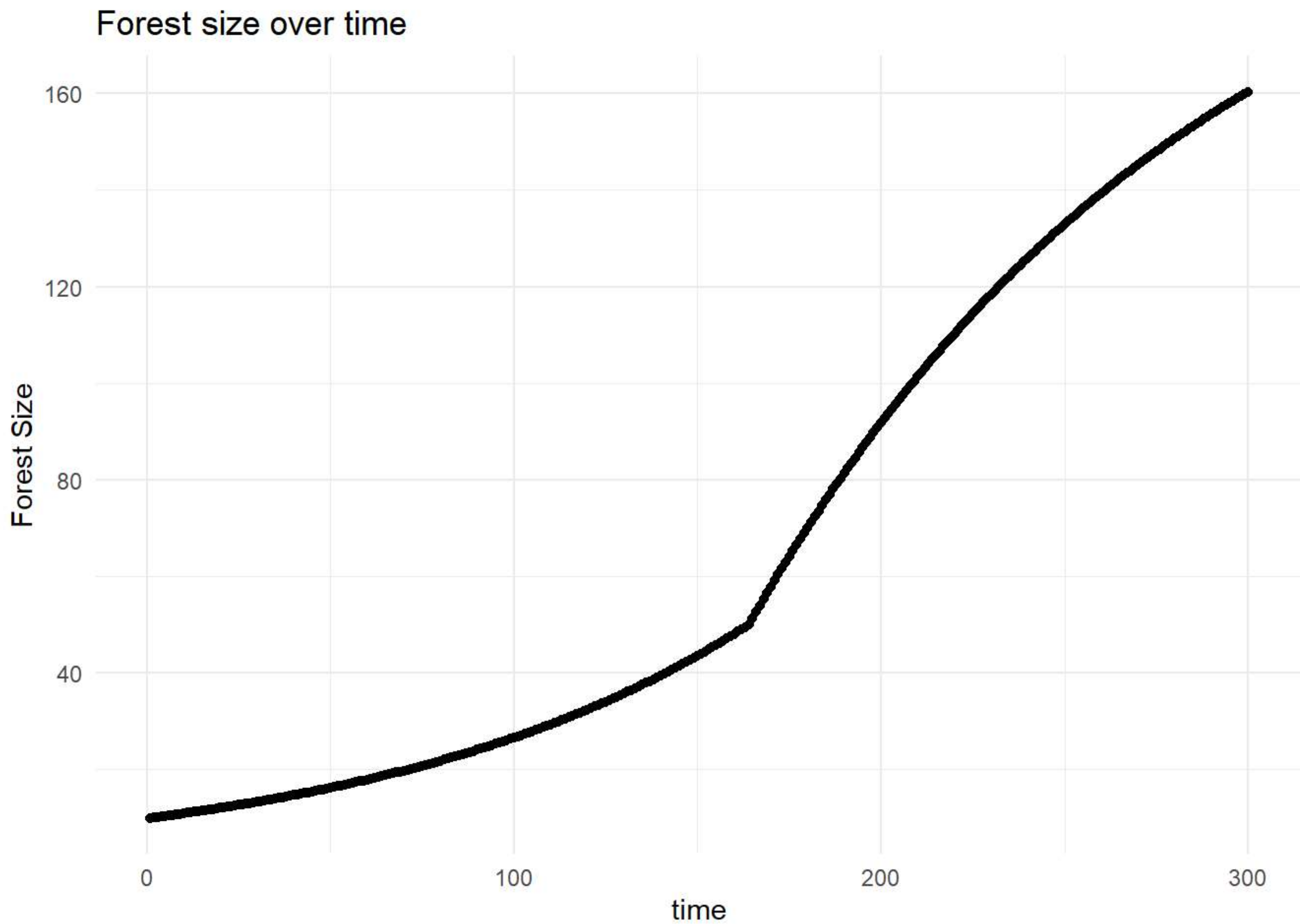
```r
# lets add names
colnames(sens_C$X) <- c("r", "K", "g", "thresh")

# View head
head(sens_C$X)
```

```
               r        K        g thresh
[1,] 0.009875211 215.6890 1.738333     50
[2,] 0.009011438 221.9325 1.647243     50
[3,] 0.012364407 267.1421 2.263124     50
[4,] 0.010845089 218.6934 1.426489     50
[5,] 0.009454891 265.3010 1.912286     50
[6,] 0.011415218 281.8210 1.737262     50
```

```
# gets results for 200 years (evaluating every year)
simtimes <- seq(from = 1, to = 300)
parms <- list(r = sens_C$X[1,"r"], K = sens_C$X[1,"K"], g = sens_C$X[1,"g"], thresh = sens_C$X[1,

result <- ode(y = Cinitial, times = simtimes, func = forest_growth, parms = parms)

head(result)
```

```
     time        1
[1,]    1 10.00000
[2,]    2 10.09924
[3,]    3 10.19947
[4,]    4 10.30069
[5,]    5 10.40292
[6,]    6 10.50616
```

```
colnames(result) <- c("time", "C")
# turn it into a data frame
result <- as.data.frame(result)
ggplot(result, aes(time, C)) +
  geom_point() +
  labs(x = "time",
       y = "Forest Size",
       title = "Forest size over time") +
  theme_minimal()
```



This is still creating a graph for just one simulation as we indexed for the first row in our data. Let's create a max size function and a wrapper function that can be used for all parameters.

Compute metrics and wrapper function

```
# Create max size function
max_c <- function(result){

  max_c <- max(result$C)

  return(list(max_c))

}

# Wrapper function
wrapper <- function(K, g, r, thresh, Cinitial, simtimes, forest_growth_func, max_c_func) {
    parms <- list(r = r, K = K, g = g, thresh = thresh)
    result <- ode(y = Cinitial, times = simtimes, func = forest_growth_func, parms = parms, metho
    colnames(result) <- c("time", "C")

    # get metrics
    metrics <- max_c_func(as.data.frame(result))
    return(metrics)
}
```

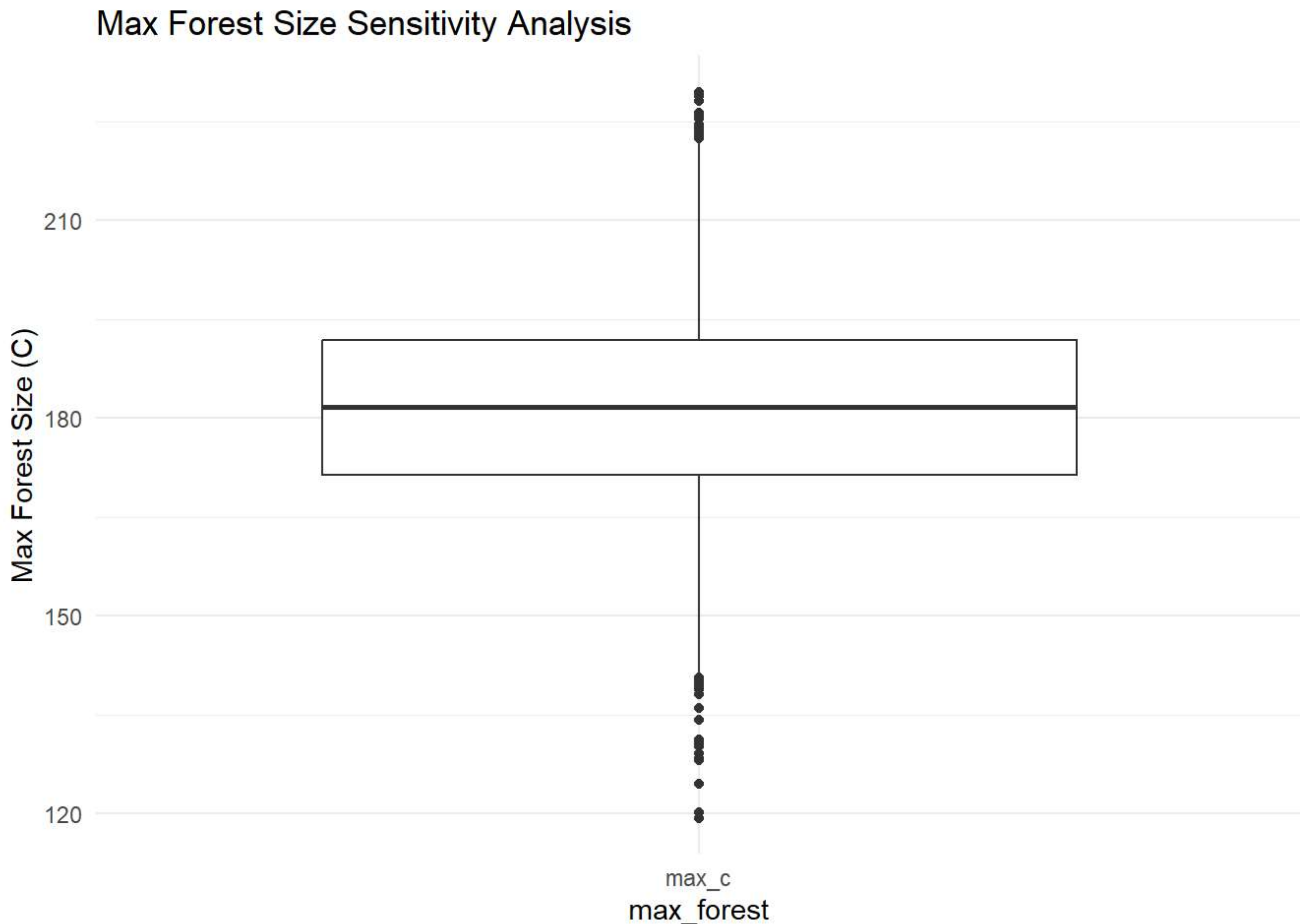
```
# test
wrapper(
  r = 0.01, K = 250, Cinitial = 10, simtimes = seq(from = 1, to = 300),
  forest_growth_func = forest_growth, max_c_func = max_c, g = 2, thresh = 50
)
```

```
[[1]]
[1] 183.4948
```

5. Graph the results

Run the wrapper for all parameters and look at results

```
# now use pmap as we did before

allresults <- as.data.frame(sens_C$X) %>%
  pmap(wrapper, Cinitial = Cinitial,
       simtimes = simtimes,
       forest_growth_func = forest_growth,
       max_c_func = max_c)

# extract out results from pmap into a data frame
allres <- allresults %>% map_dfr(~ tibble(max_c = .x[[1]]))


# create boxplots
tmp <- allres %>% pivot_longer(cols = everything(), names_to = "max_forest", values_to = "value")
ggplot(tmp, aes(max_forest, value)) +
  geom_boxplot() +
  labs(title = "Max Forest Size Sensitivity Analysis",
       y = "Max Forest Size (C)") +
  scale_y_continuous() +
  theme_minimal()
```



Compute the sobol indicies for each metric

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

sens_C_maxsize <- sensitivity::tell(sens_C, allres$max_c)

# first-order indices (main effect without co-variance)
rownames(sens_C_maxsize$S) <- c("r", "K", "g", "thresh")
sens_C_maxsize$S
```

```
          original         bias std. error   min. c.i.  max. c.i.
r       0.38180089 -0.001768268 0.03009359  0.31928172 0.44046552
K       0.37815918 -0.001027182 0.02832577  0.32633449 0.44029208
g       0.22131061 -0.003606954 0.03158504  0.16819327 0.28874054
thresh -0.02682503 -0.002918599 0.03134490 -0.08733245 0.04164784
```

```
# total sensitivity index -note that this partitions the output variance
rownames(sens_C_maxsize$T) <- c("r", "K", "g", "thresh")
sens_C_maxsize$T
```

```
           original          bias   std. error     min. c.i.     max. c.i.
r       4.321177e-01  3.205012e-04 2.202564e-02  3.867949e-01  4.740860e-01
```

```
K       4.127815e-01  2.039456e-03 2.400454e-02  3.591555e-01  4.540908e-01
g       2.466799e-01 -2.174528e-04 1.365622e-02  2.184908e-01  2.770175e-01
thresh -2.036149e-13  2.498487e-13 1.967431e-13 -8.548258e-13 -9.635145e-14
```

6. Conclusion

From the sensitivity analysis, we see that r (exponential growth rate) and K (carrying capacity) have the most influence on max forest size. Climate change can have a direct impact on both growth rates. For example, increases in drought may cause a decrease in growth rate due to less water availability.