Assignment 7: Growth

Bri Baker, Ian Brunjes, & Scout Leonard

05/13/2022

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,)

```r
#source function
source(here("R", "dgrowth.R"))

#view function:
dgrowth
```

```
## function (Time, C, parms)
## {
##     if (C < parms$thresh) {
##         rate = parms$r * C
##     }
##     else {
##         rate = parms$g * (1 - (C/parms$K))
##     }
##     return(list(rate))
## }
```

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

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)

```r
tm = seq(1, 300, 1) # model run time: 300 years

Cinit = 10 # initial kilograms of Carbon

gps = list(
  K = 250, # carrying capacityL kg of Carbon
  r = 0.01, # exponential rate of growth before canopy closure
  g = 2, # linear growth after canopy closure kilograms/year
  thresh = 50 # canopy closure threshold (kilograms of carbon)
)

#results
res = ode(Cinit, tm, dgrowth, parms = gps)

#rename results columns
colnames(res)=c("time","carbon")
```

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

```r
ggplot(as.data.frame(res), aes(time, carbon)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Growth Model: 300 Years",
       x = "Time (years)",
       y = "Carbon (kilograms)") +
  theme(plot.title = element_text(hjust = 0.5))
```

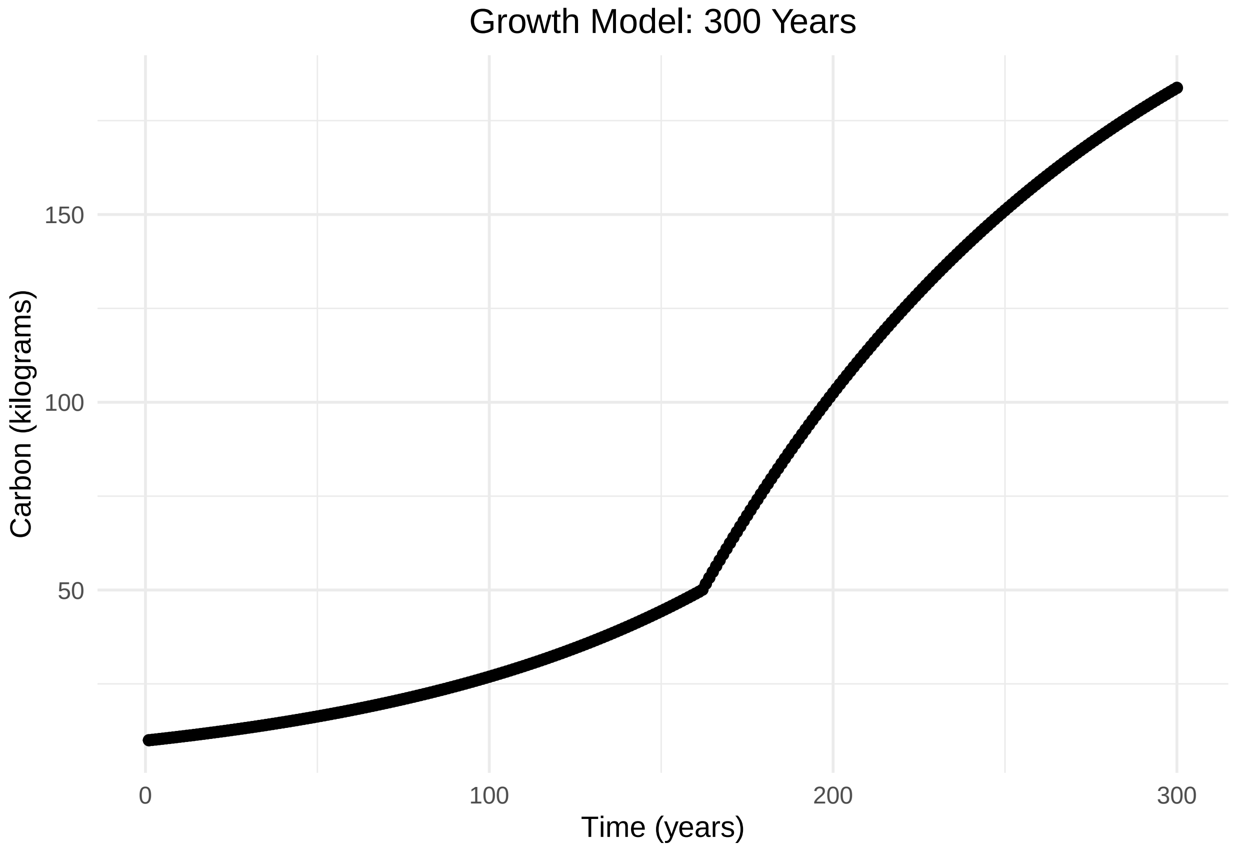


Figure 1: Growth model over 300 years using ODE Solver, starting with initial forest size of 10 kilograms of carbon and the following parameters: 250 kilograms of carbon carrying capacity, 0.01 exponential growth rate before canopy closure, and 2 kilograms per year growth rate after canopy closure.

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)

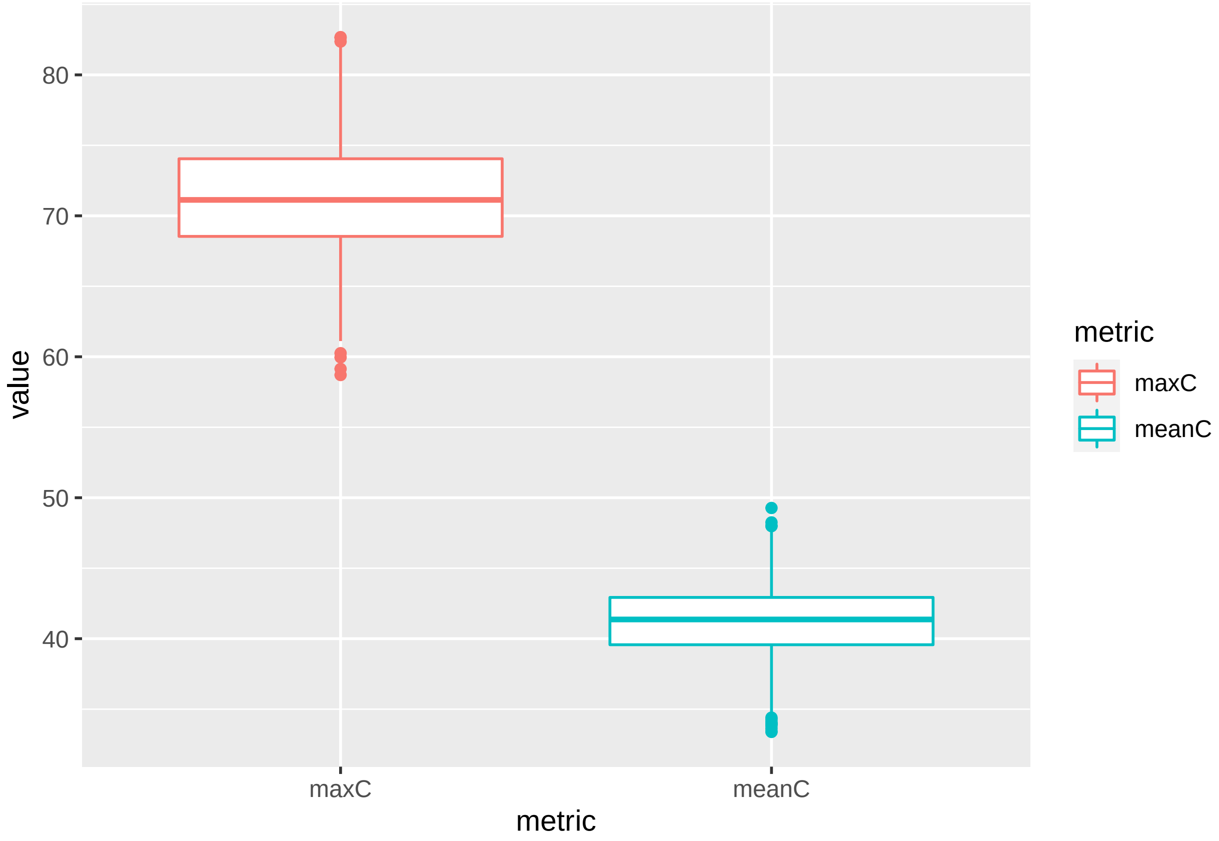
Assume that parameters are all normally distributed with means as given above and standard deviation of 10% of mean value

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)

```r
# Mean parameter values
u_K = 250
u_thresh = 50
u_r = 0.01
u_g = 0.2

# Standard deviation factor
sdf = 0.1

# Build two sample parameter sets
Cinit = 10
np = 100

X1 = cbind.data.frame(
  K = rnorm(mean = u_K, sd = u_K * sdf, n = np),
  thresh = rnorm(mean = u_thresh, sd = u_thresh * sdf, n = np),
  r = rnorm(mean = u_r, sd = u_r * sdf, n = np),
  g = rnorm(mean = u_g, sd = u_g * sdf, n = np)
)

X2 = cbind.data.frame(
  K = rnorm(mean = u_K, sd = u_K * sdf, n = np),
  thresh = rnorm(mean = u_thresh, sd = u_thresh * sdf, n = np),
  r = rnorm(mean = u_r, sd = u_r * sdf, n = np),
  g = rnorm(mean = u_g, sd = u_g * sdf, n = np)
)

# Create sobol object with parameter sets
sens_carbon = sobolSalt(model = NULL,X1, X2, nboot = 300)
colnames(sens_carbon$X) = c("K","thresh", "r", "g")

parms = sens_carbon$X
```

```r
# Run our diffEQ model, extracting max and mean forest carbon

# Function to compute relevant metrics
compute_metrics = function(result) {
  maxC = max(result$carbon)
  meanC = mean(result$carbon)

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

# Wrapper function for running solver and getting metrics for all params
C_wrapper = function(K, thresh, r, g, Cinit, simtimes, func) {
  gps = list(
    K = K,
    r = r,
    g = g,
    thresh = thresh
  )

  result = ode(Cinit, simtimes, func, gps, method = "daspk")
  result = as.data.frame(result)
  colnames(result) = c("time","carbon")

  metrics = compute_metrics(result)
  return(metrics)
}

# Use pmap with wrapper function to execute against sobol parameters
allresults = as.data.frame(parms) %>%
  pmap(C_wrapper, Cinit = Cinit, simtimes = tm, func = dgrowth)

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

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



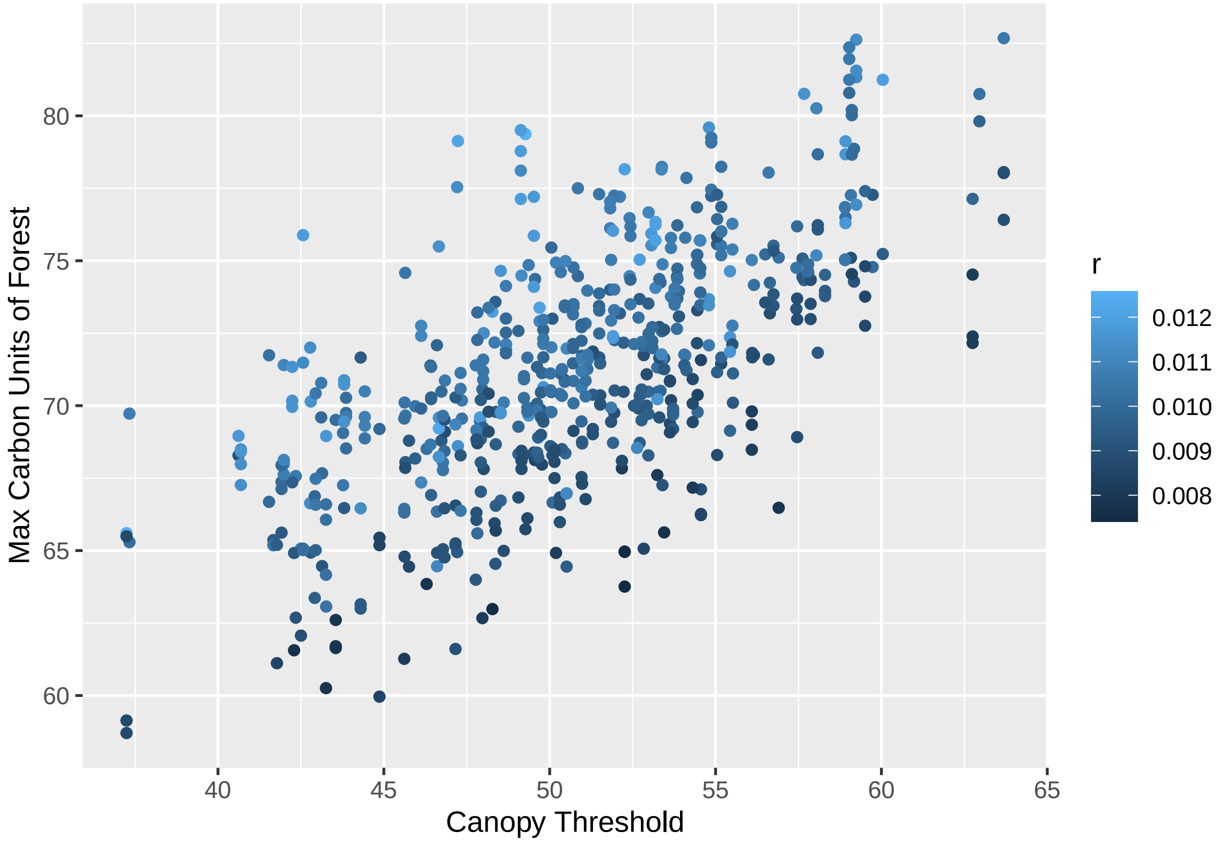
```
# Compute sobol indices for each metric
# Max carbon
sens_carbon_max = sensitivity::tell(sens_carbon, allres$maxC)
row.names(sens_carbon_max$S) = colnames(parms)
row.names(sens_carbon_max$T) = colnames(parms)

print(sens_carbon_max)
```

```
##
## Call:
## sobolSalt(model = NULL, X1 = X1, X2 = X2, nboot = 300)
##
## Model runs: 600
##
## Model variance: 18.84666
##
## First order indices:
##          original         bias std. error    min. c.i.  max. c.i.
## K      -0.1511900  0.007842517 0.10739262 -0.388034152 0.04863219
## thresh  0.3852239 -0.004480350 0.08761816  0.223371775 0.55495556
## r       0.1747168 -0.000673109 0.10226042 -0.007226004 0.38491033
## g       0.1603332 -0.003807190 0.09887242 -0.023647728 0.35757657
##
## Total indices:
##          original         bias  std. error   min. c.i.  max. c.i.
## K      0.01810515 0.0007177633 0.004233697 0.008067957 0.02470776
## thresh 0.56612007 0.0138030834 0.078029908 0.382362042 0.69562118
## r      0.31770645 0.0055632594 0.061040761 0.180301255 0.41300516
## g      0.29044219 0.0031761415 0.052529407 0.178526026 0.37789808
```

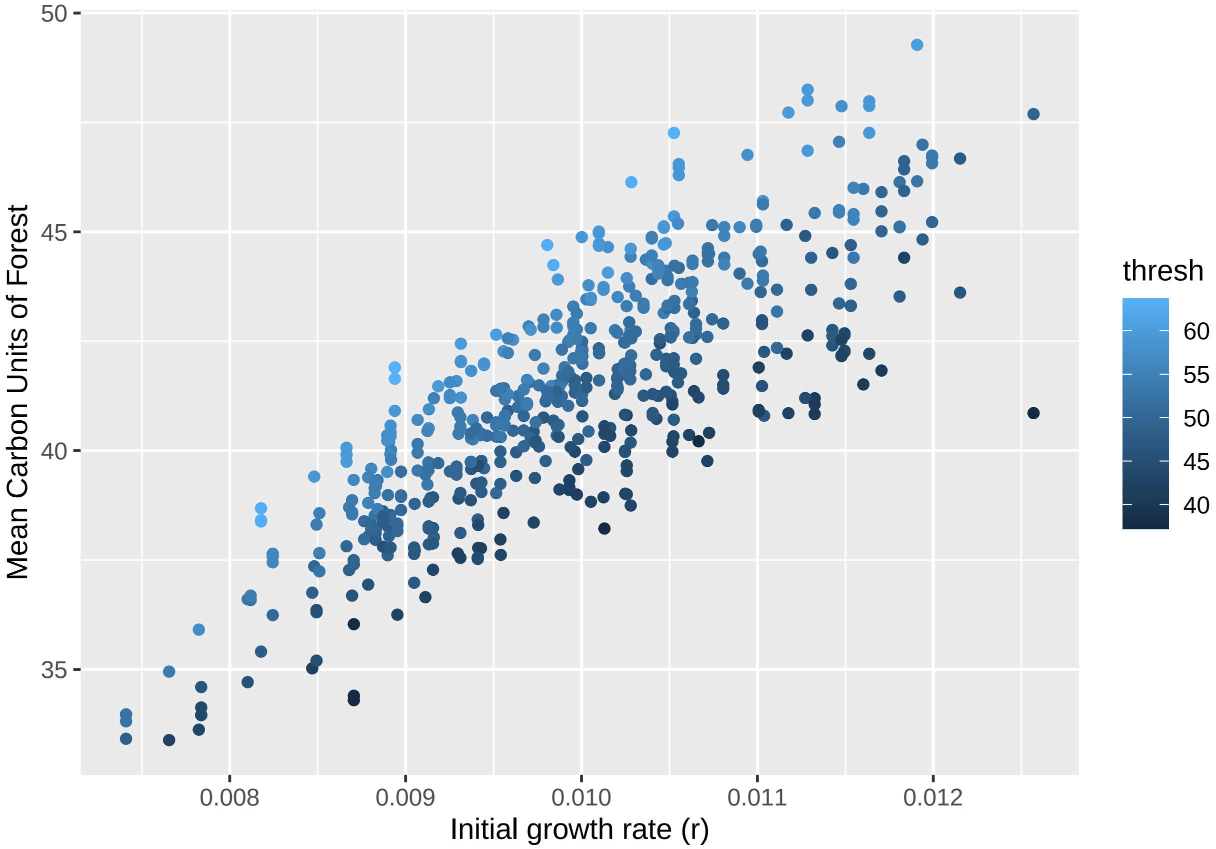
```
# Mean carbon
sens_carbon_mean = sensitivity::tell(sens_carbon, allres$meanC)
row.names(sens_carbon_mean$S) = colnames(parms)
row.names(sens_carbon_mean$T) = colnames(parms)
print(sens_carbon_mean)
```

```
##
## Call:
## sobolSalt(model = NULL, X1 = X1, X2 = X2, nboot = 300)
##
## Model runs: 600
##
## Model variance: 7.903904
##
## First order indices:
##           original         bias std. error   min. c.i.  max. c.i.
## K      -0.14259708 -0.007123867 0.10295104 -0.34584566 0.06363506
## thresh  0.16039855 -0.009432273 0.10600820 -0.03923007 0.38412063
## r       0.65416647 -0.008146150 0.05871449  0.56597729 0.79911356
## g      -0.07086221 -0.006719425 0.10414365 -0.26496506 0.15089544
##
## Total indices:
##           original         bias   std. error    min. c.i.   max. c.i.
## K      0.002001729 7.176107e-05 0.0004605686 0.0008942342 0.002713545
## thresh 0.309307301 8.524317e-03 0.0541848429 0.1777886378 0.390467243
## r       0.820383644 6.940854e-03 0.1017479882 0.6103168142 1.005164690
## g       0.038119318 1.247763e-03 0.0082577340 0.0180666322 0.049431993
```

```
# More plotting
# Make data frame for plotting
results_df = cbind.data.frame(
  parms,
  mean = sens_carbon_mean$y,
  max = sens_carbon_max$y
)

# For max values, threshold then r have most impact
ggplot(results_df, aes(x = thresh, y = max, col = r))+geom_point()+labs(y="Max Carbon Units of Forest",
```



```
# For mean values, r then threshold have most impact
ggplot(results_df, aes(x = r, y = mean, col = thresh))+geom_point()+labs(y="Mean Carbon Units of Forest
```



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)

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)