```
#### setup ####

# clear environment
rm(list=ls())

# load packages
library(tidyverse)
library(deSolve)

# figure settings
fig_theme <- theme_bw() +
  theme(axis.text = element_text(size = 10, color="black"),
        axis.title = element_text(size = 12),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 12),
        legend.box.margin = margin(-10, -10, -10, -10))


#### equations ####

# seeds
# S[t+1] = s(1-g)S[t] + c(y_J*J[t] + y_A*A[t])/(1 + alpha(y_J*J[t] +
y_A*A[t]))
# seeds produced this year = seeds that didn't germinate and survived
in the seed bank + seed produced this year, which depends on the size
of adults and juveniles and intraspecific competition

# juveniles
# J[t+1] = gS[t] + (1-m)J[t]
# juveniles this year = seeds that germinated and survived + juveniles
that didn't mature

# adults
# A[t+1] = mJ[t] + kA[t]
# adults this year = juveniles that matured and adults that survived
from last year


#### parameters ####

# initial conditions
S0 <- 100   # initial seed bank
J0 <- 0   # initial juvenile trees
A0 <- 0   # initial adult trees

# simulation conditions
simtime <- 50  # years of simulation
```

```
# plant traits (all per year)
s <- 0.8   # survival in the seed bank (fake value)
g <- 0.4   # proportion of seeds that germinate (fake value)
c <- 39000 / 1000   # seeds per gram (https://www.cabi.org/isc/
datasheet/18119)
alpha <- 0.2   # reduction in seed production based on plant biomass
(fake value)
m <- 0.8   # juvenile survival and maturation (fake value)
k <- 0.9   # adult survival (fake value)
q <- 4   # conversion from juvenile to adult biomass (fake value)

# biomass-light parameters
max_light <- 1800   # max light value (Drew's figure)
b <- 2   # max biomass size in grams (Drew's figure)
a <- b / max_light   # linear coefficient (Drew's figure)
h1 <- 400   # half saturation constant for saturating curve (Drew's
figure)
h2 <- 800   # half saturation constant for S-curve (fake value)


#### biomass-light functions ####

# linear function
bl_lin <- function(light){

  a <- b / max_light   # linear coefficient (Drew's figure)

  biomass <- light * a

  return(biomass)

}

# saturating function
bl_sat <- function(light){

  b1 <- b + (b * (h1 + max_light) / max_light - b)   # max biomass for
saturating curve (to match linear)

  biomass <- (b1 * light / (h1 + light))

  return(biomass)

}

# S-curve
bl_scu <- function(light){

  b2 <- b + (b * (h2^2 + max_light^2) / max_light^2 - b)   # max
```

```
biomass for S-curve (to match linear)

  biomass <- (b2 * light^2 / (h2^2 + light^2))

  return(biomass)

}

# dataframe of test functions
bl_test <- tibble(light_val = rep(seq(0, max_light, length.out = 200),
3),
                  fun_shape = rep(c("linear", "saturating", "S-
curve"), each = 200)) %>%
  mutate(biomass = case_when(fun_shape == "linear" ~
bl_lin(light_val),
                             fun_shape == "saturating" ~
bl_sat(light_val),
                             fun_shape == "S-curve" ~
bl_scu(light_val)),
         fun_shape = fct_relevel(fun_shape, "linear", "saturating"))

# figure of function test
pdf("output/light_biomass_function_shapes.pdf")
ggplot(bl_test, aes(light_val, biomass, color = fun_shape)) +
  geom_line() +
  geom_vline(xintercept = 10, linetype = "dashed", color = "black") +
  geom_vline(xintercept = 250, linetype = "dashed", color = "black") +
  xlab(expression(paste("Light availability (", mu, "moles ", m^-2, "
", sec^-1, ")", sep = ""))) +
  ylab("Biomass (g)") +
  guides(color = guide_legend(title = "Function shape")) +
  fig_theme +
  theme(legend.position = c(0.8, 0.2))
dev.off()


#### simulation function ####

sim_fun <- function(light, fun_shape){

  # juvenile biomass as a function of light
  y_J <- ifelse(fun_shape == "linear", bl_lin(light), ifelse(fun_shape
== "saturating", bl_sat(light), bl_scu(light))) * 3 # assume this was
over one month and increase for longer growing season

  # adult biomass
  y_A <- q * y_J

  # initialize populations
  S <- rep(NA,simtime)
```

```
  J <- rep(NA,simtime)
  A <- rep(NA,simtime)

  S[1] <- S0
  J[1] <- J0
  A[1] <- A0

  # simulate population dynamics
  for(t in 1:(simtime - 1)){

    # population size
    S[t+1] = s * (1 - g) * S[t] + c * (y_J * J[t] + y_A * A[t]) / (1 +
alpha * (y_J * J[t] + y_A * A[t]))
    J[t+1] = g * S[t] + (1 - m) * J[t]
    A[t+1] = m * J[t] + k * A[t]

    # correct to prevent negative numbers
    S[t+1] = ifelse(S[t+1] < 1, 0, S[t+1])
    J[t+1] = ifelse(J[t+1] < 1, 0, J[t+1])
    A[t+1] = ifelse(A[t+1] < 1, 0, A[t+1])
  }

  # save data
  dfN = tibble(time = 1:simtime, seeds = S, juveniles = J, adults = A)
%>%
    mutate(biomass = juveniles * y_J + adults * y_A,
           trees = juveniles + adults)

  # return
  return(dfN)
}


#### test a few scenarios ####

# light is at 10 and biomass growth is a linear function of light
test_10_lin <- sim_fun(10, "linear") %>%
  as_tibble() %>%
  mutate(Light = 10,
         fun_shape = "linear")

# light is at 10 and biomass growth is a saturating function of light
test_10_sat <- sim_fun(10, "saturating") %>%
  as_tibble() %>%
  mutate(Light = 10,
         fun_shape = "saturating")

# light is at 10 and biomass growth is an S-curve function of light
test_10_scu <- sim_fun(10, "S-curve") %>%
  as_tibble() %>%
```

```
  mutate(Light = 10,
         fun_shape = "S-curve")

# light is at 250 and biomass growth is a linear function of light
test_250_lin <- sim_fun(250, "linear") %>%
  as_tibble() %>%
  mutate(Light = 250,
         fun_shape = "linear")

# light is at 250 and biomass growth is a saturating function of light
test_250_sat <- sim_fun(250, "saturating") %>%
  as_tibble() %>%
  mutate(Light = 250,
         fun_shape = "saturating")

# light is at 250 and biomass growth is an S-curve function of light
test_250_scu <- sim_fun(250, "S-curve") %>%
  as_tibble() %>%
  mutate(Light = 250,
         fun_shape = "S-curve")

# combine them
test_scen <- rbind(test_10_lin, test_10_sat, test_10_scu,
test_250_lin, test_250_sat, test_250_scu) %>%
  mutate(fun_shape = fct_relevel(fun_shape, "linear", "saturating"),
         Light = as.factor(Light))

# figures
pdf("output/test_scenarios_time_series.pdf")
ggplot(test_scen, aes(time, seeds, color = fun_shape, linetype =
Light)) +
  geom_line() +
  xlab("Time (years)") +
  ylab("Seeds") +
  guides(color = guide_legend(title = "Function shape")) +
  fig_theme

ggplot(test_scen, aes(time, juveniles, color = fun_shape, linetype =
Light)) +
  geom_line() +
  xlab("Time (years)") +
  ylab("Juveniles") +
  guides(color = guide_legend(title = "Function shape")) +
  fig_theme

ggplot(test_scen, aes(time, adults, color = fun_shape, linetype =
Light)) +
  geom_line() +
  xlab("Time (years)") +
  ylab("Adults") +
```

```
  guides(color = guide_legend(title = "Function shape")) +
  fig_theme

ggplot(test_scen, aes(time, trees, color = fun_shape, linetype =
Light)) +
  geom_line() +
  xlab("Time (years)") +
  ylab("Trees") +
  guides(color = guide_legend(title = "Function shape")) +
  fig_theme

ggplot(test_scen, aes(time, biomass, color = fun_shape, linetype =
Light)) +
  geom_line() +
  xlab("Time (years)") +
  ylab("Biomass (g)") +
  guides(color = guide_legend(title = "Function shape")) +
  fig_theme
dev.off()
```