

Kelsay_Volkel_Assignment4

Taylor and Shea

10/11/2020

Question 1: Workplan (3 pts)

Describe your workplan for this assignment. Have you changed anything from the first group assignment?

Our workplan was to look at the assignment individually and then meet on MS Teams to discuss and work on it. I (Shea) wrote out the answers in R markdown as we discussed the assignment, and Taylor followed up with proofreading and editing. I (Shea) submitted the final draft of this assignment to Canvas.

Question 2: Natural selection (10 pts)

In class we examined a situation where survival varied linearly with climate change; one homozygote saw a decrease in survival while the heterozygote and other homozygote had increased survival as the climate changed. Many organisms have a roughly quadratic relationship between temperature and survival, as approximated below.

We decided to approach this question looking at both relative and absolute fitness. For both absolute and relative fitness, we know that each genotype's fitness depends on the genotype's survival probability (l) and fecundity (m):

```
w <- l * m
```

The relationship between time (t) and survival in these curves is given by this R code:

Also, added is a line for survival assuming the co-dominant heterozygote has intermediate fitness.

```
t <- seq(0, 50, by = 0.01) # time
l11 <- eval(expression(-1/2000 * (t - 25)^2 + 0.6)) # survival of A1A1
l22 <- eval(expression(-1/1600 * (t - 35)^2 + 0.8)) # survival of A2A2

plot(l11 ~ t, type = "l", ylim = c(0, 1), ylab = "Genotype Survival",
     xlab = "Time", col = "red")
lines(l22 ~ t, col = "blue")
# Added legend to distinguish genotypes
legend("topleft", c("A1A1", "A1A2", "A2A2"), lty = 1, col = c("red",
```

```

"purple", "blue"))

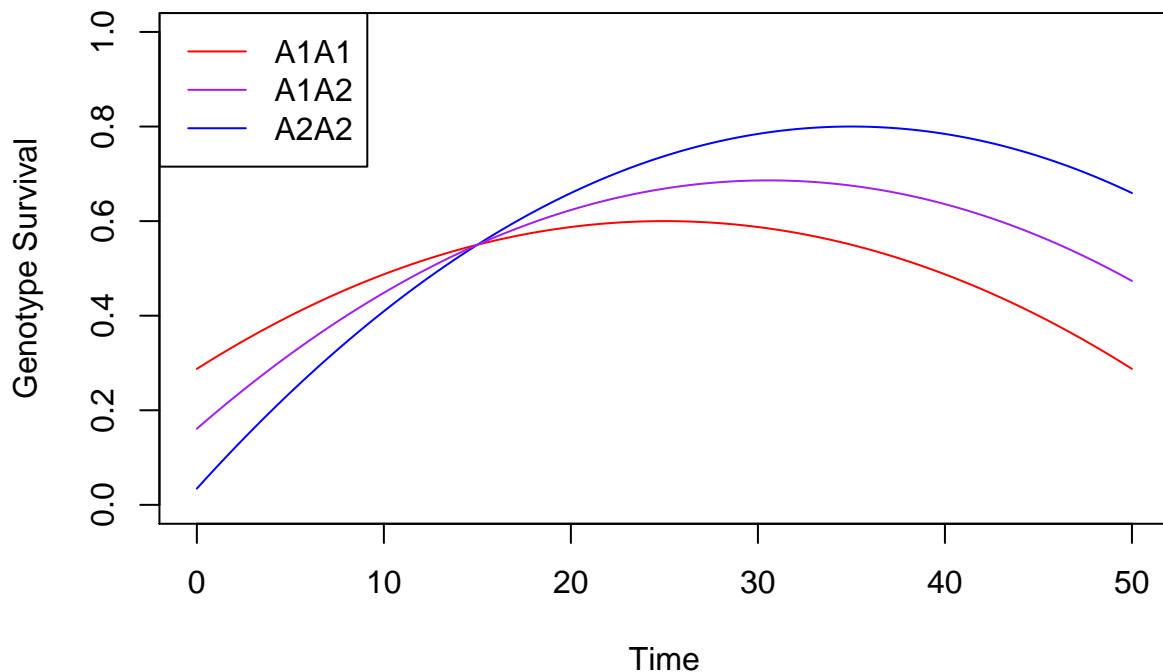
# w = l * m Fecundity m = 2 for all genotypes
w11_absolute <- l11 * 2
w22_absolute <- l22 * 2

# Heterozygote fitness will always be right in the middle of
# both homozygotes for co-dominance, which is average of
# homozygote fitness
w12_absolute <- (w11_absolute + w22_absolute)/2

# Divide by fecundity to get survival over time.
l12_absolute <- w12_absolute/2

# Plot the heterozygote, too
lines(l12_absolute ~ t, col = "purple")

```



Using these relationships produce plots for fraction allele A1 over time, population size over time, and average fitness over time. Assume $m = 2$, p (proportion of A1) = 1500/2000, $N_0=2000$ and that the alleles are codominant. Describe the dynamics represented by the plots.

Just like with the absolute fitness case, we know that each genotype's fitness depends on the genotype's survival probability (l) and fecundity(m):

```
w <- 1 * m
```

Fecundity for all genotypes is constant and equal ($m = 2$), so only the variability in survival over changing conditions of each genotype will change the relative fitness of each genotype. If we choose to go by relative fitness rather than absolute fitness, and we assume co-dominance, the relative fitness for each genotype initially are going to be:

```
A1A1: w11 <- 1
```

```
A1A2: w12 <- 1 - s/2
```

```
A2A2: w22 <- 1 - s
```

Based on the graph above, A1A1 is initially the favored genotype since its survival (l11) is initially greater than A2A2 (l22). With our assumption of codominance, we can assume that A1A2 has an intermediate level of survival, and therefore an intermediate level of fitness. The value s determines how large the difference in relative fitness is among the three genotypes and can be given any value we want if the true fitness differences are unknown. For simplicity, we'll assume s remains the same under changing conditions.

However, based on the graph, we know that conditions, and therefore relative fitness of each genotype changes with time. From the time on the graph where the survival probability of the A2A2 genotype is greater than the A1A1 genotype to the end of the time series, A2A2 will instead be the most fit genotype. Assuming s is the same when A2A2 becomes the most fit genotype, the relative fitnesses at this point will be:

```
A1A1: w11 <- 1 - s
```

```
A1A2: w12 <- 1 - s/2
```

```
A2A2: w22 <- 1
```

The relative fitnesses of the homozygote genotypes have reversed while the heterozygote will still have the same intermediate level of fitness at this point. (Heterozygote relative fitness remains constant over time.)

In this code, we figured out where in the time series that the relative fitnesses of the homozygotes switch, so we can assign a relative fitness value of 1 to the more fit homozygote at the appropriate times of the time series. We created five vectors for each homozygote genotype because we wanted to compare when there are different values of s (different magnitudes of fitness difference among genotypes).

```
m <- 2 # fecundity, though this doesn't really matter to relative fitness much here since all genotype.

# Absolute fitness values for the homozygotes based on
# quadratic survival over time
w11 <- l11 * m # unscaled fitness value for A1A1
w22 <- l22 * m # unscaled fitness value for A2A2

# Loop to figure out when each homozygote is more fit over
# time (which ever w is bigger at that time)
w_diff <- c() # empty vector for fitness difference between homozygotes
for (i in 1:5001) {
  w_diff[i] <- w11[i] - w22[i] # calculates fitness difference between homozygotes
}

# When is A1A1 more fit than A2A2?
w11_fitter <- which(w_diff > 0)

# Creating a bunch of empty vectors for A1A1 relative fitness
# Creating several because we will compare with difference s
# values
w11_relative_a <- vector(mode = "numeric", length = 5001)
w11_relative_b <- vector(mode = "numeric", length = 5001)
```

```

w11_relative_c <- vector(mode = "numeric", length = 5001)
w11_relative_d <- vector(mode = "numeric", length = 5001)
w11_relative_e <- vector(mode = "numeric", length = 5001)

# Assigning relative fitness of 1 to A1A1 relative fitness
# vectors for part of time series where A1A1 genotype is the
# most fit.
w11_relative_a[w11_fitter] <- 1 # assume fitness is 1 when w11 is bigger
w11_relative_b[w11_fitter] <- 1 # assume fitness is 1 when w11 is bigger
w11_relative_c[w11_fitter] <- 1 # assume fitness is 1 when w11 is bigger
w11_relative_d[w11_fitter] <- 1 # assume fitness is 1 when w11 is bigger
w11_relative_e[w11_fitter] <- 1 # assume fitness is 1 when w11 is bigger

# When is A2A2 more fit than A1A1? (Just the opposite)
w22_fitter <- which(w_diff < 0)

# Creating a bunch of empty vectors for A2A2 relative fitness
# Again, creating several because we will compare with
# difference s values
w22_relative_a <- vector(mode = "numeric", length = 5001)
w22_relative_b <- vector(mode = "numeric", length = 5001)
w22_relative_c <- vector(mode = "numeric", length = 5001)
w22_relative_d <- vector(mode = "numeric", length = 5001)
w22_relative_e <- vector(mode = "numeric", length = 5001)

# Assigning relative fitness of 1 to A2A2 relative fitness
# vectors for part of time series where A2A2 genotype is the
# most fit.
w22_relative_a[w22_fitter] <- 1 # assume fitness is 1 when w22 is bigger
w22_relative_b[w22_fitter] <- 1 # assume fitness is 1 when w22 is bigger
w22_relative_c[w22_fitter] <- 1 # assume fitness is 1 when w22 is bigger
w22_relative_d[w22_fitter] <- 1 # assume fitness is 1 when w22 is bigger
w22_relative_e[w22_fitter] <- 1 # assume fitness is 1 when w22 is bigger

```

The code shows the five different values that we chose for s (the difference in fitness of the inferior genotypes) to show the difference in fitness among all three genotypes. The larger s is, the greater the difference in fitness among genotypes.

```

# s values
s <- c(0, # no selective advantage of any genotype, a
      0.001, # a very minuscule advantage of one genotype over others, b
      0.005, # still very minuscule advantage of one genotype over others, c
      0.01, # small advantage, d
      0.5) # large advantage of one genotype over others, e

# for s = 0, i.e., no selection
w11_relative_a[w22_fitter] <- 1 - s[1] # fills in the rest of the A1A1 vector when it is not the most f
w22_relative_a[w11_fitter] <- 1 - s[1] # fills in the rest of the A2A2 vector when it is not the most f
w12_a <- 1 - (s[1]/2) # note heterozygote relative fitness is just a constant

# Same thing, but for s = 0.001
w11_relative_b[w22_fitter] <- 1 - s[2]
w22_relative_b[w11_fitter] <- 1 - s[2]

```

```

w12_b <- 1 - (s[2]/2)

# For s = 0.005
w11_relative_c[w22_fitter] <- 1 - s[3]
w22_relative_c[w11_fitter] <- 1 - s[3]
w12_c <- 1 - (s[3]/2)

# For s = 0.01
w11_relative_d[w22_fitter] <- 1 - s[4]
w22_relative_d[w11_fitter] <- 1 - s[4]
w12_d <- 1 - (s[4]/2)

# For s = 0.5
w11_relative_e[w22_fitter] <- 1 - s[5]
w22_relative_e[w11_fitter] <- 1 - s[5]
w12_e <- 1 - (s[5]/2)

```

Okay, so now we can figure out how the proportion of A1 allele changes over time now that that we have established our relative fitness given our assumptions.

```

# Initial proportion of the A1 allele in the population We
# created many p1 objects for each s value just so that the
# for loops run right
p1_a <- 1500/2000
p1_b <- 1500/2000
p1_c <- 1500/2000
p1_d <- 1500/2000
p1_e <- 1500/2000

# Average fitness equations for each value of s Note average
# fitness can change over time.
wt_a <- w11_relative_a * (p1_a^2) + w12_a * 2 * p1_a * (1 - p1_a) +
  w22_relative_a * ((1 - p1_a)^2)
wt_b <- w11_relative_b * (p1_b^2) + w12_b * 2 * p1_b * (1 - p1_b) +
  w22_relative_b * ((1 - p1_b)^2)
wt_c <- w11_relative_c * (p1_c^2) + w12_c * 2 * p1_c * (1 - p1_c) +
  w22_relative_c * ((1 - p1_c)^2)
wt_d <- w11_relative_d * (p1_d^2) + w12_d * 2 * p1_d * (1 - p1_d) +
  w22_relative_d * ((1 - p1_d)^2)
wt_e <- w11_relative_e * (p1_e^2) + w12_e * 2 * p1_e * (1 - p1_e) +
  w22_relative_e * ((1 - p1_e)^2)

# For s = 0
for (i in 1:5000) {
  p_prime_a <- p1_a[i] * ((p1_a[i] * w11_relative_a[i]) + (1 -
    p1_a[i]) * w12_a)/wt_a[i]
  p1_a <- c(p1_a, p_prime_a)
}

# For s = 0.001
for (i in 1:5000) {
  p_prime_b <- p1_b[i] * ((p1_b[i] * w11_relative_b[i]) + (1 -
    p1_b[i]) * w12_b)/wt_b[i]

```

```

    p1_b <- c(p1_b, p_prime_b)
  }

  # For s = 0.005
  for (i in 1:5000) {
    p_prime_c <- p1_c[i] * ((p1_c[i] * w11_relative_c[i]) + (1 -
      p1_c[i]) * w12_c)/wt_c[i]
    p1_c <- c(p1_c, p_prime_c)
  }
  head(which(p1_c >= 1)) # figure out when fixation reached

```

```
## [1] 326 327 328 329 330 331
```

```

p1_c[c(326:5001)] <- 1 # Change all values to 1 once fixation reached

  # For s = 0.01
  for (i in 1:5000) {
    p_prime_d <- p1_d[i] * ((p1_d[i] * w11_relative_d[i]) + (1 -
      p1_d[i]) * w12_d)/wt_d[i]
    p1_d <- c(p1_d, p_prime_d)
  }
  head(which(p1_d >= 1)) # figure out when fixation reached

```

```
## [1] 164 165 166 167 168 169
```

```

p1_d[c(164:5001)] <- 1 # Change all values to 1 once fixation reached

  # For s = 0.5
  for (i in 1:5000) {
    p_prime_e <- p1_e[i] * ((p1_e[i] * w11_relative_e[i]) + (1 -
      p1_e[i]) * w12_e)/wt_e[i]
    p1_e <- c(p1_e, p_prime_e)
  }
  head(which(p1_e >= 1)) # figure out when fixation reached

```

```
## [1] 5 6 7 8 9 10
```

```
p1_e[c(5:5001)] <- 1 # Change all values to 1 once fixation reached
```

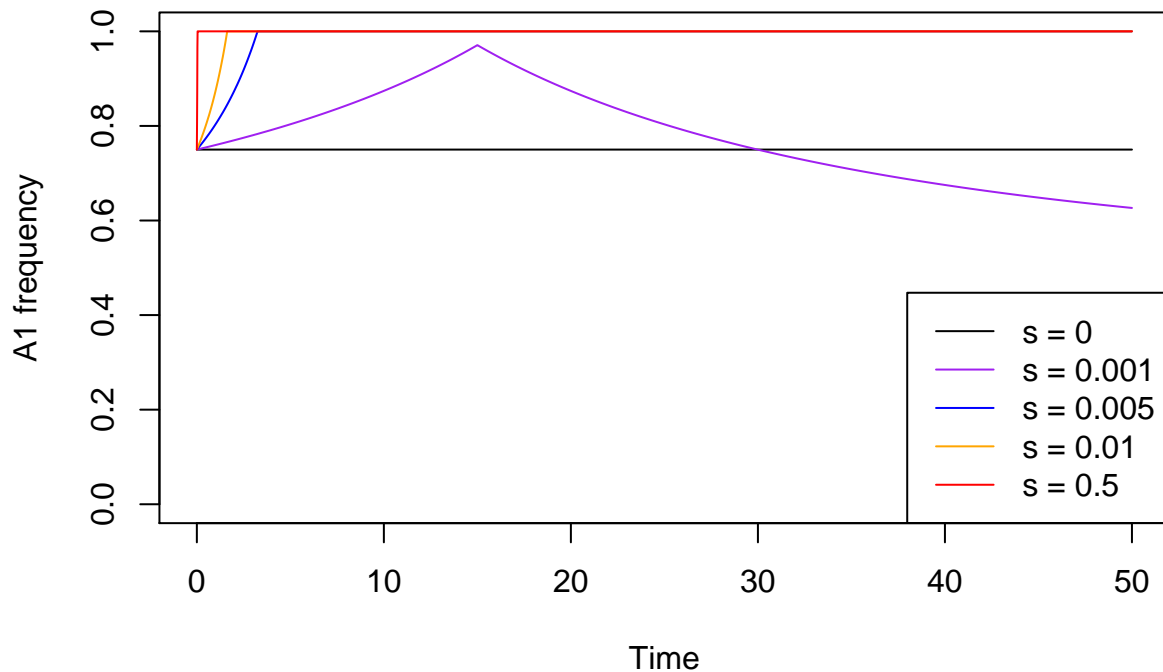
Now let's plot the frequency of A1A1 over time for each value of s.

```

plot(p1_a ~ t, type = "l", ylim = c(0, 1), ylab = "A1 frequency",
     xlab = "Time") # s = 0
lines(p1_b ~ t, col = "purple")
lines(p1_c ~ t, col = "blue")
lines(p1_d ~ t, col = "orange")
lines(p1_e ~ t, col = "red")

legend("bottomright", c("s = 0", "s = 0.001", "s = 0.005", "s = 0.01",
  "s = 0.5"), lty = 1, col = c("black", "purple", "blue", "orange",
  "red"))

```



With A1 starting at a 75% allele frequency, the A1 allele will usually reach fixation fairly quickly long before environmental change occurs, which is problematic as this genotype will be less advantageous when the environmental change happens. How quickly it reaches fixation depends on how big it's selective advantage is initially. The exception is when the relative fitness advantage of A1A1 (s) is nonexistent (no selective advantage or neutral, $s = 0$). If there is no selective advantage among genotypes (neutral genes), the allele frequency of A1 remains neutrally stable.

Let's try this based on the absolute fitnesses, too.

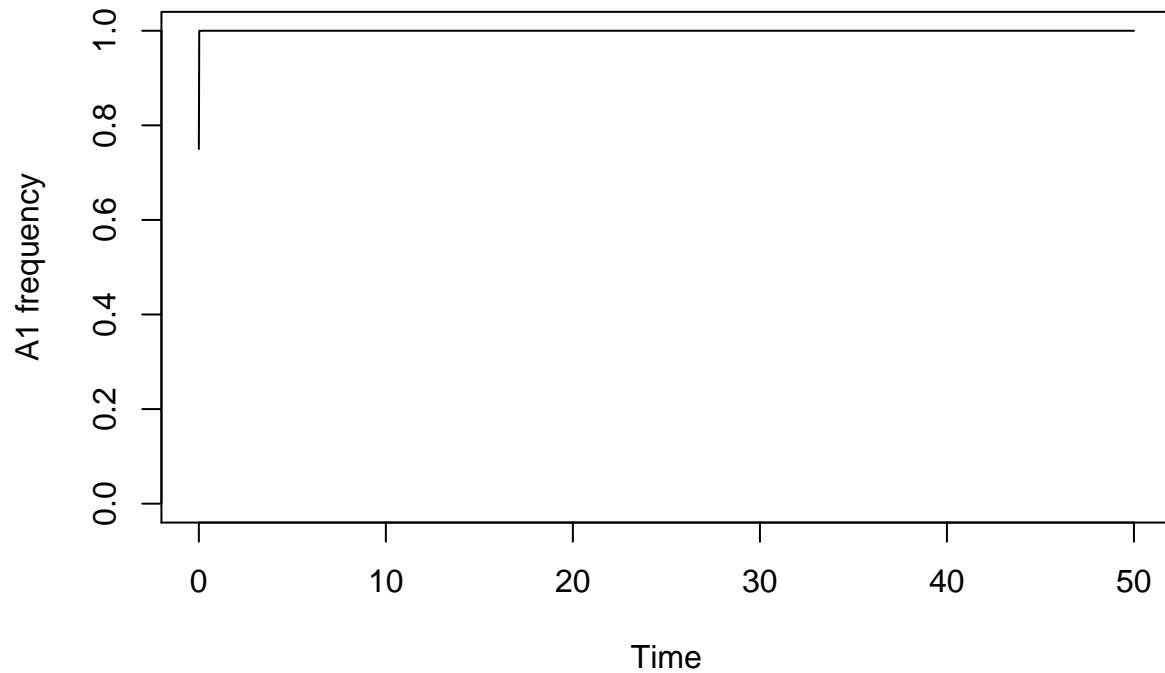
```
p1_absolute <- 1500/2000

w11_absolute <- l11 * 2
w22_absolute <- l22 * 2
w12_absolute <- (w11_absolute + w22_absolute)/2
wt_absolute <- w11_absolute * (p1_absolute^2) + w12_absolute *
  2 * p1_absolute * (1 - p1_absolute) + w22_absolute * ((1 -
    p1_absolute)^2)

for (i in 1:5000) {
  p_prime_absolute <- p1_absolute[i] * ((p1_absolute[i] * w11_absolute[i]) +
    (1 - p1_absolute[i]) * w12_absolute[i])/wt_absolute[i]
  p1_absolute <- c(p1_absolute, p_prime_absolute)
}
fixed <- min(which(p1_absolute >= 1)) # figure out when fixation reached
p1_absolute[c(fixed:5001)] <- 1

plot(p1_absolute ~ t, type = "l", ylim = c(0, 1), ylab = "A1 frequency",
```

```
xlab = "Time") # s = 0
```



The A1 allele goes to fixation very quickly. This matches the higher s value scenarios when estimated with relative fitness. This means the selection is strong for the A1 allele.

Now let's calculate and plot the population size over time.

```
N0 <- 2000 # initial population size

# s = 0
n_a <- N0
for (i in 1:5000) {
  n_prime_a <- n_a[i] * wt_a[i] # basically  $n[t+1] = n[t] * wt[t]$ 
  n_a <- c(n_a, n_prime_a)
}

# s = 0.001
n_b <- N0
for (i in 1:5000) {
  n_prime_b <- n_b[i] * wt_b[i]
  n_b <- c(n_b, n_prime_b)
}

# s = 0.005
n_c <- N0
for (i in 1:5000) {
  n_prime_c <- n_c[i] * wt_c[i]
```



```

    n_c <- c(n_c, n_prime_c)
}

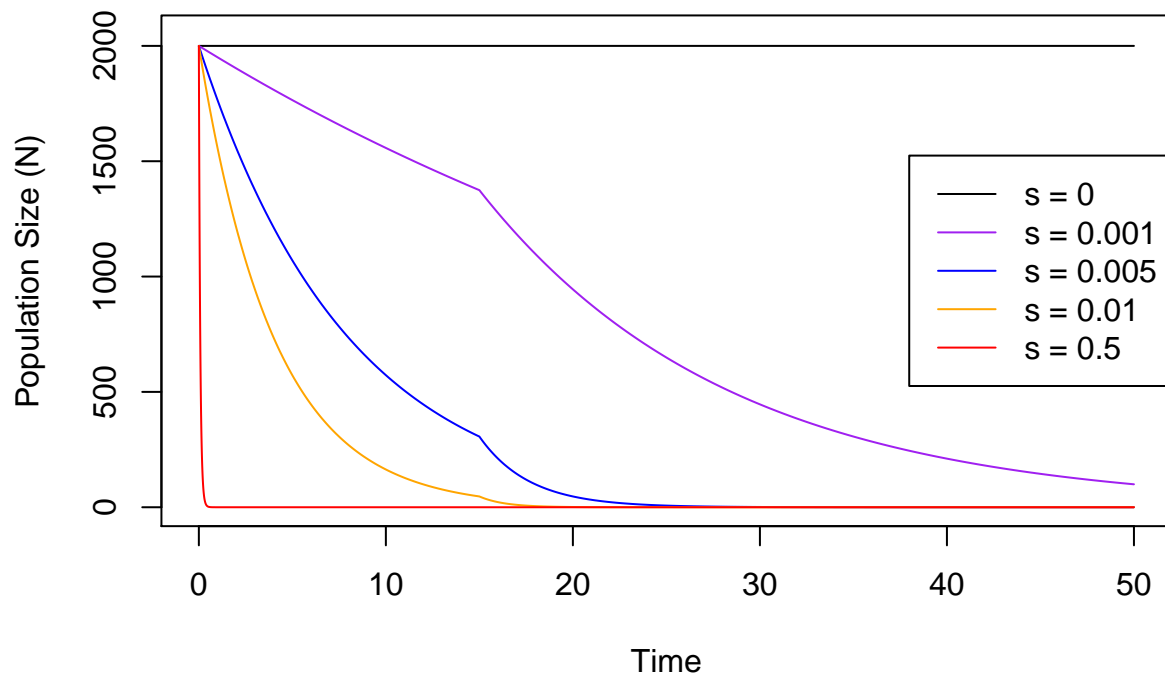
# s = 0.01
n_d <- NO
for (i in 1:5000) {
  n_prime_d <- n_d[i] * wt_d[i]
  n_d <- c(n_d, n_prime_d)
}

# s = 0.5
NO <- 2000 # initial population size
n_e <- NO
for (i in 1:5000) {
  n_prime_e <- n_e[i] * wt_e[i]
  n_e <- c(n_e, n_prime_e)
}

plot(n_a ~ t, type = "l", ylab = "Population Size (N)", xlab = "Time",
     ylim = c(0, 2050))
lines(n_b ~ t, col = "purple")
lines(n_c ~ t, col = "blue")
lines(n_d ~ t, col = "orange")
lines(n_e ~ t, col = "red")

legend("right", c("s = 0", "s = 0.001", "s = 0.005", "s = 0.01",
  "s = 0.5"), lty = 1, col = c("black", "purple", "blue", "orange",
  "red"))

```



Again, unless the selective advantage of the superior homozygote is minuscule or there is no selection (no advantage/disadvantage over time), then the population will go extinct. If no selection, the population size remains constant (at least in a deterministic model). If there is basically any selective difference among genotypes, the population will still be greatly reduced even if it doesn't go extinct.

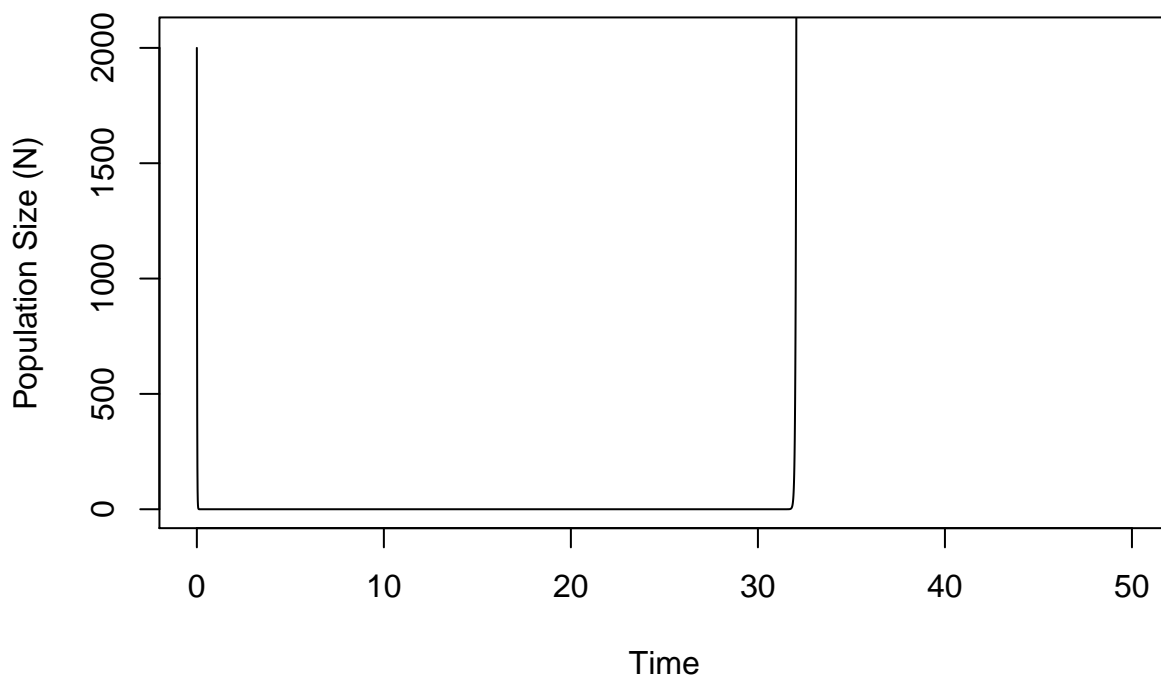
Let's do the same thing for absolute fitness

```
N0 <- 2000 # initial population size

# s = 0
n_absolute <- N0
for (i in 1:5000) {
  n_prime_absolute <- n_absolute[i] * wt_absolute[i] # basically  $n[t+1] = n[t] * wt[t]$ 
  n_absolute <- c(n_absolute, n_prime_absolute)
}
min(n_absolute) # nearly, but not quite zero since this is a deterministic model
```

```
## [1] 5.97264e-158
```

```
plot(n_absolute ~ t, type = "l", ylab = "Population Size (N)",
     xlab = "Time", ylim = c(0, 2050))
```

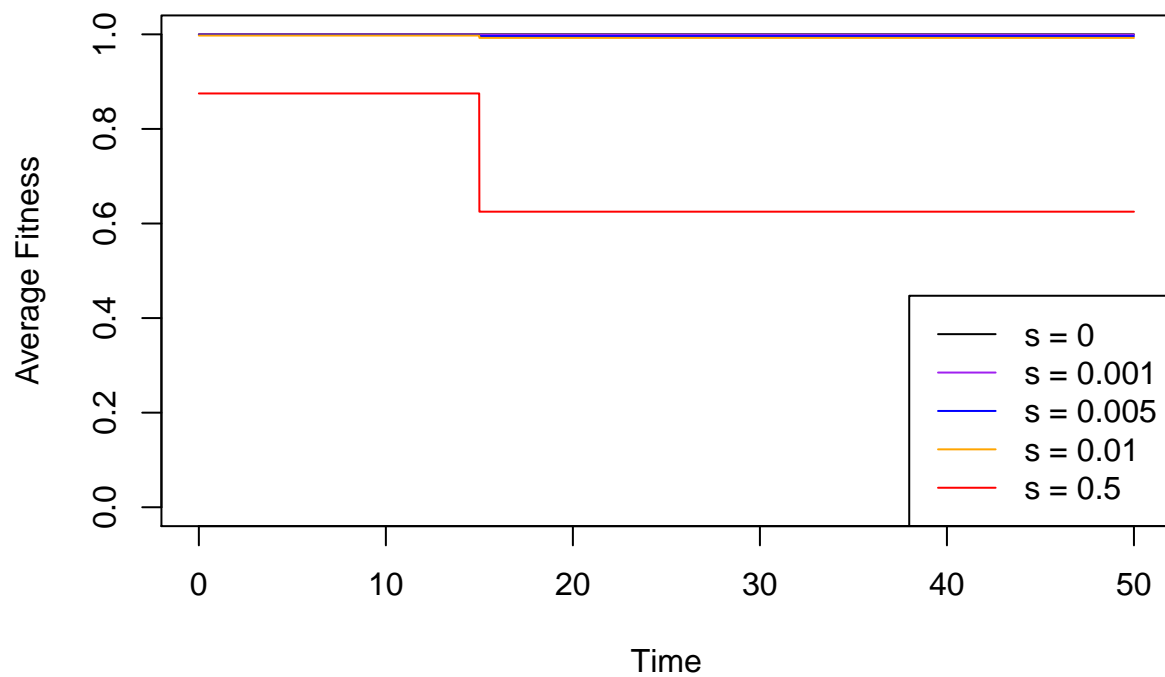


In a realistic scenario, this population would have gone extinct with changing conditions similar to many of the models based on relative fitness, but because the deterministic model allows fractional population sizes, the population does not quite reach zero, and grows exponentially eventually, likely when average fitness increases enough for the population to grow.

Now let's plot average fitness over time (already calculated earlier).

```
plot(wt_a ~ t, type = "l", ylim = c(0, 1), xlab = "Time", ylab = "Average Fitness")
lines(wt_b ~ t, col = "purple")
lines(wt_c ~ t, col = "blue")
lines(wt_d ~ t, col = "orange")
lines(wt_e ~ t, col = "red")

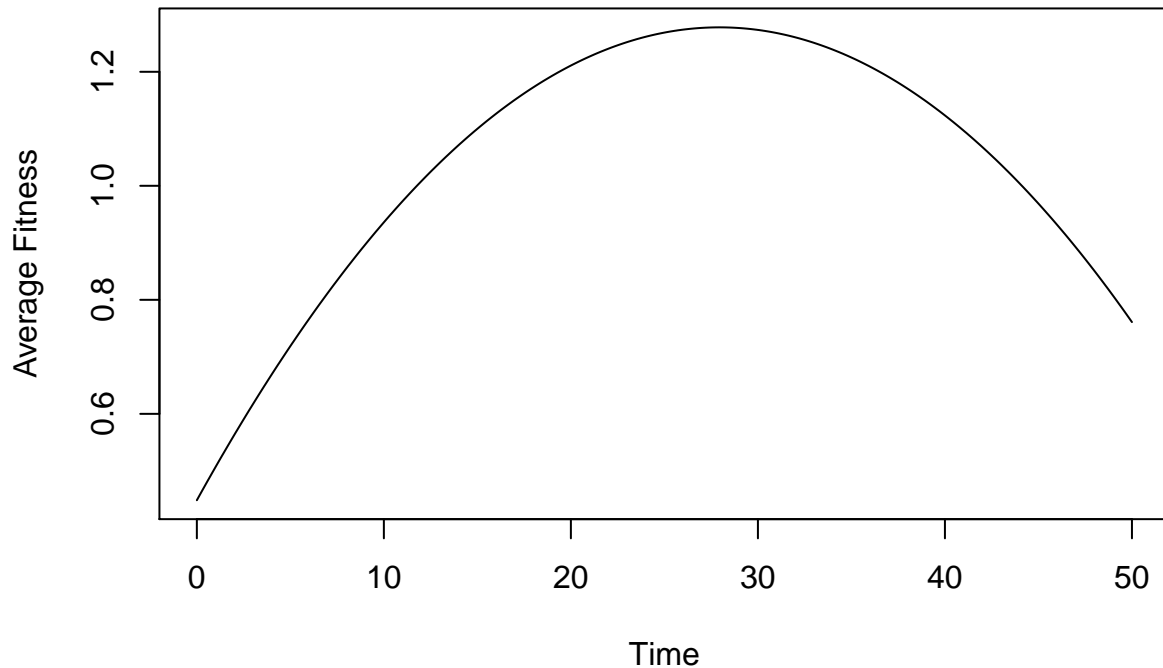
legend("bottomright", c("s = 0", "s = 0.001", "s = 0.005", "s = 0.01",
  "s = 0.5"), lty = 1, col = c("black", "purple", "blue", "orange",
  "red"))
```



Unless there is no selection, average fitness will decrease and then remain constant at that value after the inflection point (on the first graph) where the A2A2 genotype becomes more fit than the A1A1 genotype. The decrease in fitness is larger with larger s values.

Again, let's do the same thing for absolute average fitness

```
plot(wt_absolute ~ t, type = "l", xlab = "Time", ylab = "Average Fitness")
```



Average fitness increases with time, but the rate of increase slows as environmental conditions begin to change until it peaks and begins to decrease. However, the population would have gone extinct before this peak fitness is reached.

It's important to note differences in the plots based on relative and absolute fitness. Relative fitness models where the value of s is close to the absolute fitness difference among genotypes show similar results in plots, in that A_1 reaches fixation quickly, the population goes extinct (or would go extinct in the absolute fitness case if fractional population values less than 1 dropped to 0 like they would realistically because you can have a fraction of individuals), and fitness eventually decreases by the end of the time series. Though the relative fitness models don't show the same adaptive peak pattern the absolute fitness model shows because the relative fitness of the genotypes has only one value (constant, in the case of the heterozygote) or two values (1 or $1-s$ for the homozygotes) over time, where absolute fitnesses change continuously in a quadratic manner.

Overall, this shows the concern that, at least under a situation of co-dominance, that a species can easily be at risk of extinction if genetic diversity is lost quickly or rare when major environmental change occurs (and to some degree if diversity not initially very strong). Even if the population could manage to persist, its population can drop drastically and be harder to conserve. How long the population persists depends on the relative fitness and proportions of the existing diversity in the population (and probably population size). Though in this case the absolute fitness shows that this population is not genetically diverse enough to persist. And this is all without considering stochasticity. Other evolutionary mechanisms can either counteract or exacerbate this issue.

Question 3: Heterozygote advantage (7 pts)

Use the idea of adaptive topography to describe a case of heterozygote advantage (as in Figure 5.14). You may choose any biologically plausible values that results in a stable equilibrium with $0.5 < p < 1$, where p is the proportion of A1 in a two allele system. Show the result in a plot and give the equilibrium value of p .

The relative fitness of the heterozygote is 1 (more fit than both homozygotes), and we'll assume that the relative fitness of both homozygotes is equal to $1 - s$. We'll again try out different values for s in a similar fashion to the previous question.

```
s <- seq(0.1, 1, 0.1) # s values from 0.1-1 by increment of 0.1
p <- seq(0, 1, 0.01) # vector of p values (A1 allele frequencies)

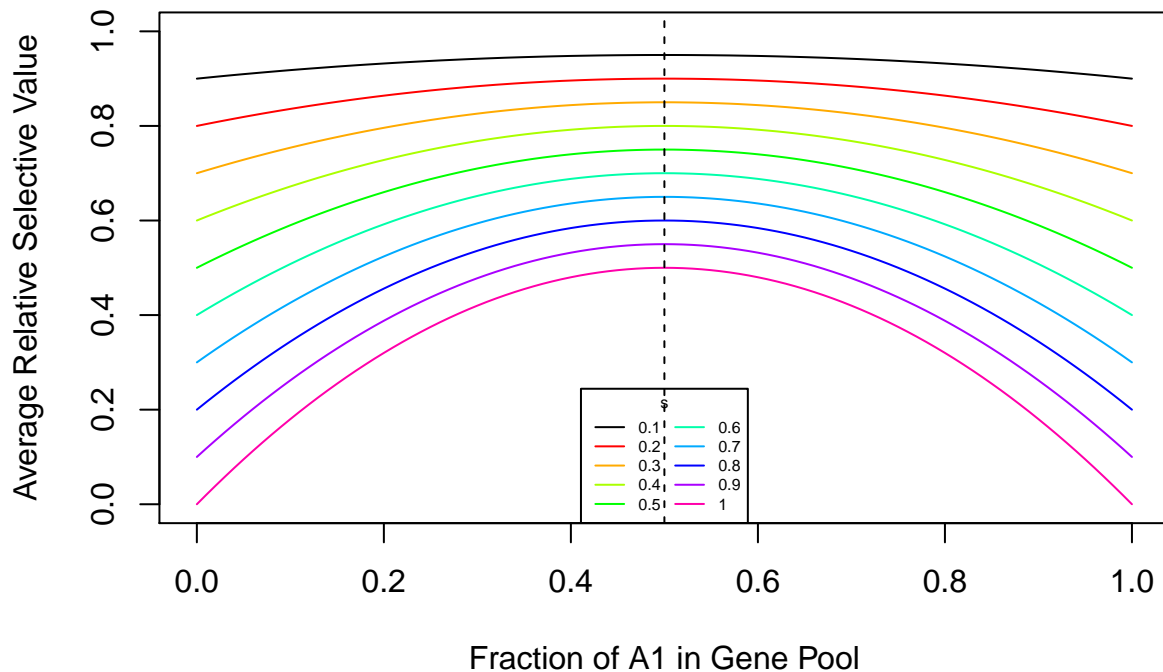
# For average fitness to peak at 0.5 with a heterozygote
# advantage, no matter what value of s, the relative
# fitnesses of A1A1, A1A2, and A2A2, respectively, need to
# be: w11 = 1 - s w12 = 1 w22 = 1 - s

# 1 = p^2 + 2p(1-p) + (1-p)^2 A1A1 + A1A2 + A2A2

# For loop that creates a list of vectors of the calculated
# average fitness relative to A1 frequency (p) for different
# s values.
wbar <- NULL
for (i in 1:length(s)) {
  wbar[[i]] <- (1 - s[i]) * p^2 + 2 * p * (1 - p) * 1 + ((1 -
    p)^2) * (1 - s[i])
}

# Plot average fitness relative to p for each value of s
colors <- rainbow(9)
plot(wbar[[1]] ~ p, type = "l", ylim = c(0, 1), ylab = "Average Relative Selective Value",
     xlab = "Fraction of A1 in Gene Pool")
for (i in 2:10) {
  lines(wbar[[i]] ~ p, lty = 1, col = colors[i - 1])
}
abline(v = p[51], lty = 2)

legend("bottom", c("0.1", "0.2", "0.3", "0.4", "0.5", "0.6",
  "0.7", "0.8", "0.9", "1"), lty = 1, col = c("black", colors),
     cex = 0.5, title = "s", ncol = 2)
```



In all of these scenarios, fitness peaks at $p = 0.5$ (dotted line), i.e., when both alleles are equally prevalent and this equilibrium is stable when $0.5 < p < 1$. The smaller the heterozygote advantage, then the less steep the adaptive hill and the higher the fitness at the peak. The converse is also true. The larger the heterozygote advantage compared to the homozygotes, the steeper the adaptive peak and the lower the fitness at the peak. There is no peak if the heterozygote is no more fit than the other genotypes (no selection).

Question 4: Genetic drift and maintenance of genetic diversity (10 pts)

Last assignment we considered the minimum viable population size of a population of conservation concern. Long term conservation also requires maintenance of genetic diversity. Using simulations and starting with the example of drift mixed with selection from the chapter, determine the minimum population size to provide a 99% chance of preserving both alleles for 100 generations. (This assumes constant population size). Explain (in words) how genetic would interact with demographic stochasticity in a realistic population, as studied in the previous assignment. Does this have any implications for conservation management?

Adapt MATLAB function on page 188 to R.

```
# w11, relative fitness of A1A1 genotype w12, relative
# fitness of A1A2 genotype w22, relative fitness of A2A2
```

```

# genotype p0, initial proportion of A1 allele n, population
# size time, time (number of generations)

drift_ns <- function(w11, w12, w22, p0, n, time) {
  p <- p0 # starting value for p
  for (t in 1:time) {
    # for however many generations you choose
    a1 <- 0 # starting amount A1 alleles
    for (i in 1:(2 * n)) {
      # genes drawn from population random draw of A1 or A2 allele,
      # genetic drift
      if (runif(1) < p[t]) {
        a1 <- a1 + 1 # if A1 allele drawn, add 1 to A1 allele count
      }
    }
    pd <- a1/(2 * n) # proportion of A1 allele in population after drift

    # Natural selection portion based on allele frequencies
    # resulting from drift
    wbar <- (pd^2) * w11 + 2 * pd * (1 - pd) * w12 + ((1 -
      pd)^2) * w22 # average fitness
    pprime <- (pd * w11 + (1 - pd) * w12) * pd/wbar # change in p over time
    p <- c(p, pprime) # vector of A1 allele frequencies
  }
  return(p) # return that vector
}

```

Okay, now let's use the book values for each of the variables (looks like a heterozygote advantage with initially a 50/50 split in allele frequencies), and incrementally N0 until we have a large enough population to maintain genetic diversity for 100 generations. Let's start with 2 individuals like the book does, as absurd as that is.

```

# w11 = 0.5 w12 = 1 w22 = 0.5 p0 = 0.5 n = 2 time = 100

# 100 simulations of 100 generations
sim2 <- NULL
for (i in 1:100) {
  sim2[[i]] <- drift_ns(0.5, 1, 0.5, 0.5, 2, 100)
}

# Change list to dataframe to make it easier to calculate
# probability of genetic diversity persistence
sim2_df <- data.frame(matrix(unlist(sim2), ncol = 100, byrow = FALSE))

# Probability of genetic diversity persistence (proportion of
# simulations where neither allele became fixed)
sim2_prob <- (length(which(sim2_df[100, ] != 0 & sim2_df[100,
  ] != 1)))/100
sim2_prob # number will be a little different each time

```

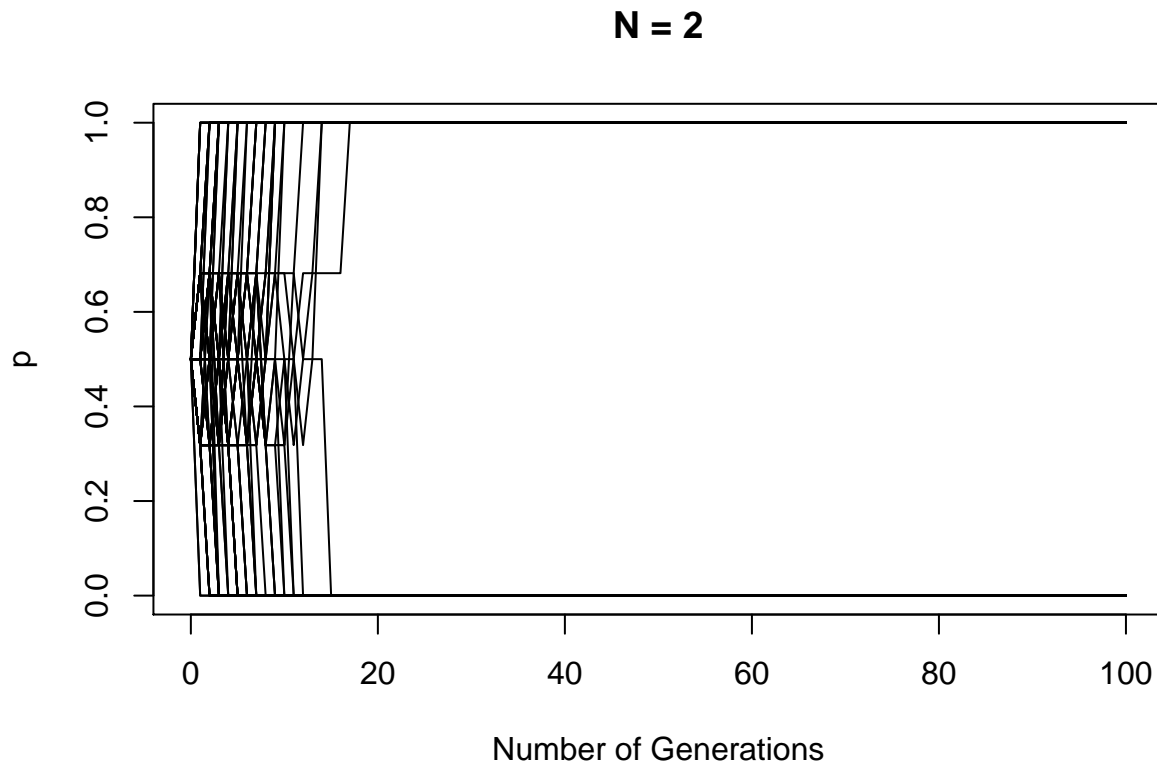
```
## [1] 0
```



```

# Plot simulation results (results will vary each time code
# is run)
plot(sim2_df[, 1] ~ c(0:100), type = "l", ylab = "p", xlab = "Number of Generations",
     main = "N = 2", ylim = c(0, 1))
for (i in 2:100) {
  lines(sim2_df[, i] ~ c(0:100))
}

```



Two individuals are not sufficient to maintain genetic diversity under these circumstances.

Let's increase it to 10 individuals to see if that helps much.

```

sim10 <- NULL
for (i in 1:100) {
  sim10[[i]] <- drift_ns(0.5, 1, 0.5, 0.5, 10, 100)
}

sim10_df <- data.frame(matrix(unlist(sim10), ncol = 100, byrow = FALSE))

sim10_prob <- (length(which(sim10_df[100, ] != 0 & sim10_df[100,
] != 1)))/100
sim10_prob

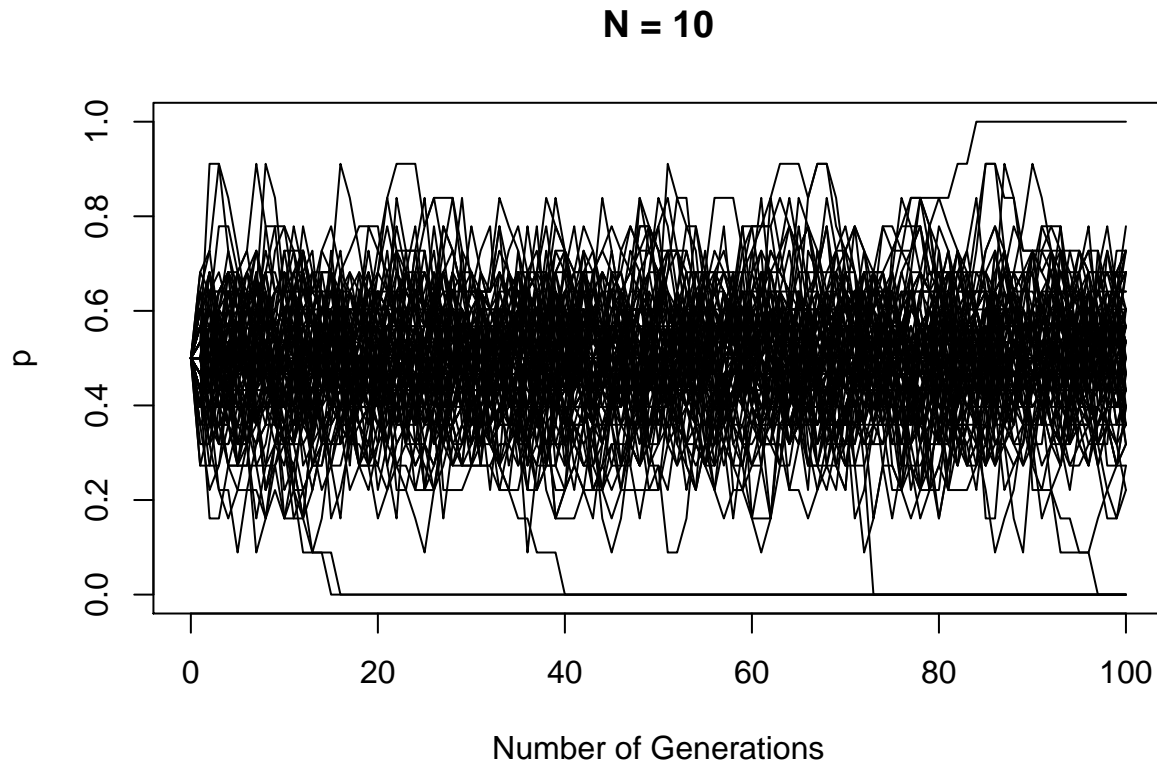
## [1] 0.94

```

```

plot(sim10_df[, 1] ~ c(0:100), type = "l", ylab = "p", xlab = "Number of Generations",
     main = "N = 10", ylim = c(0, 1))
for (i in 2:100) {
  lines(sim10_df[, i] ~ c(0:100))
}

```



So even with just 10 individuals, the chances of genetic diversity persisting is already around 85%. Not bad. Now let's increase it to 12 individuals.

```

sim12 <- NULL
for (i in 1:100) {
  sim12[[i]] <- drift_ns(0.5, 1, 0.5, 0.5, 12, 100)
}

sim12_df <- data.frame(matrix(unlist(sim12), ncol = 100, byrow = FALSE))

sim12_prob <- (length(which(sim12_df[100, ] != 0 & sim12_df[100,
] != 1)))/100
sim12_prob

```

```
## [1] 0.95
```

```

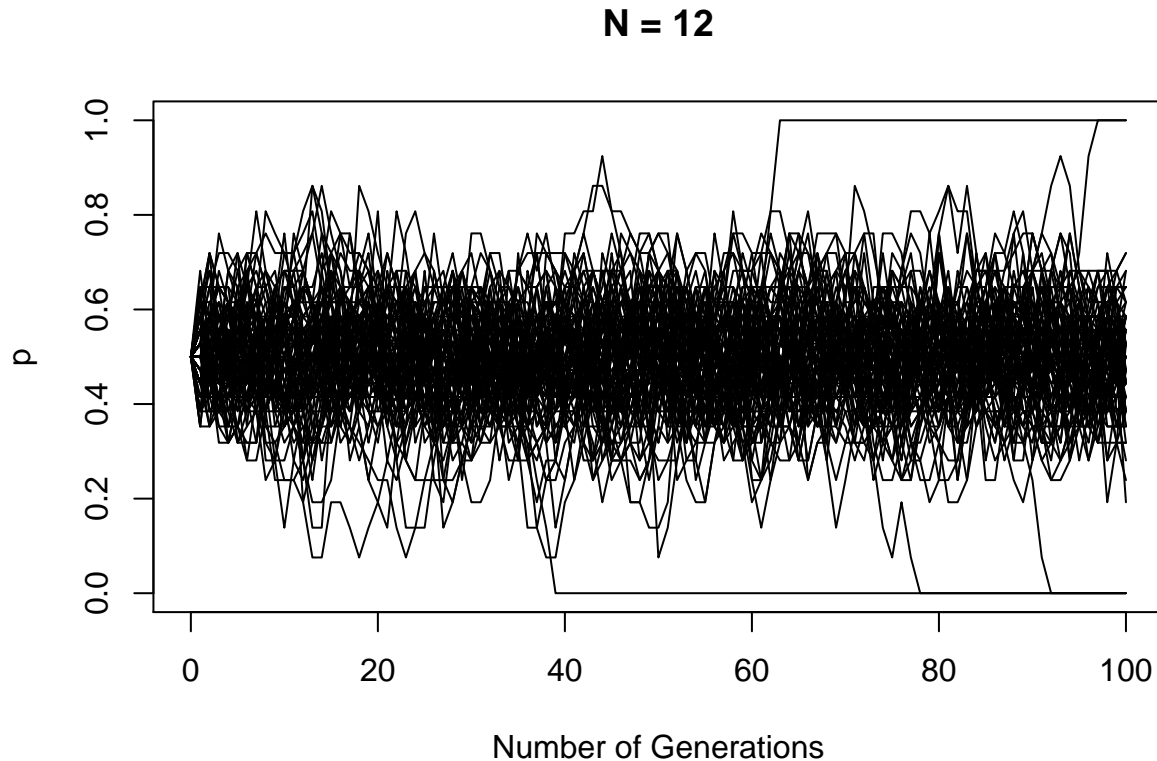
plot(sim12_df[, 1] ~ c(0:100), type = "l", ylab = "p", xlab = "Number of Generations",
     main = "N = 12", ylim = c(0, 1))
for (i in 2:100) {

```

```

    lines(sim12_df[, i] ~ c(0:100))
}

```



Around 91%-97%, getting closer!

Let's try 13 individuals

```

sim13 <- NULL
for (i in 1:100) {
  sim13[[i]] <- drift_ns(0.5, 1, 0.5, 0.5, 13, 100)
}

sim13_df <- data.frame(matrix(unlist(sim13), ncol = 100, byrow = FALSE))

sim13_prob <- (length(which(sim13_df[100, ] != 0 & sim13_df[100,
] != 1)))/100
sim13_prob

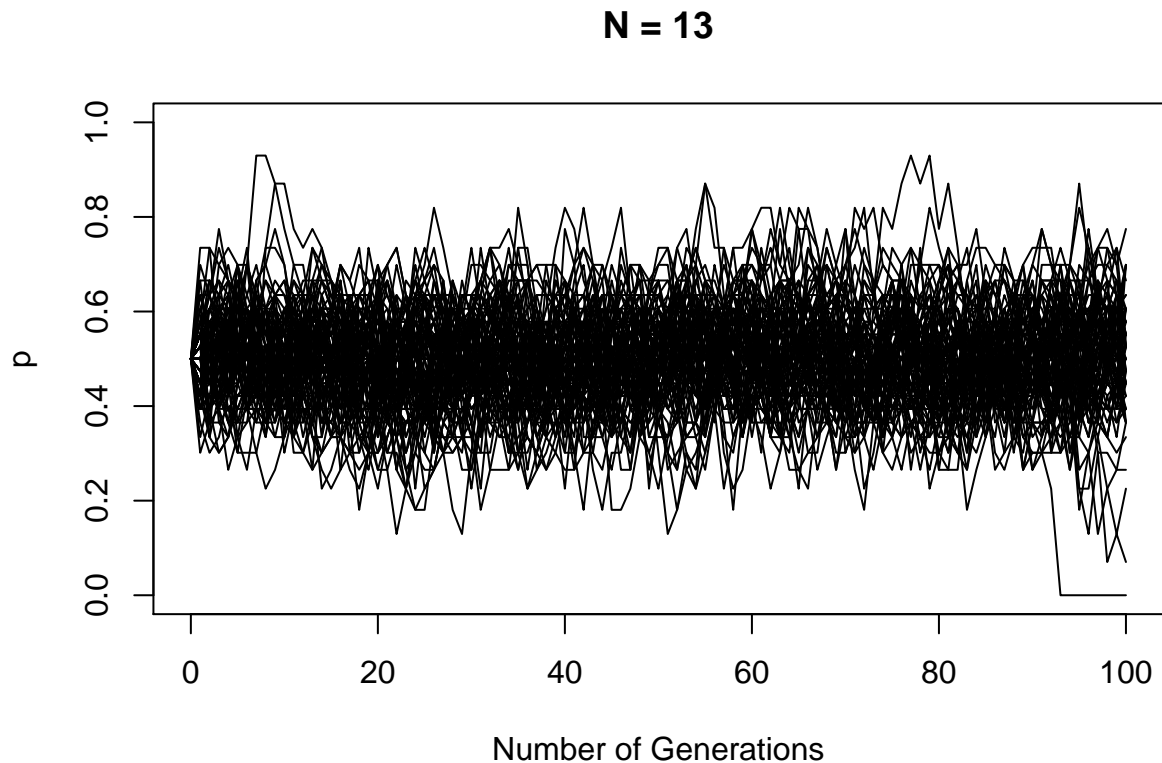
```

```
## [1] 0.99
```

```

plot(sim13_df[, 1] ~ c(0:100), type = "l", ylab = "p", xlab = "Number of Generations",
     main = "N = 13", ylim = c(0, 1))
for (i in 2:100) {
  lines(sim13_df[, i] ~ c(0:100))
}

```



Can sometimes get up to 99% persistence, but not consistently.

Let's increase 15 individuals, which is likely the approximate threshold.

```
sim15 <- NULL
for (i in 1:100) {
  sim15[[i]] <- drift_ns(0.5, 1, 0.5, 0.5, 15, 100)
}

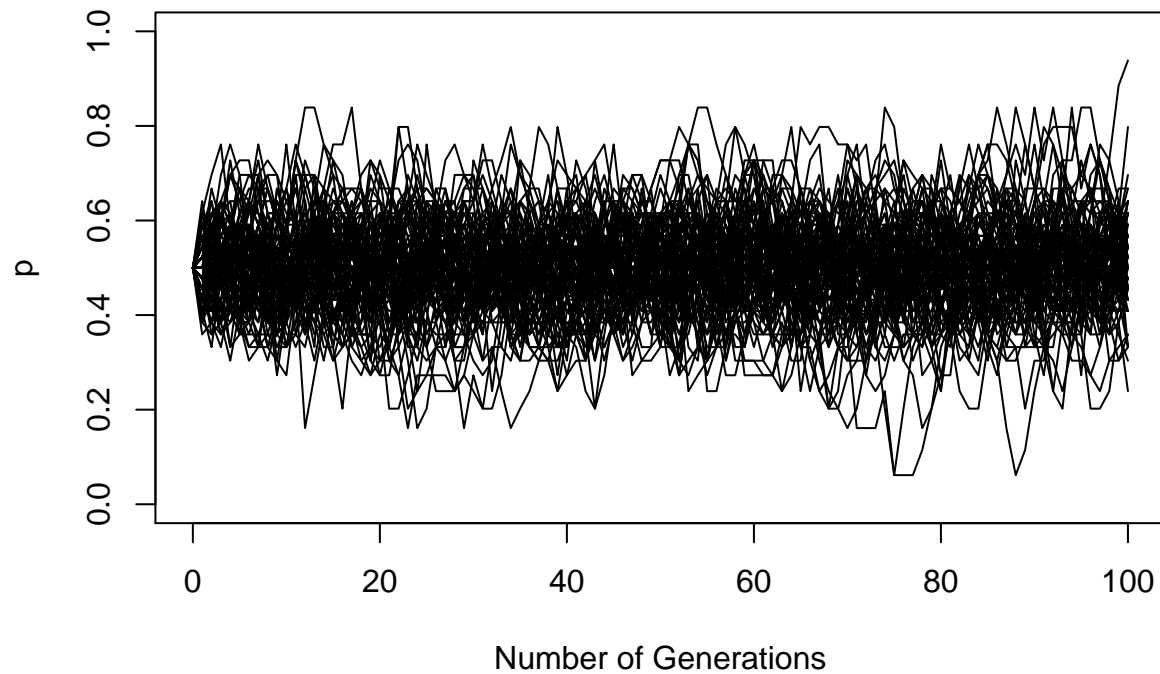
sim15_df <- data.frame(matrix(unlist(sim15), ncol = 100, byrow = FALSE))

sim15_prob <- (length(which(sim15_df[100, ] != 0 & sim15_df[100,
] != 1)))/100
sim15_prob
```

```
## [1] 1
```

```
plot(sim15_df[, 1] ~ c(0:100), type = "l", ylab = "p", xlab = "Number of Generations",
     main = "N = 15", ylim = c(0, 1))
for (i in 2:100) {
  lines(sim15_df[, i] ~ c(0:100))
}
```

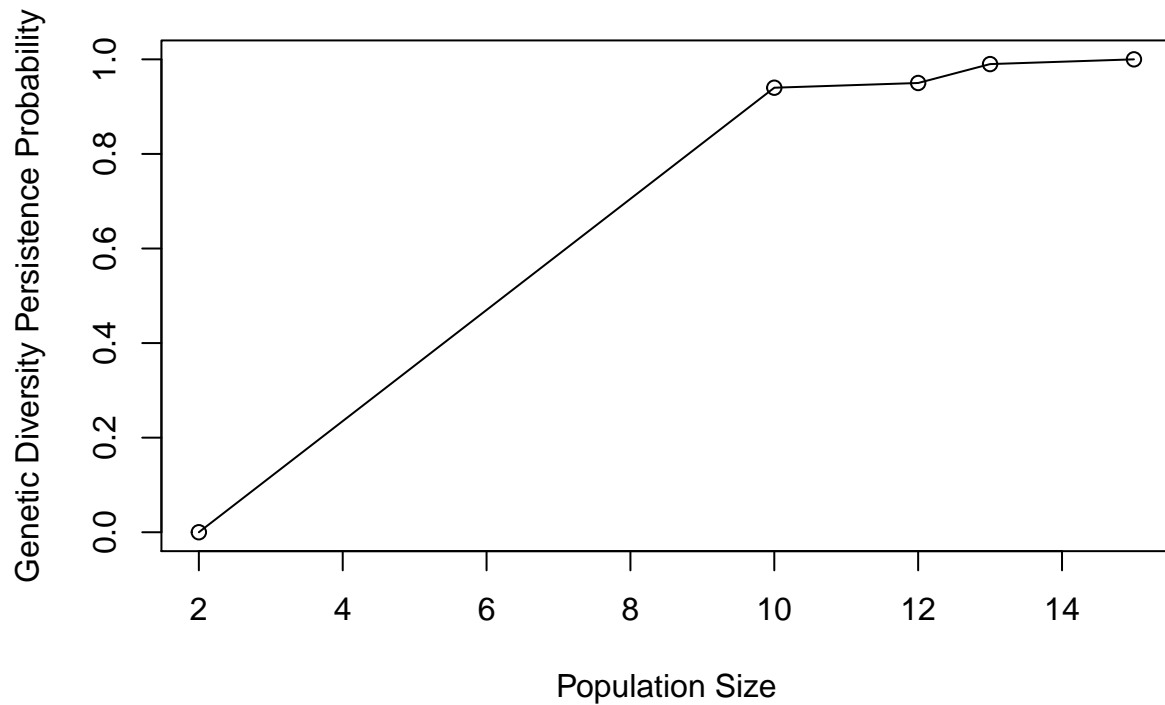
N = 15



More consistently, simulations yield 99%-100% persistence of genetic diversity.

Let's graph this trend.

```
plot(c(sim2_prob, sim10_prob, sim12_prob, sim13_prob, sim15_prob) ~  
      c(2, 10, 12, 13, 15), ylab = "Genetic Diversity Persistence Probability",  
      xlab = "Population Size")  
lines(c(sim2_prob, sim10_prob, sim12_prob, sim13_prob, sim15_prob) ~  
       c(2, 10, 12, 13, 15))
```



Under these conditions, it only takes about 15 individuals for genetic diversity to be maintained over 100 generations. However, in relation to demographic stochasticity, more than 15 individuals would be needed for the population itself (not just its genetic diversity) to persist. From the previous homework, it took over 100 individuals for the population to persist. Therefore, it's better to pick the larger minimum population size that's needed to maintain both genetic diversity and the population itself. In this case, maintaining genetic diversity is not that difficult. However, in the earlier example for question 2 under changing conditions where one allele is likely to be fixed, maintaining genetic diversity can be equally or more important than the amount of individuals. For conservation purposes, it's generally best to care about both population size and genetic diversity though.