Genetic Drift and the Coalescent

Gaurav Kandlikar

November 20, 2015

Question 1

What is the expected number of successes in a sample of size 10 from the binomial distribution with probability of success p=0.1?

First, figure this out analytically based on the formulas from class.

$$E[X] = np = 0.1 * 10 = 1$$

Second, write a simulation in R to confirm this.

```
reps <- 10000
n <- 10
p <- 0.1

rr <- rbinom(reps, n, p)
table(rr)
```

```
## rr
##    0    1    2    3    4    5    6    7
## 3538 3805 1954  580  101   20    1    1
```

```
mean(rr)
```

```
## [1] 0.997
```

The mean of 10000 randomly drawn samples of size 10 from a binomial distribution with $p = 0.1$ is ~1.

Using R to simulate genetic drift Consider that `p` is the allele frequency in the population; `N` is the number of diploid individuals.

```
p <- 0.1
N <- 10
count <- rbinom(1, 2*N, p)
t2_freq <- round(count/(2*N), 3); t2_freq
```

```
## [1] 0.1
```

The frequency of the allele has shifted from p_i to 0.1

Question 2

Write a function in R that will simulate T generations of genetic drift for L independent SNPs. Keep track of the allele frequencies of each of the L SNPs in each of the T generations. All SNPs should start in the initial generation at frequency p.

```
gene_frequencies <- function(num_generations = 1000, init_freq = 0.1, num_genes = 3,
                             pop_size = 1000) {
  full_matrix <- matrix(nrow = num_generations, ncol = num_genes)
  full_matrix[1, ] <- rep(init_freq, num_genes)
  for (i in 2:num_generations) {
    freq <- rbinom(num_genes, 2*pop_size, full_matrix[i-1, ])/(2*pop_size)
    full_matrix[i, ] <- freq
  }
  return(full_matrix)
}
```

Question 3

a) Use the function that you just wrote to simulate drift with the following parameters: N=100, L=1000, T=10000, p=0.1

```
freqs <- gene_frequencies(num_generations = 10000, init_freq = 0.1, num_genes = 1000, pop_size = 100)
```

b) How many of the 1000 SNP are at frequency 0 at the end of the simulation (in generation 10000)? c) How many are at frequency 1?

Table the last row of the output matrix:

```
table(freqs[nrow(freqs),])
```

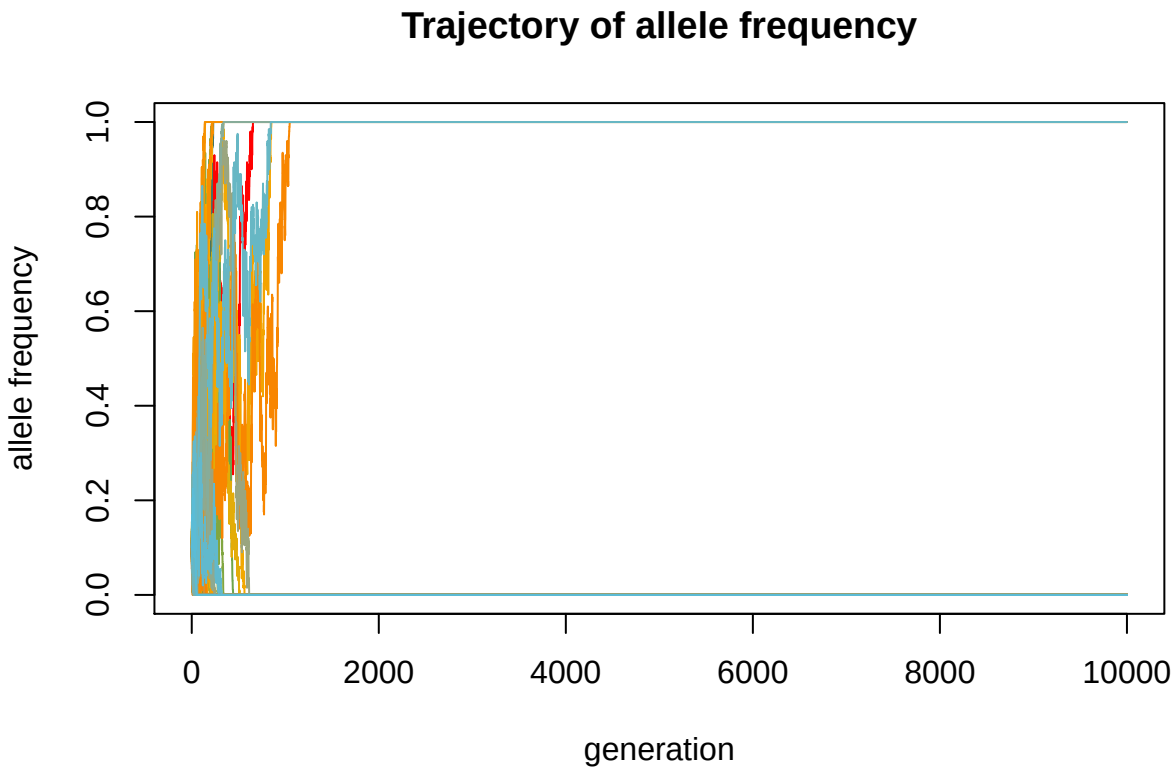
```
##
##   0   1
## 893 107
```

~ 900 of the 1000 SNPs were lost from the population; the remaining ~100 reached fixation within the population.

Does this value agree with the theoretical prediction for the probability of fixation of a neutral allele?
The result above agrees with the expectation that the $p(fixation) = p_i$.

d) Make a plot of the allele frequency trajectories for 100 of the SNPs.

```
cols <- wes_palette("Darjeeling", n = 100, "continuous")
matplot(freqs[,1:100], type = "l", lty = 1, col = cols, main = "Trajectory of allele frequency",
        ylab = "allele frequency", xlab = "generation")
```



e) Repeat the simulation, but this time set p=0.6

```
freqs <- gene_frequencies(num_generations = 10000, init_freq = 0.6, num_genes = 1000, pop_size = 100)

# How many of the 1000 SNP are at frequency 0/1 at the end of the simulation (in generation 10000)?
table(freqs[nrow(freqs),])
```

```
##
##   0   1
## 404 596
```

About 600 of the SNPs went to fixation; the remaining ~400 were lost from the population.

f) Does this value agree with the theoretical prediction for the probability of fixation of a neutral allele?
Yes- the probability of fixation of a neutral allele is equal to its initial frequency in the population, which in this case is 0.6.

Question 4

Let's look at the effect of the population size on patterns of genetic drift. Repeat the simulation, but this time, set N=10, 500 and 1000. Keep the other parameters the same (p=0.1; L=1000; T=10000). Again, N is the number of diploids.

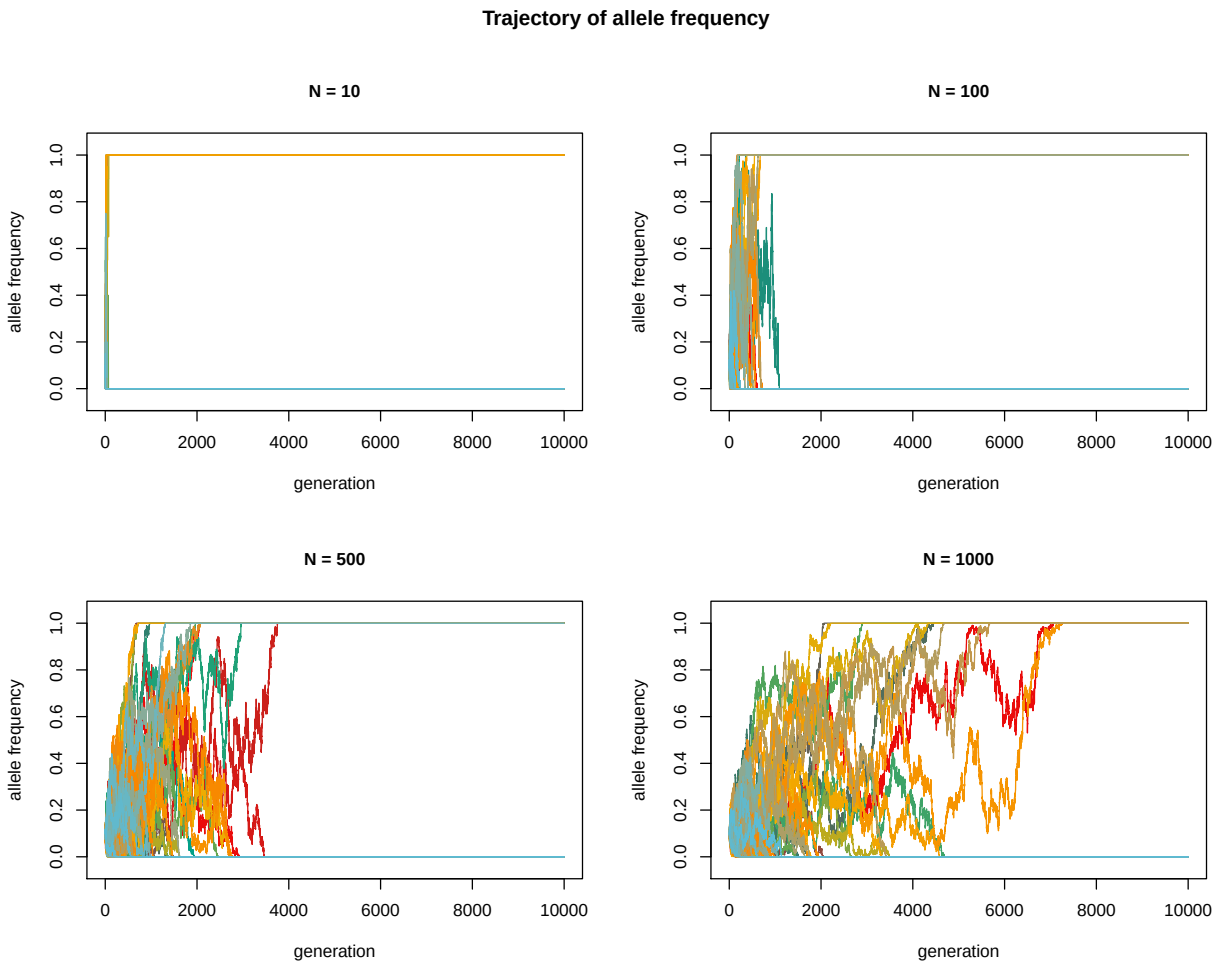
```
pop_sizes <- c(10, 100, 500, 1000)
frequencies_by_size <- lapply(pop_sizes, function(x)
  gene_frequencies(num_generations = 10000, init_freq = 0.1, num_genes = 1000, pop_size = x))

str(frequencies_by_size)
```

```
## List of 4
##  $ : num [1:10000, 1:1000] 0.1 0 0 0 0 0 0 0 0 0 ...
##  $ : num [1:10000, 1:1000] 0.1 0.105 0.09 0.06 0.08 0.095 0.1 0.14 0.1 0.075 ...
##  $ : num [1:10000, 1:1000] 0.1 0.106 0.107 0.113 0.133 0.126 0.118 0.125 0.146 0.137 ...
##  $ : num [1:10000, 1:1000] 0.1 0.108 0.098 0.112 0.12 ...
```

```
names(frequencies_by_size) <- pop_sizes
cols <- wes_palette("Darjeeling", n = 100, "continuous")
par(mfrow = c(2,2), oma = c(0,0,3,0))

for(i in 1:4){
  plot(frequencies_by_size[[i]][,1], type = "l", ylim = c(-0.05, 1.05), xlab = "generation",
       ylab = "allele frequency", main = paste("N =", pop_sizes[i]), cex.main = 1, col  = cols[1])
  sapply((2:100), function(x) lines(frequencies_by_size[[i]][,x], col = cols[x]))
}
title("Trajectory of allele frequency", outer = T)
```



c) Based on examination of the plots, how does the population size affect allele frequency change?

Alleles seem to become fixed or extinct in a small population much faster than in a large population.

d) For each population size, in what proportion of simulation replicates did the derived allele become fixed by the end of the simulation?

```
prop_fixed <- function(x) {
  vv <- x[nrow(x), ]
  rr <- sum(vv == 1)/ncol(x)
  return(rr)
}
sapply(frequencies_by_size, prop_fixed)
```

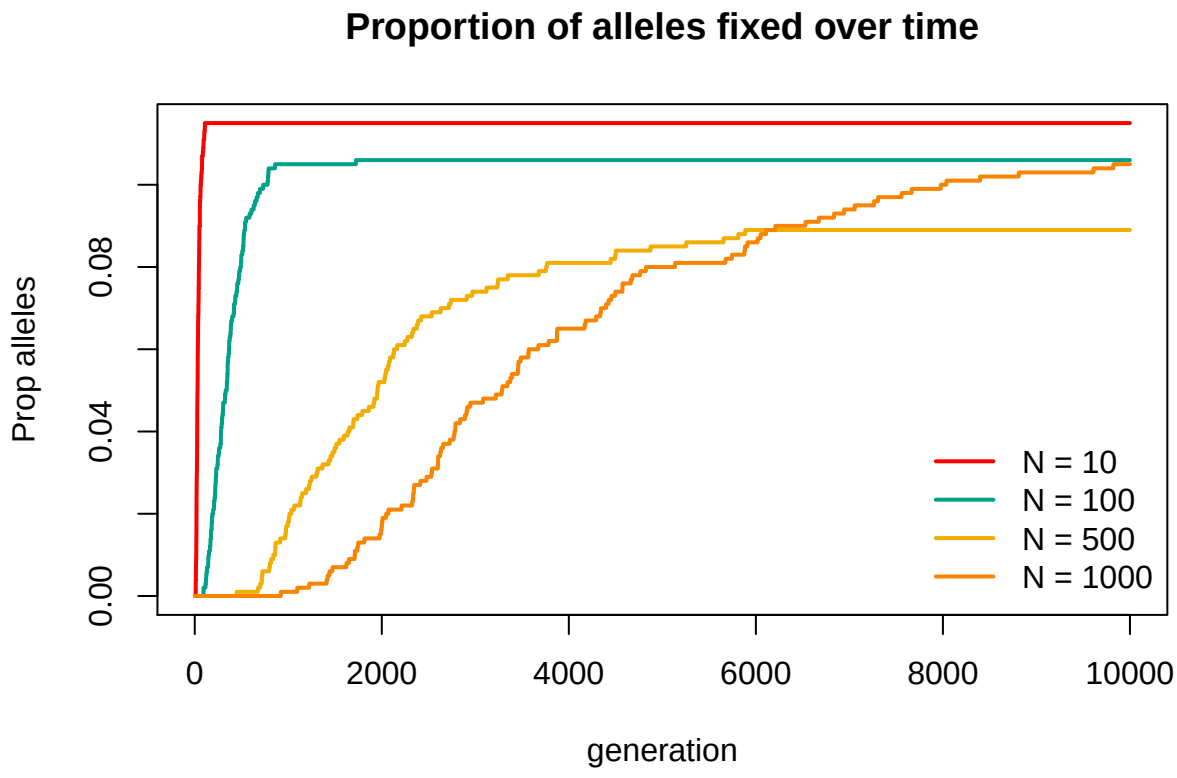
```
##    10   100   500  1000
## 0.115 0.106 0.089 0.105
```

As seen above, the proportion of replicates that went to fixation was about the expected value of 0.1 regardless of the population size. Another interesting way to explore this question would be plot the proportion of fixed sites as a function of time- I would expect that alleles go to fixation much faster in the small populations:

```r
prop_fixed_by_time <- function(x){

  vv <- numeric(nrow(x))
  for (i in 1:nrow(x)) {
    qq <- x[i, ]
    vv[i] <- sum(qq == 1)/ncol(x)
  }
  return(vv)
}

fixed_alleles <- sapply(frequencies_by_size, prop_fixed_by_time)
cols <- wes_palette("Darjeeling", n = 4)
par(mfrow = c(1,1))
plot(fixed_alleles[,1], type = "l", col = cols[1], lwd = 2,
     main = "Proportion of alleles fixed over time", ylab = "Prop alleles", xlab = "generation")
for(i in 2:4) {lines(fixed_alleles[,i], col = cols[i], lwd = 2)}
legend("bottomright", lty = 1, col = cols, legend = c("N = 10", "N = 100", "N = 500", "N = 1000"),
       bty = "n", lwd = 2)
```



e) How does this probability of fixation estimated from the simulations match with the theoretical prediction?

Even though the expected value of p is equal regardless of the population size, theory tells us that the variance in p does indeed depend on N. In other words,

$$E[p_{t+1}] = p_t$$
$$V[p_{t+1}] = \frac{p_t * (1 - p_t)}{2N}$$

The equation for variance shows that the variance of p_t decreases with increasing population size, so we expect the allele frequencies to show more drastic shifts in small populations than in large ones.

Exploring the exponential distribution

```
# E[X] in exponential distribution == 1/rate
rate = c(5,10)
1/rate; sapply(rate, function(x) mean(rexp(n = 100, rate = x)))
```

```
## [1] 0.2 0.1
```

```
## [1] 0.19529941 0.08541866
```

Question 5

What is the rate of coalescence for a sample size of 2 chromosomes in a population of size 2N chromosomes? This is akin to the probability that 2 chromosomes find a common ancestor in the previous generation.

The rate of coalescence for two chromosomes in a population of size 2N is $1/2N$. If we approach this from the other direction, we can think that coalescent time is exponentially distributed with a mean of $2N$. Since the mean of an exponential distribution is equal to the inverse of its rate, we can infer that the rate of this distribution is $1/2N$.

Question 6

Perform 10,000 simulations of coalescent times for a sample size of 2 chromosomes from this population of size 2N=20,000.

a) What is the average time to the most recent common ancestor (TMRCA) in your simulations? b) What is the theoretical expectation?

```
reps <- 10000
N <- 10000

rate <- 1/((2*N))
coal_times <- rexp(reps, rate)

2*N ; mean(coal_times)
```

```
## [1] 20000
```

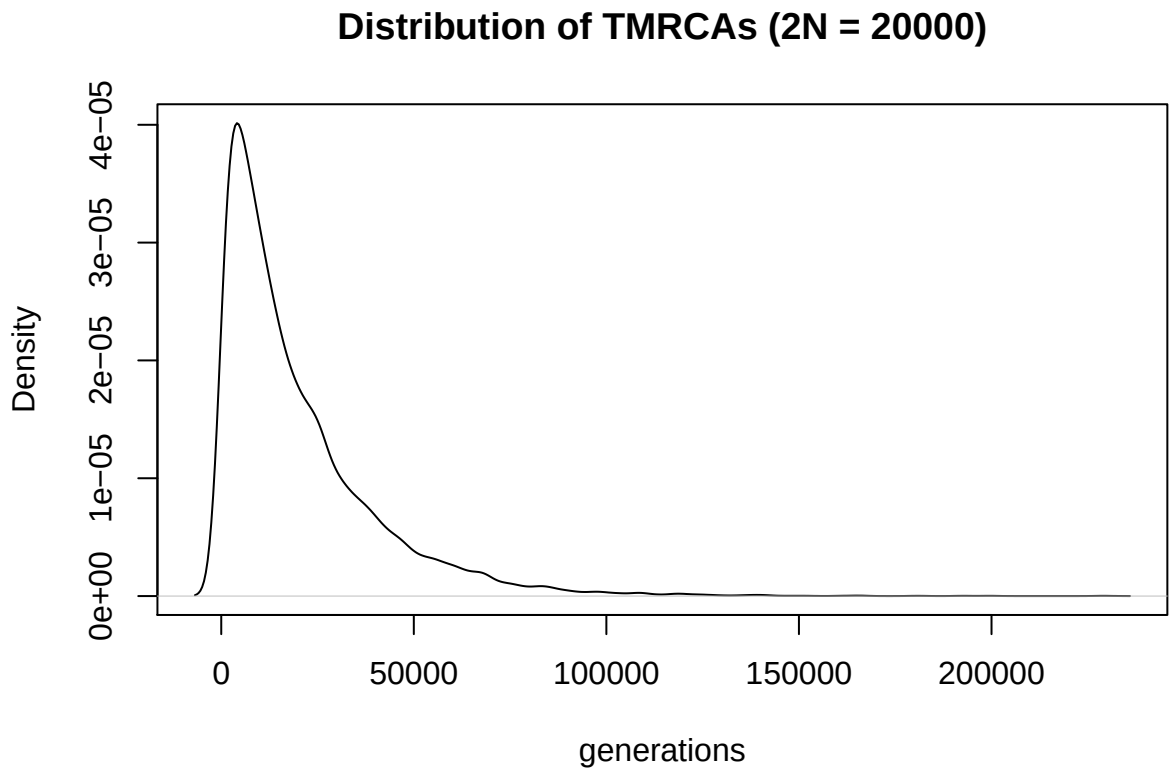
```
## [1] 19840.22
```

c) How do the two values compare?

As shown above, the average time to MRCA in the simulations is ~20000. The theoretical expectation for coalescence time is $2N = 20000$ generations.

d) Make a density plot of your simulated coalescent times.

```
plot(density(coal_times), main = "Distribution of TMRCAs (2N = 20000)", xlab = "generations")
```



e) What is the standard deviation of the coalescent times?

```
sd(coal_times)
```

```
## [1] 20254.82
```

The standard deviation of this is also equal to ~20000. This makes sense, as the sd of an exponential distribution is, like its mean, equal to $1/rate$.

Question 7

Repeat the simulation described in question 6, but this time set 2N=2,000. Answer parts a-e from question 6 for this new set of simulations.

```
reps <- 10000
N <- 1000

rate <- 1/((2*N))
coal_times <- rexp(reps, rate)

# Expected average; true average; standard deviation
2*N ; mean(coal_times); sd(coal_times)
```

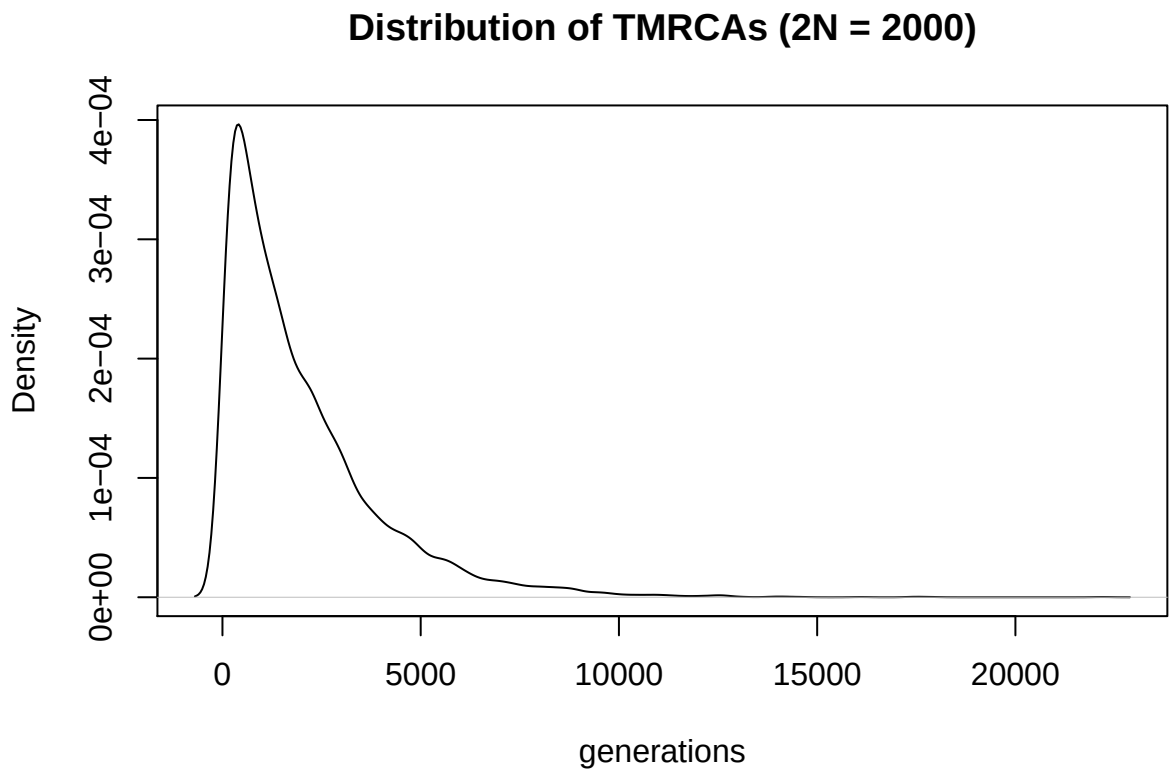
```
## [1] 2000
```

```
## [1] 1978.377
```

```
## [1] 1963.365
```

```
# Mean and SD match expecations of 2N


# Density plot:
plot(density(coal_times), main = "Distribution of TMRCAs (2N = 2000)",
     xlab = "generations")
```



b) How does the average TMRCA from the simulations in question 6 (where 2N=20,000) compare to the average TMRCA from the simulations in question 7 (2N=2,000)

The TMRCA in the 2N = 2000 population is ~2000 generations, while it is ~20000 generations in the 2N = 20000 population.

c) What can you conclude about how the population size affects the expected coalescent time?

This exercise shows that the expected time to coalescent grows linearly with the number of chromosomes in the population.

Question 8

Let's add mutations to the genealogies you've simulated and then look at what these models predict in terms of the genetic variation you might see in a sample. Set mu = 1 x 10-4

a) Add mutations to the genealogies (really the coalescent times) that you simulated in question 6 (N=20,000)

```
# Regenerate coal_times with 2N = 20000
reps <- 10000
N <- 10000

rate <- 1/((2*N))
coal_times <- rexp(reps, rate)

# mean(coal_times)

mu <- 1*10^-4

net_mutation_rate <- 2*mu*coal_times
num_snps <- rpois(length(coal_times), net_mutation_rate)
```

b) What is the average number of SNPs per geneology?

```
mean(num_snps)
```

```
## [1] 3.972
```

c) Recall that $\theta = 4N\mu$. Note, the factor-of-2-confusion: N refers to the number of diploids. So, θ is twice the number of chromosomes multiplied by mu. What is θ predicted to be in this example? d) How does the value of θ compare to the average number of SNPs seen in the simulations?

```
# Recall mu and N
mu; N
```

```
## [1] 1e-04
```

```
## [1] 10000
```

```r
# Twice the number of chromosomes multiplied by mu
theta <- 4*N*mu

# Print out expected and true SNP count
theta; mean(num_snps)
```

```
## [1] 4
```

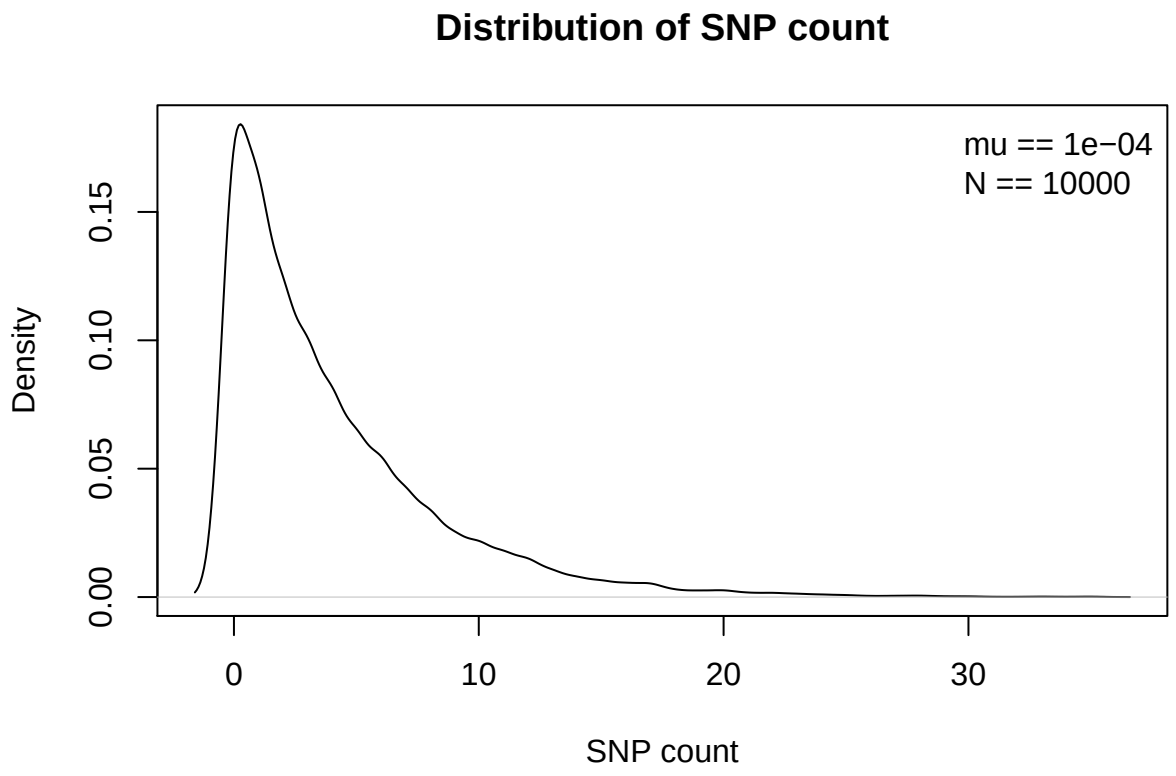
```
## [1] 3.972
```

e) Pretty neat, huh?

Sure is!

f) Make a density plot of the number of SNPs per simulation replicate.

```r
plot(density(num_snps), main = "Distribution of SNP count",
     xlab = "SNP count")
legend("topright", legend = c(bquote(mu == .(mu)), bquote(N == .(N))), bty = "n")
```



Question 10 (sic)

Repeat all the parts in question 8, but this time, set N = 2,000.

```
# Regenerate coal_times with 2N = 2000
reps <- 10000
N <- 1000

rate <- 1/((2*N))
coal_times_small <- rexp(reps, rate)

mean(coal_times)
```

```
## [1] 19893.35
```

```
mu <- 1*10^-4
num_snps_small <- numeric(reps)
for (i in 1:length(coal_times_small)) {
  net_mutation_rate <- 2*mu*coal_times_small[i]
  num_snps_small[i] <- rpois(1, net_mutation_rate)
}

mean(num_snps_small)
```

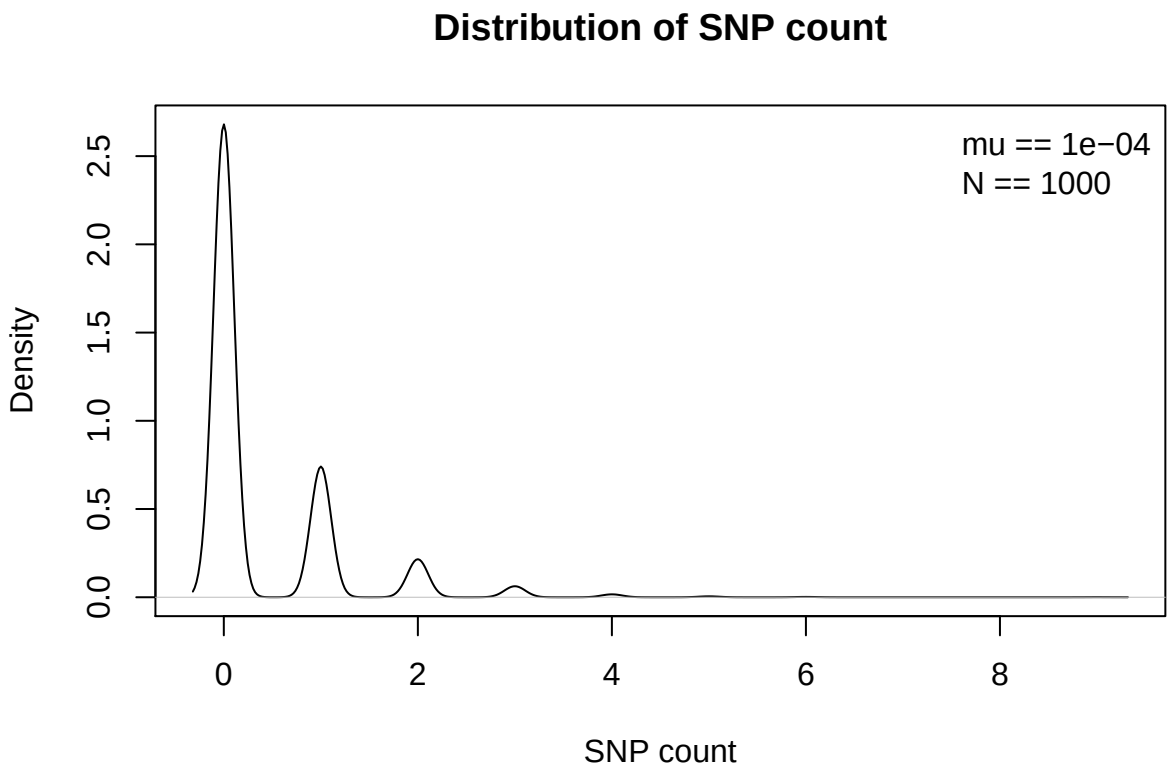
```
## [1] 0.3926
```

```
theta <- 4*N*mu; theta; mean(num_snps_small)
```

```
## [1] 0.4
```

```
## [1] 0.3926
```

```
plot(density(num_snps_small), main = "Distribution of SNP count",
     xlab = "SNP count")
legend("topright", legend = c(bquote(mu == .(mu)), bquote(N == .(N))), bty = "n")
```

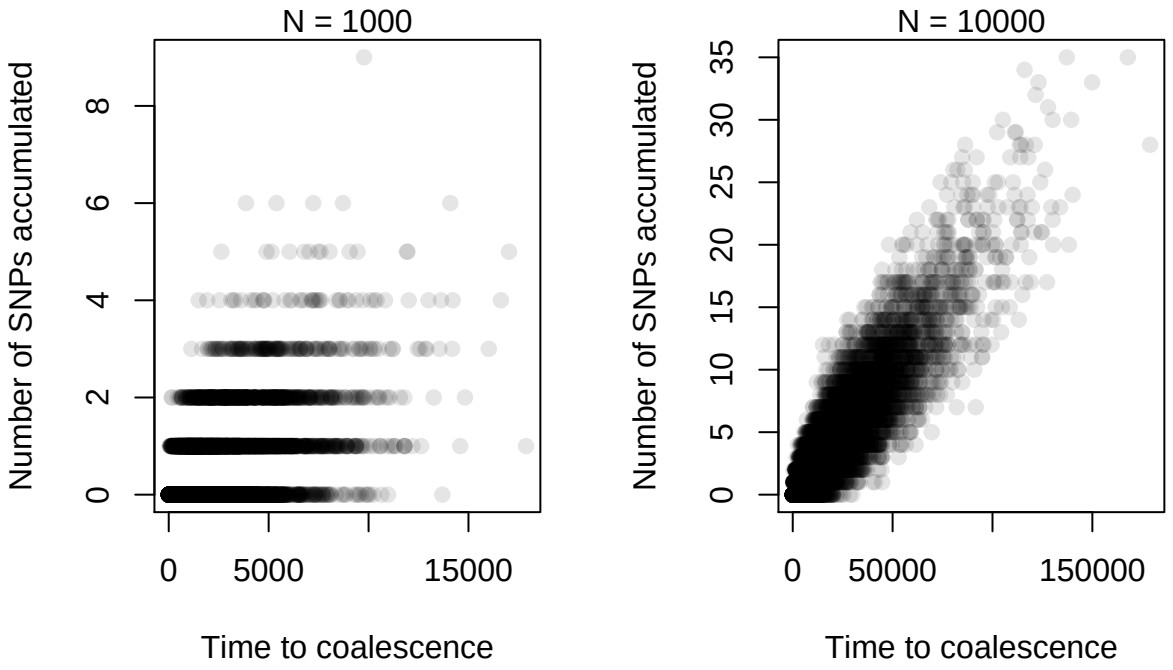


Again, $4N\mu$ = Num SNPs observed. As the N is smaller here than in the previous question, the lineages on average have fewer SNPs after the same number of generations.

Question 11

a) Make a scatter plot of the number of SNPs seen in each simulation replicate vs. the TMRCA for that simulation replicate

```
par(mfrow = c(1,2), oma = c(0,0,1,0))
color <- rgb(0,0,0,alpha=0.1)
plot(x = coal_times_small, y = num_snps_small, xlab = "Time to coalescence",
     ylab = "Number of SNPs accumulated", pch = 19, col = color)
mtext(side = 3, text = "N = 1000")
plot(x = coal_times, y = num_snps, xlab = "Time to coalescence",
     ylab = "Number of SNPs accumulated", pch = 19, col = color)
mtext(side = 3, text = "N = 10000")
title("Number of SNPs vs generations", outer = T)
```



b) How are these variables related to each other?

Broadly, it appears that at sufficiently large population sizes, the number of SNPs is significantly correlated to the TMRCA. The linear relationship seems to get stronger with population size- below are the R^2 values for the two population size scenarios (I understand that I am breaking assumptions of normally distributed data, etc., given that this is count data... but this seems to be a decent first-cut?).

```
rsq_small_pop <- summary(lm(coal_times_small~num_snps_small))$adj.r.squared
rsq_big_pop <- summary(lm(coal_times~num_snps))$adj.r.squared

round(rsq_small_pop, 4); round(rsq_big_pop, 4)
```

```
## [1] 0.2844
```

```
## [1] 0.7958
```

e) Does the number of SNPs tell you anything about the TMRCA?

If we have an estimate of SNP count in a population and a reliable model to relate SNPs to TMRCAs (having a good model requires a good estimate of mutation rate, which may not be readily available ubiquitously), we could reconstruct the demographic history of a population.

Question 12

Imagine that you sequenced a 10kb region of DNA from single individual (a diploid species; i.e. you have a sample of 2 chromosomes). You observed 10 SNPs in that 10kb interval. Note

that independent evidence suggests a mutation rate of 1 x 10-8 per base pair/generation. (Hint: this is very similar to what you've simulated)

a) Use coalescent simulations to evaluate whether a model with N=10,000 is compatible with your observed data? Put another way, what proportion of simulation replicates from this model have >= 10 SNPs?

```
N <- 10000
mu <- 1*10^-8 # Note this is per BP- earlier was chromosome-wide rate
bps <- 10000

# Step 1: simulate times to coalescent ---------
reps <- 10000
rate <- 1/(2*N)

coal_times_12 <- rexp(reps, rate)
# Mean of simulation; Expected
mean(coal_times_12); 2*N
```

```
## [1] 19982.31
```

```
## [1] 20000
```

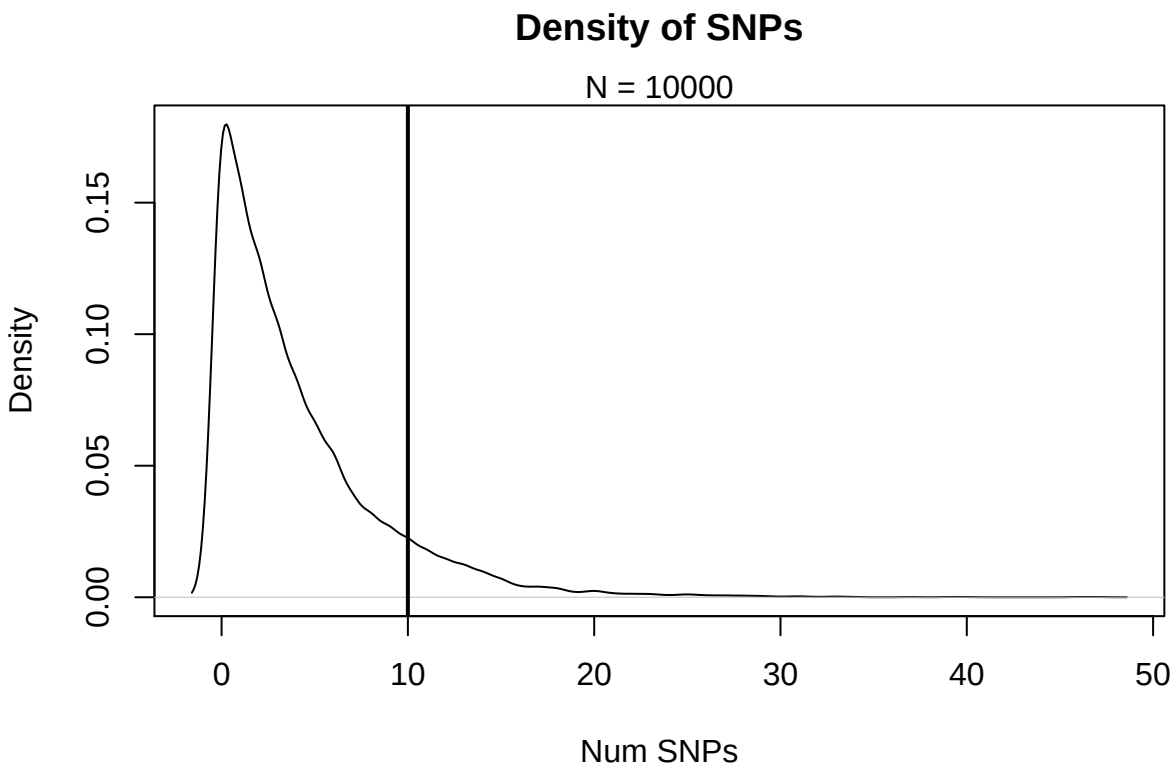
```
# SD of simulation; Expected
# sd(coal_times_12); 2*N

# Step 2: simulate SNPs ------------


net_mutation_rates <- 2*bps*coal_times_12*mu
num_snps_12 <- rpois(reps, net_mutation_rates)

# mean(num_snps_12); 4*N*mu*bps

# Plot
par(mfrow = c(1,1))
plot(density(num_snps_12), main = "Density of SNPs", xlab = "Num SNPs")
mtext(side = 3, text = "N = 10000")
abline(v = 10, lwd = 2)
```




```
# Proportion of simulations compatible
sum(num_snps_12 >= 10)/length(num_snps_12)
```

```
## [1] 0.1099
```

When $N = 10000$, ~10% of the simulations yield >= 10 SNPs.

b) Is the model with N = 1,000 compatible with your observed data? What is the proportion of simulation replicates that have >= 10 SNPs?

```
N <- 1000
mu <- 1*10^-8
bps <- 10000

# Step 1: simulate times to coalescent ---------
reps <- 10000
rate <- 1/(2*N)

coal_times_12 <- rexp(reps, rate)
# Mean of simulation; Expected
mean(coal_times_12); 2*N
```

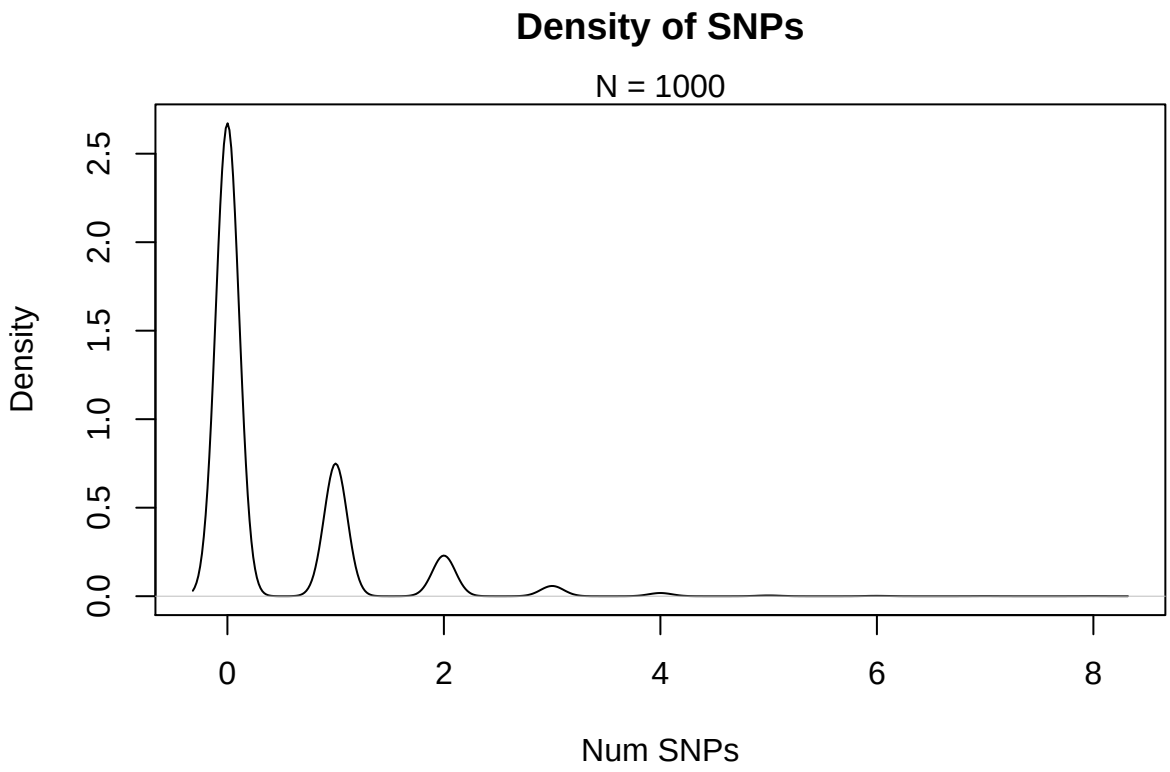
```
## [1] 1973.996
```

```
## [1] 2000
```

```r
# Step 2: simulate SNPs ------------

net_mutation_rates <- 2*bps*mu*coal_times_12
num_snps_12 <- rpois(reps, net_mutation_rates)

# Plot
par(mfrow = c(1,1))
plot(density(num_snps_12), main = "Density of SNPs", xlab = "Num SNPs")
mtext(side = 3, text = "N = 1000")
abline(v = 10, lwd = 2)
```




```r
# Proportion of simulations compatible
sum(num_snps_12 >= 10)/length(num_snps_12)
```

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

In 10000 replicate lineages of $N = 1000$, no lineage accumulated >= 10 SNPs. The simple model presented here is not compatible with the proposed population size.