HW1

Maria Firulyova

26/11/2020

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(colorRamps)
library(reshape2)
```

1

Let's assume you have a locus with two alleles, A and a, and the following genotype frequencies: 55 AA : 35 Aa : 10 aa. What is the frequency of each of the alleles?

p = (55 * 2 + 35) / 200 = 0.725

q = 1 - 0.725 = 0.275

Answer: freq(A) = 0.725, freq(a) = 0.275

2

In the example (1), does the Hardy-Weinberg equilibrium genotype ratio hold for this locus? Estimate the statistical significance

X-square estimation:

```
obs <- c(55, 35, 10)
exp <- c(100 * 0.725^2, 100 * 2 * 0.725 * 0.275, 100 * 0.275^2)
chisq.test(obs, exp)
```

```
##
##  Pearson's Chi-squared test
##
## data:  obs and exp
## X-squared = 6, df = 4, p-value = 0.1991
```

Since p-value > 0.05, I conclude that the population under H-W equilibrium.

3

Let's turn to our example R code we used in class (drift_task.R). Let's define probability of allele elimination as the fraction of the simulated populations (iterations) in which the allele frequency reached 0. Let the population size be 50. What is the estimated probability of eliminating an allele that has a starting frequency of (a) 0.5; (b) 0.1 after 20 generations? Why is there a difference? (You should use an iteration number between 100 and 1000)

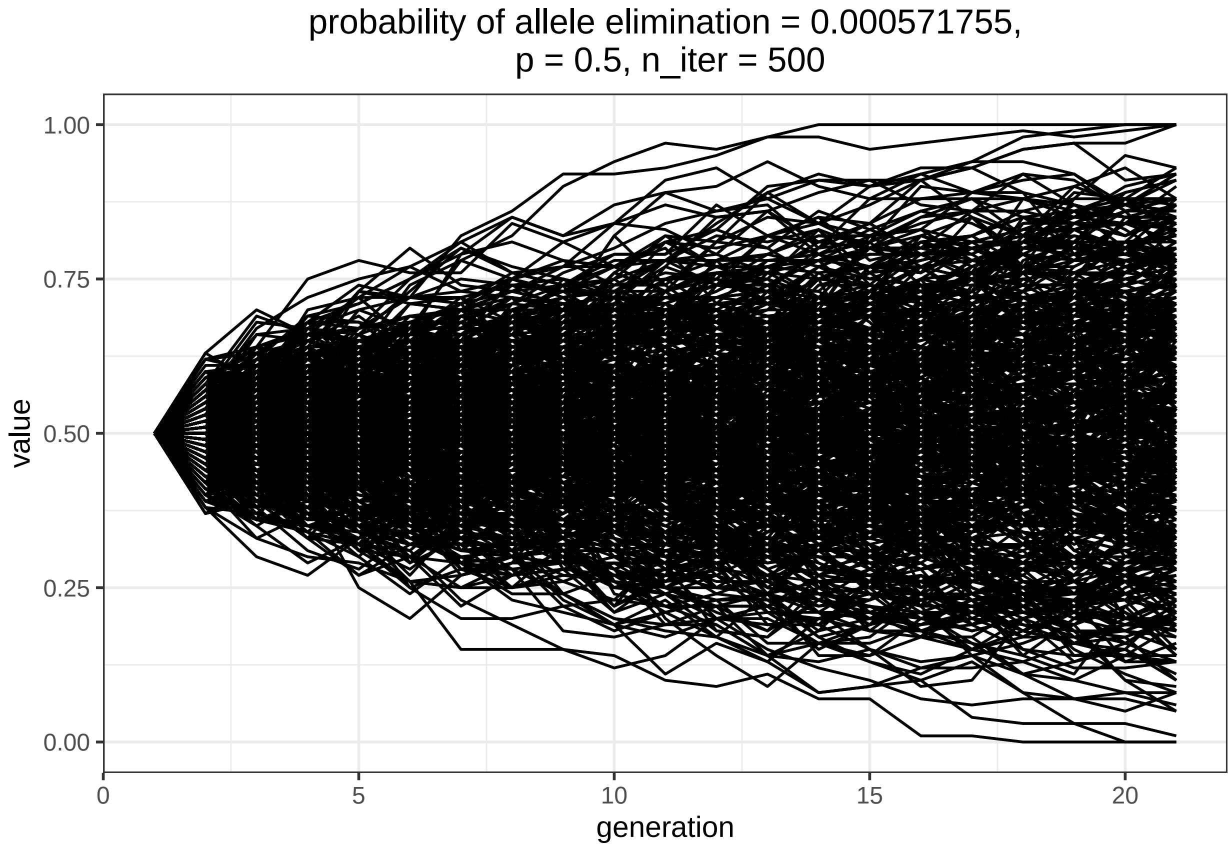
```
wright_fisher_sim <- function(p, N, gens) {
  answ = c(p)
  for (i in 1:gens){
    p_new = rbinom(1, 2*N, answ[length(answ)]) / (2*N)
    answ = c(answ, p_new)
  }
  return(answ)
}
```

```
get_elim_prob <- function(p, n_gen=20, n_iter=5e2, N=50) {
  set.seed(42)
  traj = sapply(1:n_iter, function(x) wright_fisher_sim(p, N, n_gen))
  tr_df = as.data.frame(traj)
  tr_df$generation = 1:nrow(tr_df)
  plot_df = melt(tr_df, id.vars='generation')
  # here I estimate the fraction of the populations in which the AF reached 0
  elim_prob <- plot_df %>% filter(value == 0) %>% nrow() /
    plot_df %>% filter(value != 0) %>% nrow()
  ggplot(plot_df, aes(x=generation, y=value, group=variable))+
    geom_line()+
    scale_y_continuous(limits=c(0, 1))+
    ggtitle(
      sprintf('probability of allele elimination = %g,\n p = %g, n_iter = %g',
              elim_prob, p, n_iter))+
    theme_bw()+
    theme(plot.title = element_text(hjust = 0.5))
}
```

(a)

p (the initial allele frequency) = 0.5

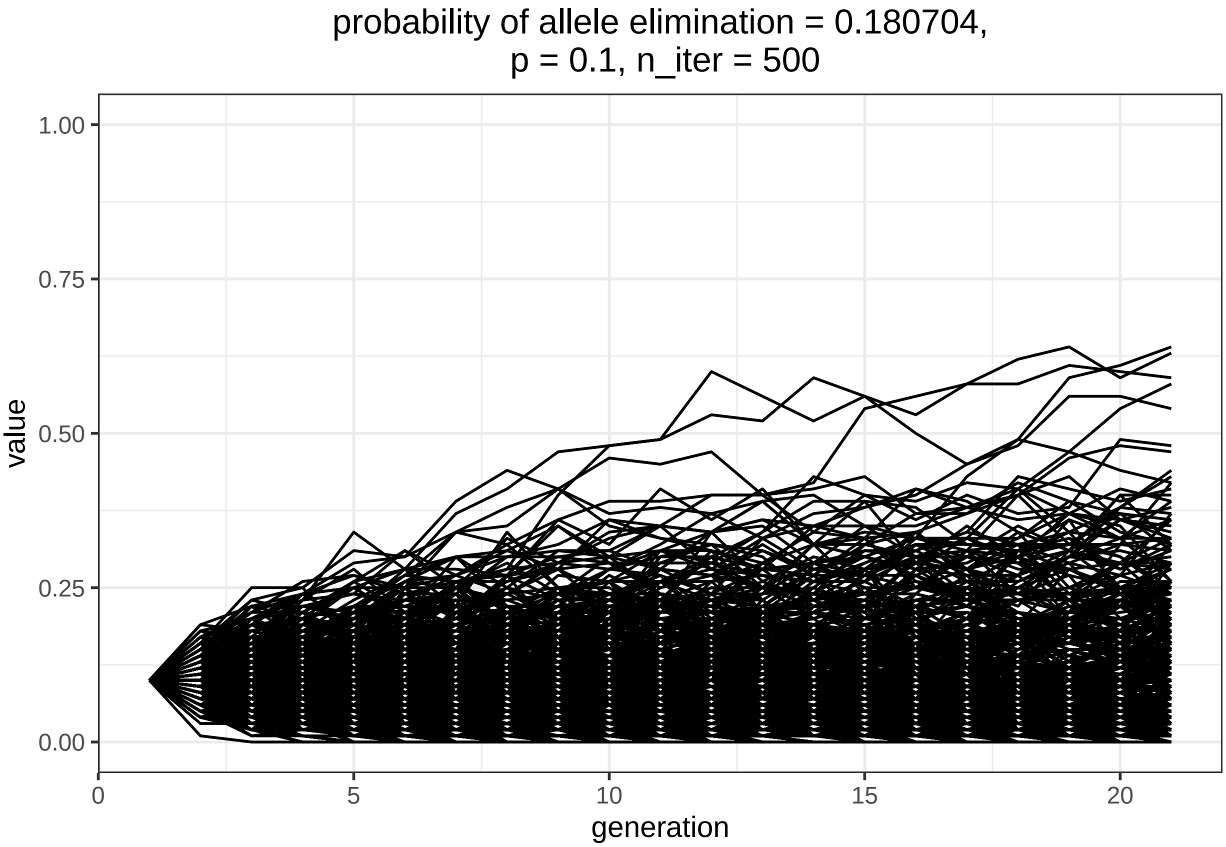
```
get_elim_prob(p = 0.5)
```



(b)

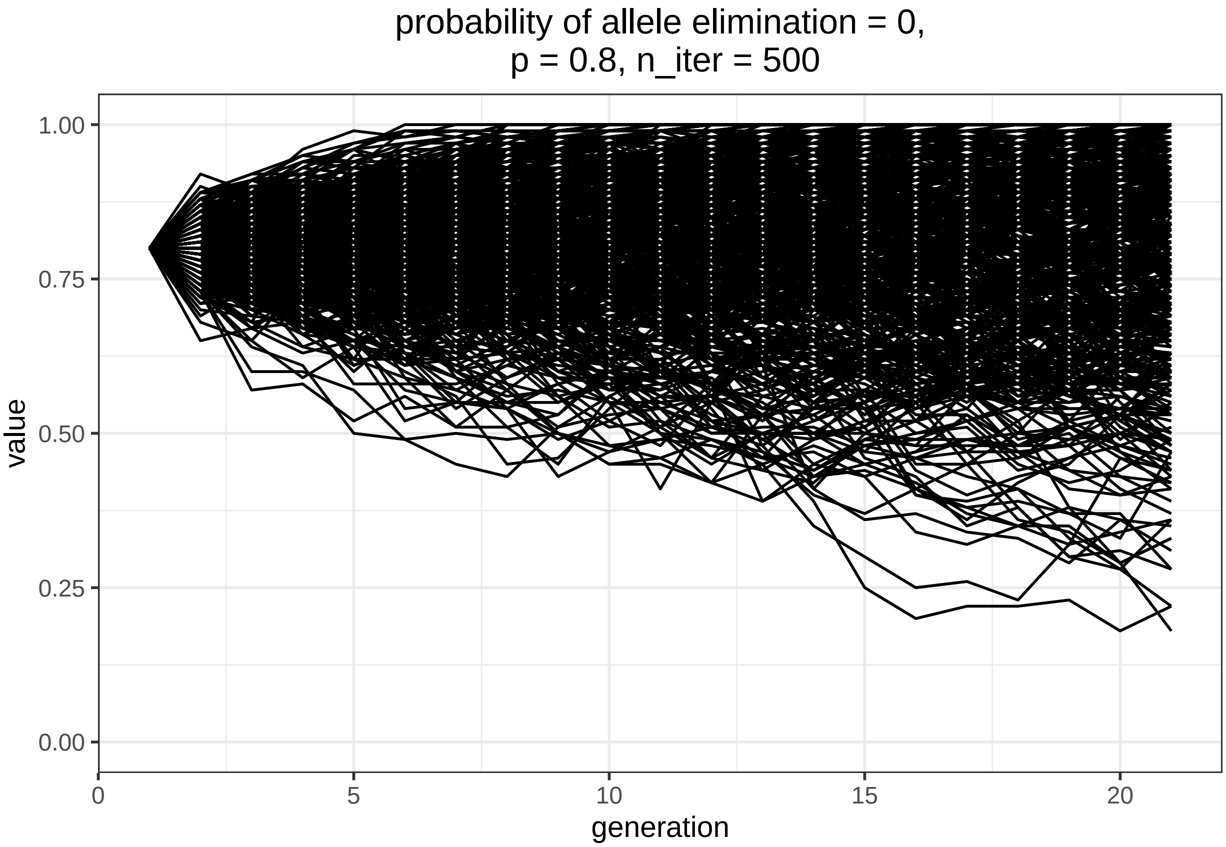
p (the initial allele frequency) = 0.1

```
get_elim_prob(p = 0.1)
```



The probability of allele elimination is higher when the initial allele frequency is lower because if we the population size is supposed to be constant for these examples (N = 50), the probability to obtain the target allele is related to the frequency of this allele. If at the initial state we have a low level of AF (e.g., AF < 50%), then the elimination of this allele in the next generation is something which we expect. We can easily see that if the p = 0.8, the fixation of the allele is more probable than the elimination.

```
get_elim_prob(p = 0.8)
```



4

Compare two following cases:

1) pp fitness = 1, pq fitness = 0.95, qq fitness = 0.9:
2) pp fitness = 1, pq fitness = 1, qq fitness = 0.9.

Run the simulations with the following parameters: N = 100000, number of iterations = 100, number of generations - 100, starting allele frequency - 0.5.

Q1: Draw a histogram of the allele frequency distribution at the final (100th) generation.

Q2: What is the mean value of an allele frequency at the end of the simulation in each case? **(see the plot titles)

Q3: Is there a difference between the two cases, and if so, why is there one?

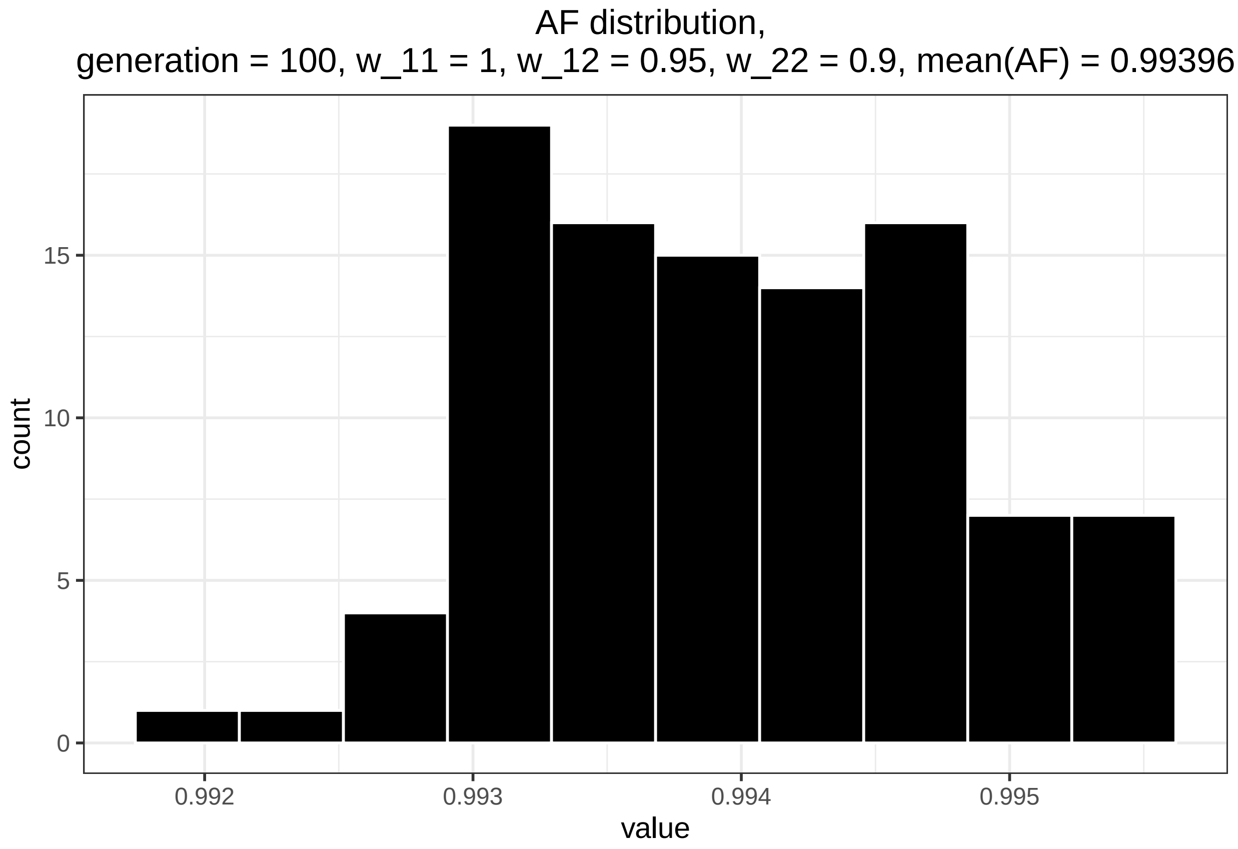
```
wright_fisher_selective <- function(p, N, gens, w_11, w_12, w_22) {
  ac = c(p * 2 * N)
  for (x in 1:gens) {
    total_fitness = p * p * w_11 + 2 * p * (1 - p) * w_12 + ((1 - p) ^ 2) * w_22
    p_gf = (p * p * w_11 + p * (1 - p) * w_12)/total_fitness
    p = rbinom(1, 2 * N, p_gf)/(2 * N)
    ac = c(ac, p * 2 * N)
  }
  return(ac)
}
```

```
get_af_hist <- function(w_11, w_12, w_22, n_iter=1e2, p=0.5, N=1e5, n_gen=1e2) {
  set.seed(42)
  traj = sapply(1:n_iter, function(x) wright_fisher_selective(p, N, n_gen, w_11, w_12, w_22))
  tr_df = as.data.frame(traj)
  tr_df$generation = 1:nrow(tr_df)
  plot_df = melt(tr_df, id.vars='generation') %>% mutate(value = value / (2 * N))
  ggplot(plot_df %>% filter(generation == 100), aes(x = value))+
    geom_histogram(col='white', fill='black', bins=10)+
    ggtitle(
      sprintf('AF distribution, \ngeneration = 100, w_11 = %g, w_12 = %g, w_22 = %g, mean(AF) = %g',
               w_11, w_12, w_22, mean((plot_df %>% filter(generation == 100))$value)))+
    theme_bw()+
    theme(plot.title = element_text(hjust = 0.5))
}
```

(a)

pp fitness = 1, pq fitness = 0.95, qq fitness = 0.9

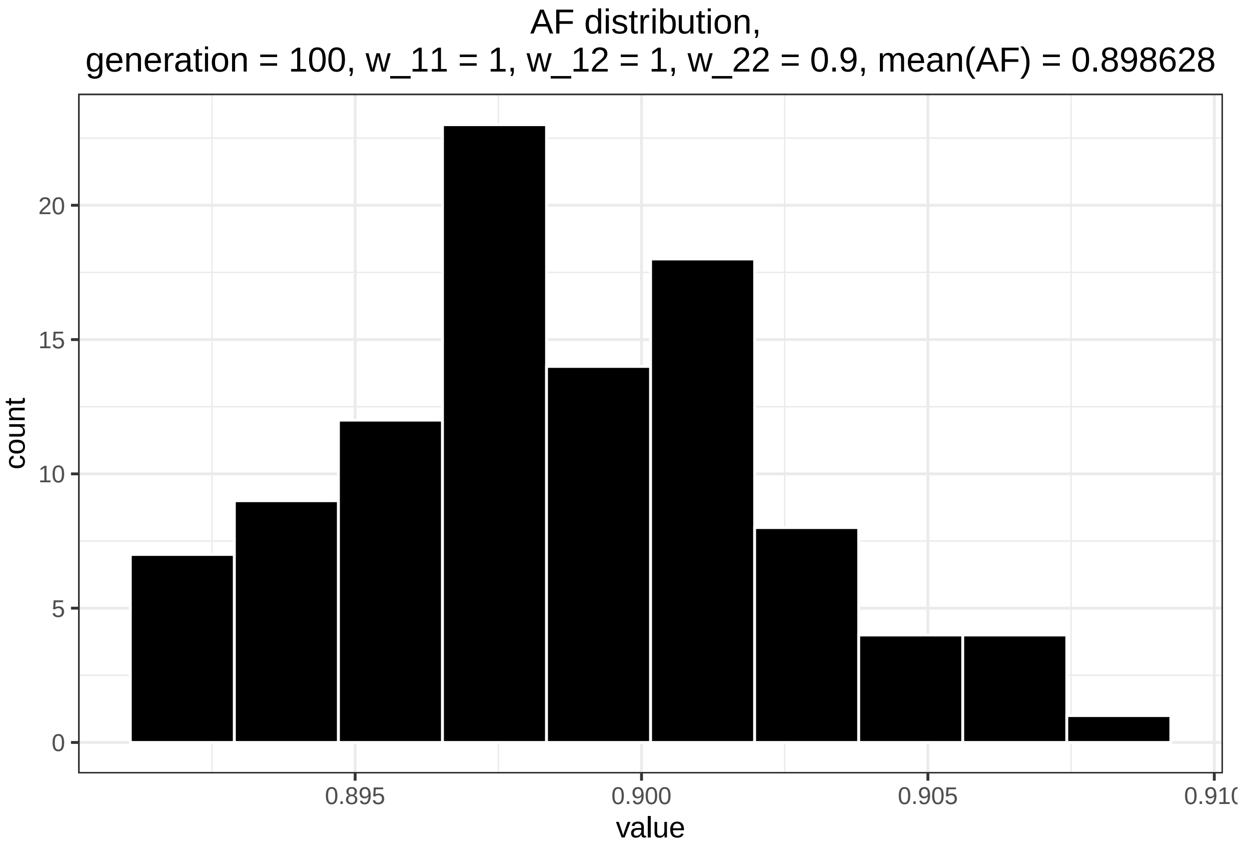
```
get_af_hist(1, 0.95, 0.9)
```



(b)

pp fitness = 1, pq fitness = 1, qq fitness = 0.9

```
get_af_hist(1, 1, 0.9)
```



The mean(AF) is decreased when we increased the pq fitness value. Since the fitness of a heterozygous genotype is high, the heterozygous genotype is more than in the case with lower pq fitness value, prioritized, so the AF is decreased.

5

Now let's investigate balancing selection.

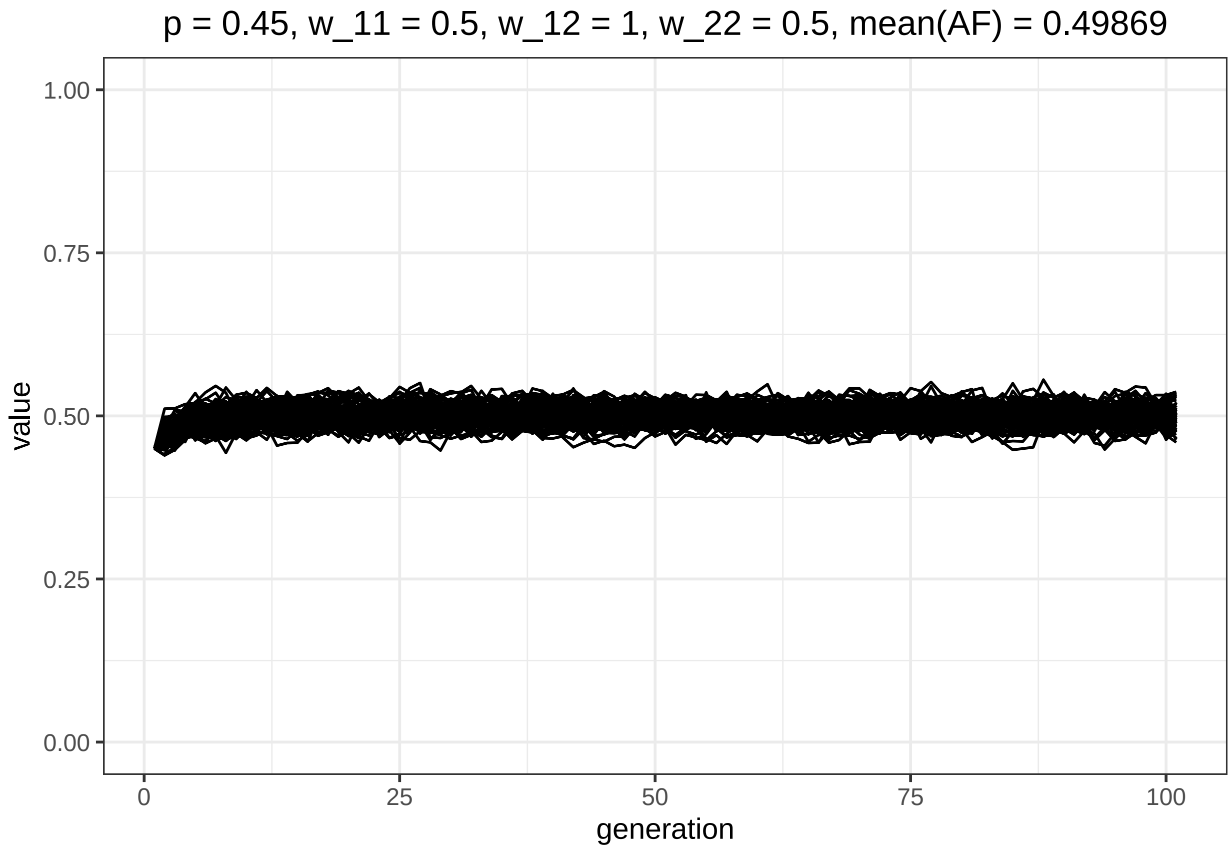
Q1: Draw the evolutionary trajectories for each combination of fitness scores.

```
get_af_hist <- function(w_11, w_12, w_22, n_iter=1e2, p=0.45, N=1e3, n_gen=1e2) {
  set.seed(42)
  traj = sapply(1:n_iter, function(x) wright_fisher_selective(p, N, n_gen, w_11, w_12, w_22))
  tr_df = as.data.frame(traj)
  tr_df$generation = 1:nrow(tr_df)
  plot_df = melt(tr_df, id.vars='generation') %>% mutate(value = value / (2 * N))
  ggplot(plot_df, aes(x=generation, y=value, group=variable))+
    geom_line()+
    scale_y_continuous(limits=c(0, 1))+
    ggtitle(sprintf('p = %g, w_11 = %g, w_12 = %g, w_22 = %g, mean(AF) = %g', p, w_11, w_12, w_22,
                    mean((plot_df %>% filter(generation == 100))$value)))+
    theme_bw()+
    theme(plot.title = element_text(hjust = 0.5))
}
```

(a)

w(pp) = 0.5, w(pq) = 1, w(qq) = 0.5, p = 0.45

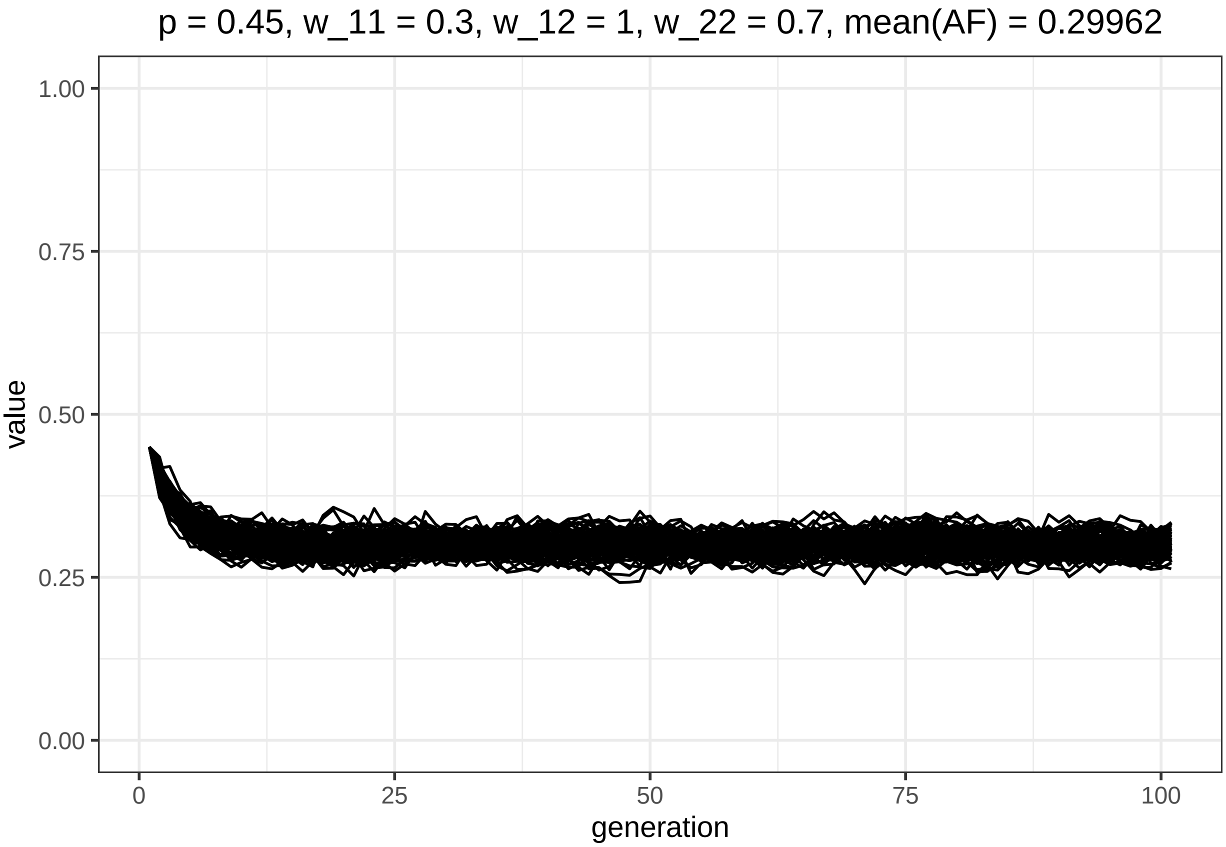
```r
get_af_hist(0.5, 1, 0.5)
```



(b)

w(pp) = 0.3, w(pq) = 1, w(qq) = 0.7, p = 0.45

```r
get_af_hist(0.3, 1, 0.7)
```



Q2: Does the outcome of the two simulations differ? Why?

For the first case, the fitness scores of both dominant and recessive homozygous are equal to each other, which means that the trajectory of allele frequencies will seek to the p = 0.5 and fluctuate around this AF. For the second one, the recessive homozygous genotype has more fitness score than dominant homozygous genotype, which lead to the decreasing of AF.

Q3: Calculate the mean total fitness of the population (weighted sum of genotype frequencies) at the end of the simulation for each combination of fitness scores. Do the values differ and why?

The values differ, and I guess it is connected with the previous question. For example, the w(qq) is more than w(pp) for the second case, so the recessive allele provides more advantage => mean(AF) is decreased, and vice versa.