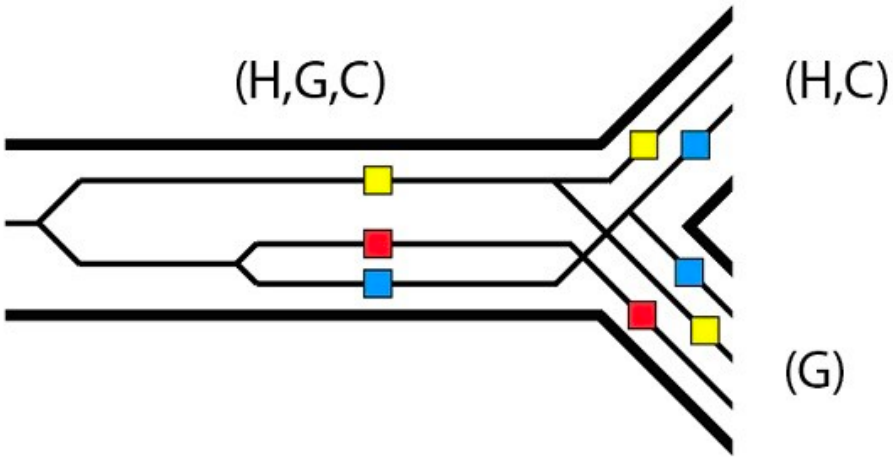


Evolution Laboratory Notebook
Biological Sciences 115L
Spring 2023

Laurence D. Mueller
Michael R. Rose
J.J. Emerson
María Rebolleda-Gómez
Tony Long

Table of Contents

Working with *Drosophila melanogaster*

Why *Drosophila*?

All the experiments in this class use the common laboratory fruit fly, *Drosophila melanogaster*. There are a number of reasons why we use fruit flies. The lab fruit fly has been used for research in evolution and genetics for the last 100 years, so we know a lot about it. The fruit fly is easy to raise in large numbers and it has a short generation time. The short generation time makes the fruit fly convenient for studying multi-generation phenomena, like evolution. As you will see in this laboratory course, useful genetic mutants of *Drosophila* and specially created lines are already available. These genetic variations allow us to do experiments that could not be done with almost any other organism. As of the first decade of the 21st Century, *Drosophila* is one of a select group of animals that has had entire genomes sequenced. This gives us a solid foundation of genomic information for specific studies of genetics and evolution.

How to Handle *Drosophila*

The life cycle of *Drosophila melanogaster* is outlined in the figure on page 5. Experiments in this laboratory involve the handling of adults only. Although it is possible to handle eggs and larvae, it is considerably more difficult than handling the adults.

Because the adults can fly, they need to be knocked out before you count, sex, or genetically type them. We knock them out using CO_2 anesthesia. Specific techniques for using CO_2 will be demonstrated in the lab, but you should always remember the following facts about CO_2. Exposing adults to excessive amounts or prolonged exposure to CO_2 can kill or severely incapacitate the adult fly. When flies are immobile on the CO_2 plates, you should carefully control the flow of carbon dioxide to keep it at the lowest level possible. One easy way to determine the lowest possible level of CO_2 is to keep turning the CO_2 down until you see the flies start to wake up. At that point you have just passed the minimum flow of CO_2 needed, and should slightly increase the gas flow. Flies should not be knocked out for more than about 10 minutes. Thus only put as many flies on your platform as you can handle in about 10 minutes, if the flies need to be alive after anesthesia. You also have to be especially careful with females that are less

than 8 hours old, because even modest amounts of CO_2 can sterilize young females.

Males and females are easily distinguished. Males have a solid black patch on the tip of their abdomen. A ring of bristles surrounds their genitals. Females lack the black patch at the tip of the abdomen, and have brown stripes across the back of their abdomens. In very young adults, less than 6 hours from emergence from the pupa, the pigments in the bodies may be very light. This makes "sexing" the adults much more difficult. Be careful in these situations.

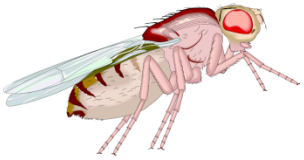
The Life Cycle of *Drosophila melanogaster*

Eggs 21 hrs at 25° C

Larvae - 4 days
three instars, duration
affected by density and temperature

Pupae - 4 days

Adults- 30-80 days
longevity affected by many factors including genetics, males and females show a gradual decline in reproductive capacity with age.



Random Genetic Drift

Introduction

Gene, or "allele," frequencies fluctuate in experimental populations due to random effects caused by reproduction "choosing" or "sampling" a finite number of gametes to create the fertilized eggs of the next generation. This does not happen because of external factors. It is just like the fluctuation in the amount of money you have when you play poker or blackjack. Sometimes the cards favor you. Sometimes they favor another player. In the same way, luck sometimes favor one allele over another. This process is called "genetic drift."

Genetic drift is particularly strong when the population size is small. But on average the effects of drift are not biased. It is just as likely that random drift will cause a particular allele to increase OR decrease in frequency. We can't predict what will happen to anyone allele frequency over a long period of time, as a result of drift. But we can predict that, over a long period of time, drift will tend to cause populations that were initially the same in allele frequency to become different. Quantitatively, genetic drift causes a smear of allele frequencies.

The best way to see the effect of drift is to monitor allele frequencies in a large number of populations that are made up of a small number of individual organisms. Because the effects of drift are not directional, we expect the average frequency of a particular allele over all the monitored populations to remain about the same as it was when the experiment started. But the "variance" of the allele frequency, among replicate populations, should increase as evolution proceeds. The exact fashion in which variance is measured will be discussed later, but for now you can think of the variance as a measure of how much allele frequencies vary among populations. High variance among populations means that the allele frequencies of different populations are quite different from each other. Low variance means that the allele frequencies are quite similar.

For example, if we monitor ten populations for the frequency of allele A at a locus, then if there is *low variance* the ten allele frequencies might be:

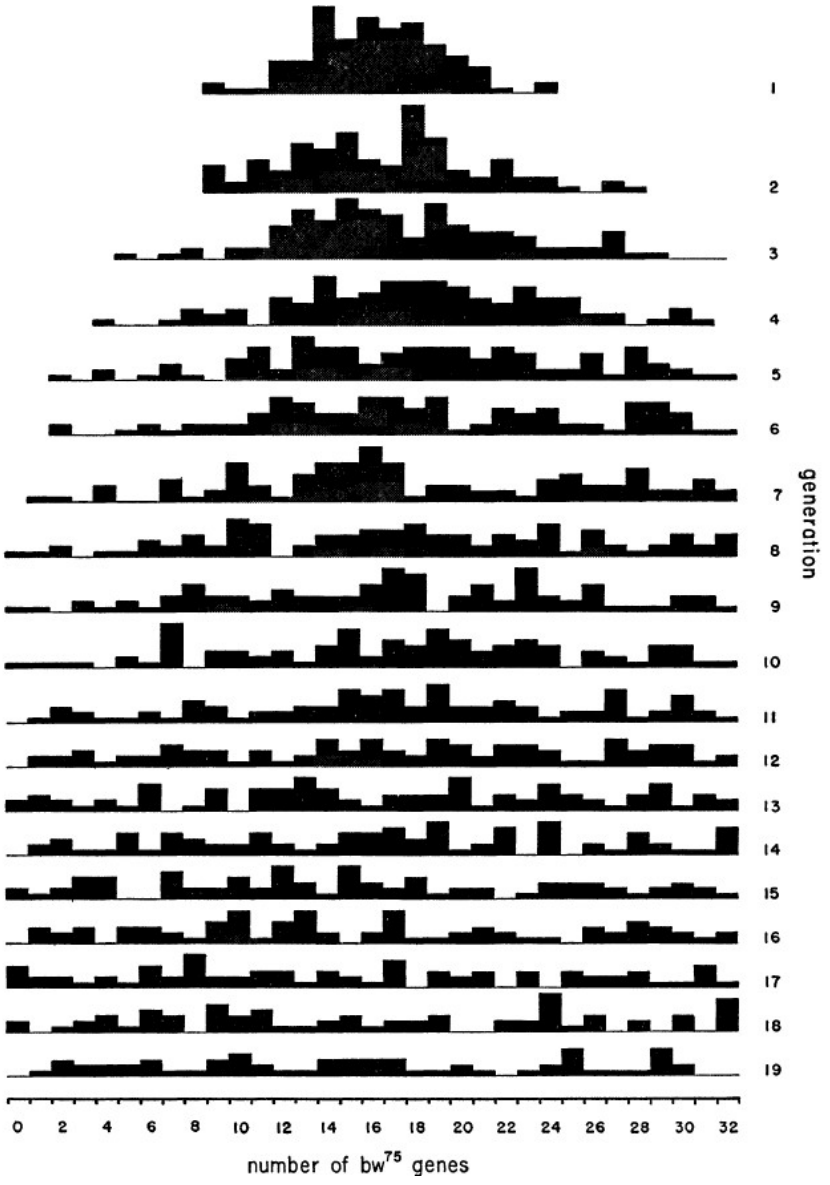
0.12 0.11 0.13 0.11 0.09 0.10 0.08 0.11 0.12 0.11

Alternatively, if there is *high variance,* the ten allele frequencies might be

0.10 0.01 0.34 0.42 0.03 0.22 0.67 0.32 0.12 0.00

Our theoretical prediction is that, if you start with the ten low-variance allele frequencies, genetic drift for many generations might produce the ten high-variance allele frequencies. However, this prediction does not allow us to say exactly which population will evolve to which particular allele frequency. Our ability to predict what will happen with genetic drift is not that strong.

The figure below shows the effect of drift on the variance of allele frequencies over many generations:



The figure above shows the number of *bw*^{75} allele at the autosomal brown locus in 105 populations of *D. melanogaster*. Each population consisted of 8 males and 8 females.

Next we will make these qualitative generalities quantitative and explicit.

Allele and Genotype Frequencies

Regular Diploid Genetic Loci

Suppose that at a single genetic locus there are two alleles, A_1 and A_2. In a diploid population, organisms have two alleles at every locus, so these two alleles will give three genotypes that we will write as A_1A_1, A_1A_2 and A_2A_2.

Suppose we have a sample of N individuals that we can classify as one of these three genotypes. We will write the numbers for each A_1A_1, A_1A_2 and A_2A_2 genotype as N_{11}, N_{12} and N_{22}, in order, respectively. If we call the frequencies of each genotype P_{ij}, their numerical values will be given by,

$$P_{11} = \frac{N_{11}}{N},$$

$$P_{12} = \frac{N_{12}}{N},$$

$$P_{22} = \frac{N_{22}}{N}.$$

If we call the frequency of the A_1 allele p_1 and the frequency of allele A_2 p_2, allele frequencies can be calculated from the genotype frequencies as follows:

$$p_1 = P_{11} + \tfrac{1}{2} P_{12},$$

$$p_2 = P_{22} + \tfrac{1}{2} P_{12}.$$

This makes sense because a heterozygote only contains half as many copies of an allele as a homozygote. Notice also that the sum of the two allele

frequencies is equal to the sum of the genotype frequencies and both of these sums equal one (1).

Loci Located on X Chromosomes

For alleles that come from loci located on the X chromosomes that determine sexual gender, the calculation of allele frequencies is different. Recall that XX gives a human female and XY gives a human male. The same thing is true in fruit flies. From a sample of N females with just two alleles at the genetic locus under study, the three X-chromosome genotypes have counts of N_{11}, N_{12} and N_{22} for the individual A_1A_1, A_1A_2 and A_2A_2 genotypes, respectively. We could use the equations above to estimate the allele frequencies among the females from these observed counts. However, it is possible that the X-chromosome allele frequencies will be different in males, so we need a different set of equations for them.

If we examine M males, let the numbers of A_1 males be M_1 and the number of A_2 males be M_2. Since there is only one X chromosome in males, there is only one copy of an A allele in each male. Then the frequency of the A_1 allele in males is given by $q_1 = M_1/M$ and the frequency of the A_2 allele is $1\text{-}q_1 = q_2 = M_2/M$.

In our experiments, we will use dominant alleles that give the heterozygote and one of the homozygotes in the females the same phenotype. **For this reason, it will be much easier to estimate the allele frequencies of genetic loci located on the X chromosome in the males**.

The Wright-Fisher Model

The major goal of our genetic drift experiment will be to observe the effects of random genetic drift on the mean and the variance of allele frequencies over several generations.

Basic Concept of the Wright-Fisher Model

Sewall Wright, an American biologist, and Ronald A. Fisher, an English statistician, independently developed a very simple model that shows how genetic drift works. We will describe this model in general terms now.

Suppose that we have N diploid individuals in a population. Then there are a total of $2N$ alleles at a diploid genetic locus in this population. If we

assume that these individuals mate by shedding their gametes into a common pool, like some fish do, then the frequency of gametes that bear a particular allele from a particular individual is on average just $1/2N$.

If the frequency of a particular allele (say the allele is called *A*) in the population is given by *p,* then on average there will be *p* times 2*N* gametes of that allele. But sometimes a heterozygous parent will not generate gametes that are exactly fifty:fifty, or even, frequencies of the two alleles that it carries. This is a principle that is used all the time in Mendelian genetics. If a heterozygous (*Aa*) parent generates two gametes, half the time they will be *A* and *a* gametes, one quarter of the time they will be two *A* gametes, and one quarter of the time they will be two *a* gametes.

In the same way, chance is involved in the combination of gametes in a population of size *N*. Each fertilization event involving two gametes can have a variety of outcomes: both gametes can have *A* alleles, both can have *a* alleles, or one *A* and one *a* allele-bearing gamete can combine. In a population with both alleles, it is mathematically possible that all offspring will end up *AA* homozygotes. Or they could all be *aa* homozygotes. That is, genetic drift can accidentally "fix" one allele or the other, even if the previous parental generation is genetically polymorphic.

In addition to this extreme possibility, the same sort of sampling effect can cause the allele frequency of a population to rise or fall, even though there is no "directional" evolutionary mechanism, like selection, acting on the population.

The mathematics that underlies the Wright-Fisher model is that of *combinatorics.* But don't let this term impress you. You use combinatorics every time you play a game of chance. Combinatorics tells us that getting dealt a bridge hand of thirteen cards all of the same "suit" (Hearts, Spades, Diamonds, or Clubs) is very rare compared to getting a mixture of two or more suits. Similarly, the changes of being dealt all four Aces and a King in five-card stud poker is very rare. In the same way, the accidental fixation of the *A* allele in one generation is an improbable (but not impossible) event in a population of ten individuals if there are only ten (out of a maximum number of 20) copies of the *A* allele in the parents of the preceding generation.

Quantitative Predictions of the Wright-Fisher Model

There are several important theoretical results that have been mathematically derived from the Wright-Fisher model of genetic drift that

will be illustrated by our genetic drift experiment. Some of these ideas are reviewed in chapter 7 of Hartl and Clark, among other textbooks in population genetics. Here we will give the major theoretical results for the Wright-Fisher model, results that supply us with predictions for our experiment.

Suppose that, in a small population with effective population size N, the initial frequency of an allele is p_0. At some time in the future, t, the frequency is p_t. The Wright-Fisher model predicts the following:

Let E represent the "expected" or mean value for a variable. Then the symbol $E(p_t)$ stands for the expectation of the random variable p_t. This is similar to the mean of the random variable p_t.

$$E(p_{t+1}) = E(p_t), \tag{1a}$$

In words, this means that genetic drift does not, **on average**, change the frequency of an allele.

The variance ("Var") is a measure of "dispersion" about the mean value. The greater the variance, the more individual values deviated from the mean or expected value for a variable. There are two variances that we can predict when genetic drift occurs.

The first of these variances is the variance in allele frequencies that arises from a single generation of random sampling of gametes in the creation of the next generation. Let $\mathrm{Var}(p_{t+1} \mid p_t)$ represent the variance in allele frequencies at time t+1 given that the allele frequency was p_t at time t. This is the variance of allele frequencies due to just a single generation of drift, which is given by the following equation.

$$Var(p_{t+1} \mid p_t) = \frac{p_t(1-p_t)}{2N}, \tag{1b}$$

Notice from this equation that, if N is absolutely huge, there will be virtually no genetic drift in a single generation, because N appears in the denominator of the right-hand side of the equation.

The second variance that we are interested in is the accumulated variance over the entire sequence of generations in which genetic drift occurred. Let $\mathrm{Var}(p_t)$ represent the variance of the allele frequency at time t. It is given quantitatively by the following equation.

$$Var(p_t) = p_0 q_0 \left[1 - (1 - \tfrac{1}{2N})^t\right]. \quad (1c)$$

Again, notice that if N is very large, the right-hand side of this equation will be very close to zero, because $1/2N$ will be close to zero and the numeral 1 (one) to a very high power is still 1 (one).

Bear in mind that equations 1(a-c) are theoretical predictions from the Wright-Fisher model. It is possible that the actual mean and variance in this experiment may be different from these predictions. One goal of this experiment will be to compare the observed mean and variance in allele frequencies from our experiment with the expected values from the Wright-Fisher model.

We will discuss methods of estimating the variance of allele frequencies from actual population samples next.

Methods

Scoring Genotypes

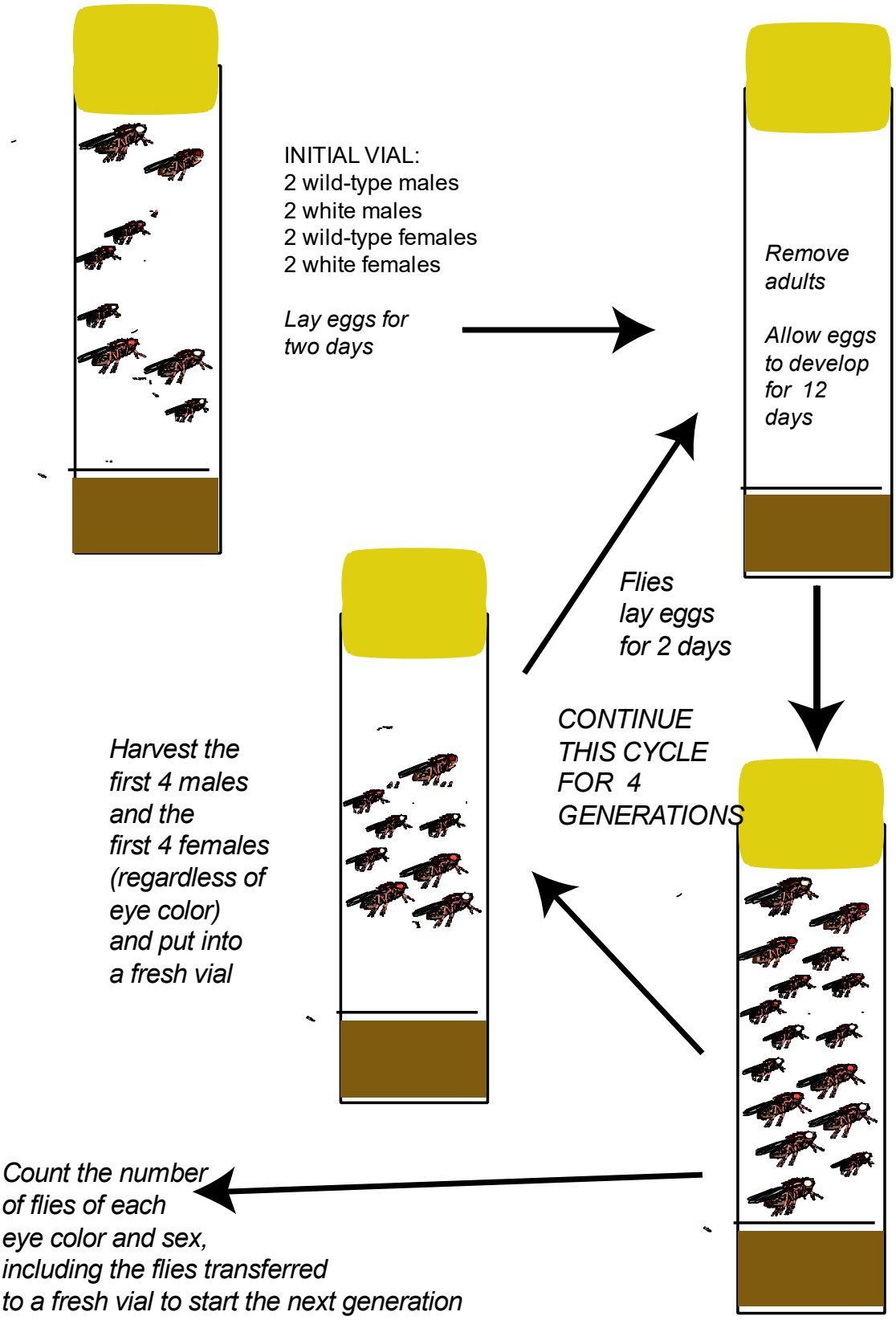
This experiment uses two alleles at the *white* (or *w*) locus, a gene that affects eye color. The gene *white* is located on the *X* chromosome of *Drosophila melanogaster*. One allele is referred to as *w*. Females homozygous for *w* have white eyes, males with one copy of *w* (a condition that is called "hemizygous") also have white eyes. The second allele, +, is wild type and is dominant to *w:* females heterozygous (*w*/+) or homozygous (+/+) for the wild allele have red eyes. Males hemizygous for the + allele also have red eyes. **The frequency of the *w* allele is most easily estimated by counting the total number of white males divided by the total number of males.**

Population Maintenance

Each group of two students will be responsible for ten populations. Each population will initially consist of four males (two wild type and two white-eyed) and four females (two wild type homozygotes and two white-eyed homozygotes). Thus, the initial frequency of the *w*-allele is 0.5. These 8 adults will be put into a single 8-dram vial and allowed to lay eggs for 2 days in a 25° C incubator, as shown in the figure. You will have a total of ten vials, one for each population.

DRIFT EXPERIMENT:

HANDLING OF EACH VIAL IN EACH GENERATION



After these two days of egg-laying in a 25° C incubator, the adults should be removed and discarded. The vials for each of your populations must be labeled with the group's name and the population number, e.g. 1, 2, …, 10. You need to do this because you will be determining the generation-by-generation trajectory for each population. Return the vials to the 25° C incubator.

Two weeks later, you will census your 10 populations. One efficient way to do this is as follows. Take a single vial and knock out all the adults using CO_2 and place them on a CO_2 platform. Take a paint brush and make a long thin line of all the adults. Take the first four males and the first four females in the line, record their phenotypes and place them into a fresh vial, as shown in the figure.

Type and record the phenotypes of all remaining adults. The remaining adults may then be discarded. However, before doing this make sure your initial sample of eight has recovered from anesthetization. If there are some flies in the group of eight (that were put in the fresh vial) which do not recover, replace them with flies of the same type from the rest of the flies that emerged in that vial.

Compute allele frequencies from all the male data. Recall that you can estimate the allele frequency from the frequency of the corresponding hemizygous male genotype. For example, if you have 10 *w* males and 30 + males, the frequency of the *w* allele is 0.25 and the frequency of the + allele is 0.75. The frequency of the male phenotypes gives the allele frequency in the current generation for each vial's population. Record your data in tabular form, as indicated in the table below:

Drift Experimental Data

Population	Genotype/Sex	Generation 1	Generation 2	Generation 3
1	w-male			
	r-male			
	w-female			
	r-female			
2	w-male			
	r-male			
	w-female			
	r-female			
3	w-male			
	r-male			
	w-female			

	r-female			
4	w-male			
	r-male			
	w-female			
	r-female			
5	w-male			
	r-male			
	w-female			
	r-female			
6	w-male			
	r-male			
	w-female			
	r-female			
7	w-male			
	r-male			
	w-female			
	r-female			
8	w-male			
	r-male			
	w-female			
	r-female			
9	w-male			
	r-male			
	w-female			
	r-female			
10	w-male			
	r-male			
	w-female			
	r-female			

This procedure will be continued for four generations, for a total of 6 weeks. At the end of the 6 weeks, the data from all groups will be shared. Each group will then conduct their own analysis of the data obtained by the entire class.

Sample Variance

Equation 1c gave the expected variance due to drift. This equation shows how the population size and the allele frequencies affect the variance that is generated by drift. At the end of this experiment, you will have data from

ten populations in the form of allele frequencies for each population at each generation. Estimating the variance in this sample is different than computing the theoretical variance expected from drift alone. This is because actual experimental data almost always differs at least slightly from the predicted results.

Suppose that in one generation the allele frequencies that you observe in the ten populations are represented by, ${}_1p$, ${}_2p$,.., ${}_{10}p$. Then the sample variance, s^2 is computed as,

$$s^2 = \frac{1}{(10-1)} \sum_{i=1}^{i=10} \left({}_i p - \overline{p} \right)^2 ,$$

where,

$$\overline{p} = \frac{1}{10} \sum_{i=1}^{i=10} {}_i p \, .$$

If you have 100 populations then you would replace the 10 in the equation above with 100.

Questions for Your Lab Report

After the experiment is over you should have the results from the other groups as well as your own. YOUR PRIMARY ANALYSIS SHOULD BE THE POOLED DATA SET FROM ALL GROUPS USING THE MALE DATA TO ESTIMATE ALLELE FREQUENCIES. In answering the questions below remember the following two point. (i) The theoretical mean and variances from the Wright Fisher model can be treated as statistical constants when comparing them to the observed means and variances. (ii) The observed means and variances are estimates with uncertainty which can be summarized with confidence intervals.

1. Explain any obvious major differences between your results and the rest of the class, if any.

2. Since everyone started their populations at the same initial allele frequency of 0.5, examine the variance in allele frequencies due to a single generation of drift using this first generation data. Is it what we would expect from equation 1b? In this experiment, notice that N is equal to 8. Since the initial allele frequency of both w and + alleles is 0.5, you have all the information needed to calculate the single generation

drift variance. Are your actual data close the prediction of equation 1b, 1/64? If not, can you think of reasons why there might be a difference between the observed variance and the variance predicted by equation 1b? [Hint: is it possible the theoretical prediction is not correct for this experiment?]

3. What happened to the observed variance in allele frequencies among over time? Compare the observed variance to the theoretical predictions of equation 1c, generation-by-generation.

4. What happened to the mean allele frequency over time? Compare to the theoretical expectation. Were there statistically significant differences between the observed mean and the expected mean allele frequencies?

References

Hartl, D. L. and A. G. Clark. 1997. *Principles of Population Genetics*. Chapter 7.

Christiansen, F. B. and M. W. Feldman. 1986. *Population Genetics*. pgs. 68-73.

Natural Selection

Introduction

The term "natural selection" refers to the differential net reproduction of genotypes arising from fitness differences among those genotypes. Those fitness differences may be expressed at different times in the life cycle. In this experiment we will examine just one component of fitness, viability. Viability is the probability of a genotype surviving from egg to adult.

To understand how natural selection acts, consider a generic life-cycle, as shown in the following figure.

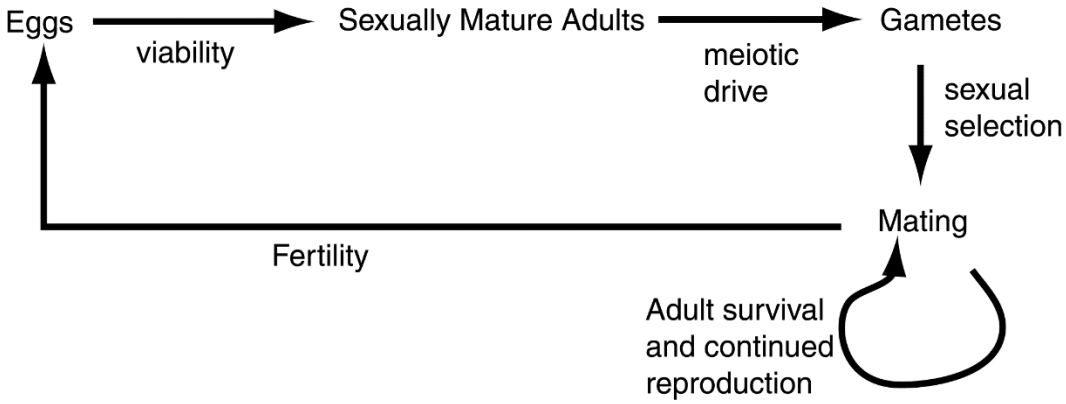


Figure. A typical life cycle that reveals the various ways selection can affect the relative numbers of genotypes that make it to the next generation.

Natural selection is not some cosmic force, orchestrating the evolution of dinosaurs, flowering plants, or modern mammals, contrary to the writings of journalists, "popular science writers," and some wayward paleontologists. Natural selection is instead about differences among genotypes in their life-history characteristics. In some environments, at some times, some genotypes have higher viability, mating success, or fecundity. What this does is increase the transmission to the next generation of the alleles that those genotypes possess. That's all. There is nothing cosmic or progressive about this differential transmission. It is "one foot in front of another." The process of natural selection has no sense of history, inertia, or goal. It is a blind mechanism.

By far the best way to understand how natural selection works, and works mechanically, is to look at the consequences of simple genetic differences in

life-history for the transmission of these genes. We will do this in the present laboratory experiment.

Though it is mechanical and immediate, the action of natural selection can be somewhat complicated. Natural selection can act at several points in this life cycle. If genotypes vary in their chance of surviving from egg-to-adult then *viability selection* can act. This is the type of natural selection that is most often studied by evolutionary geneticists in the laboratory.

But there are other points at which natural selection can arise. From Mendel's laws, we expect that heterozygous adults will produce equal numbers of gamete carrying each of the two alternative alleles. However, some alleles may systematically bias the production of gamete types such that the relative numbers are not 1:1. This process is called *meiotic drive,* and it may increase the relative frequency of one allele. This is *not* the same thing as accidental departures from 1:1 ratios arising from genetic drift. With meiotic drive there is a persistent bias in favor of one allele over other alleles. However, meiotic drive is a rare form of natural selection. In particular, it is not appropriate to use it as an explanation for every deviation from 1:1 genetic segregation. The Wright-Fisher model of accidental genetic sampling effects is more appropriate, almost all the time, as an explanation of such deviations.

Once the offspring have grown up, they are adults who must then mate to produce fertile eggs. Males often compete with each other for the opportunity to mate with females. [More on this in the next section.] Sometimes females compete for mates, too. In some cases, genetic differences confer a mating advantage to their carriers and the alleles that produce these differences may increase due to *sexual selection*. Females in turn may differ in the number of offspring they produce due to their genotype, and this will create the opportunity for *fertility selection*. If the adults are capable of reproducing more than once, then *age-specific selection* may act if there are differences in survival or fertility at later ages. This is an important topic in this laboratory course.

Methods

Our natural selection experiment will use genotypes at the *white* locus. By doing specific crosses, we can use our knowledge of Mendelian genetics to determine the expected frequencies of genotypes among the zygotes. If the frequencies of adults systematically and consistently depart from these expected zygote frequencies, we can infer that this is due to relative

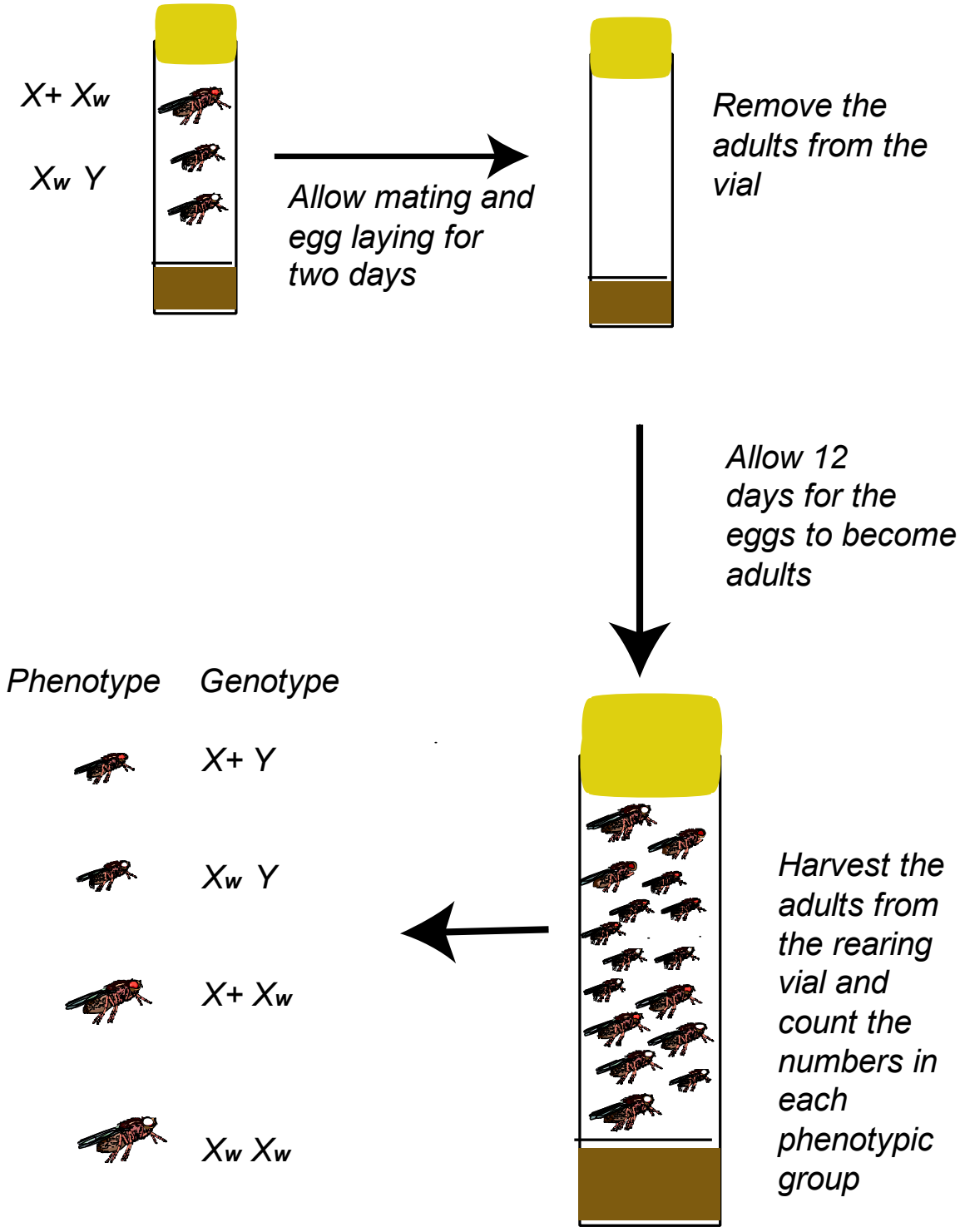
viability differences at loci for which meiotic drive does not arise. [This is the case for the *white* locus.]

You will be provided with virgin females that are heterozygous at the *w* locus and *w* males. In each vial place one female and two males. [We use an extra male to forestall the problem of males who are either unable to perform sexually or are not found to be appropriate mates by females. Much of the time, however, either male with suffice. It is even possible that the female will mate with both males.] Let this threesome mate and lay eggs for two days and then remove the adult flies. The female will almost always produce fertilized eggs that will develop into larvae.

Two weeks later, determine the phenotypes of all male and female progeny in the vials in which eggs were laid.

The initial cross is shown below, with the expected X-chromosome genotypes of the male and female offspring.

NATURAL SELECTION EXPERIMENT:



Female parent			Male parent	
w/+		×	*w*	
		Progeny		
Female offspring			Male offspring	
w/*w*	*w*/+		*w*	+
0.5	0.5	relative frequencies	0.5	0.5
White-eye	wild	phenotypes	white-eye	wild

Relative fitness estimates

In any single vial, the raw data from these experiments will be the numbers of white-eyed males and white-eyed females and the number of wild-type (red-eyed phenotype) males and females. Let these numbers be given by N_{wm}, N_{wf}, N_{+m}, and N_{+f} , respectively. Let the relative fitness of the *white* males and females be equal to 1. Then, the *relative* fitness of the wild-type males ($W_{+,m}$) and the heterozygous females ($W_{+/w,f}$) is given by,

$$W_{+,m} = \frac{N_{+m}}{N_{wm}},$$

$$W_{+/w,f} = \frac{N_{+f}}{N_{wf}},$$

From this experiment we have not been able to estimate the fitness of females homozygous for the wild type allele, $W_{+/+,f}$. This is a relative fitness, because we have taken the viability of the flies with white eyes as the point of reference, and we are ignoring possible differences in the other life-history characters.

If there are many vials where N_{wm} or N_{wf} are zero you can assume the wild type flies have a relative fitness of 1 and then estimate the relative fitness of the white males and heterozygous females using the reciprocals of the equations above.

The eye color of the red-eyed heterozygotes is essentially the same as that of the homozygotes bearing two copies of the wild-type (red eye) genotype, which implies full dominance of the red-eye allele. For convenience, in

your lab write-up you can assume that the effects of the wild-type on viability are also fully dominant. In particular, you should address the questions asked below making the assumption that $W_{+/+,f} = W_{+/w,f}$. We don't expect this type of perfect dominance for all alleles affecting viability, but it is a useful point to start with.

Natural Selection Experimental Results

Replicate	Progeny w-male	r-male	w-female	r-female
1				
Relative fitness	male		female	
2				
Relative fitness	male		female	
3				
Relative fitness	male		female	
4				
Relative fitness	male		female	
5				
Relative fitness	male		female	
6				
Relative fitness	male		female	
7				
Relative fitness	male		female	
8				
Relative fitness	male		female	
9				
Relative fitness	male		female	
10				
Relative fitness	male		female	

Questions for the Lab Report

1. Compute the relative viabilities of all the male and female genotypes in this experiment. Treat each vial as an independent sample, thus there will be 10 fitness estimates for each genotype. Put confidence intervals on the mean estimates using R and discuss the general implications of these numbers. Is there evidence for natural selection acting on relative viability in the experimental vials? Which allele appears to be favored by natural selection in this experiment?

2. Assuming that females who are homozygous for the wild-type allele have the same viability and fecundity as the heterozygotes at this locus, what would be the *ultimate* outcome of natural selection acting on the *white* locus polymorphism? [Hint: use the equations in the appendix and iterate the equations for many generations, following the change in *white* allele frequency. You can do this with computer software like Excel or R.]
3. Is it possible for both alleles to be stably maintained without heterozygote advantage in the females? [*Hint:* Use the equations in the appendix to help answer this question. You may calculate numerical examples using the techniques developed to answer the second question. If you are really ambitious, these equations can be used to derive analytical results concerning the long-term evolution of populations with the fitnesses you have estimated.]

References

Hartl and Clark. 1997. Chapter 6, pages 211-236.
Lewontin, R.C. *The Genetic Basis of Evolutionary Change*. Chapter 3.

Sexual Selection

Introduction

In animals that have two sexes, there is often a difference in the time and energy each will devote to reproduction. Usually, the female will make the larger investment of time and energy since she will produce the more energy-laden eggs, and for some animals she will be involved with some care of the offspring or developing eggs. This is obviously true of mammalian females, whose bodies have a number of features that allow mothers to nourish fetuses and newborns, such as placentas and mammaries. In addition, mammalian females often supply their offspring with solid food to supplement their milk, particularly among the carnivores, in which the young are not usually effective hunters.

But in many cases among the fish species, the most abundant vertebrates, the male invests most in the care and feeding offspring. This reaches the point of full reversal of gender roles in seahorses, in which males get "pregnant" and incubate their offspring until they "give birth," their offspring emerging from their large brood pouch. In either case, it is often found that the parent that is investing more energy in reproduction controls when and with whom mating will take place.

As a result of this asymmetry in the decision making process, the sex which invests less energy in caring for offspring, usually males among insects, will compete among themselves to be chosen as mates by females. This type of competition is called *sexual selection.* If there are characteristics that are inherited by males that give them some advantage in this competition for mates, such as structures or behaviors that females find attractive in a prospective mate, then we can expect sexual selection to favor these characteristics. In some species, like many birds-of-paradise, males have elaborate coloration, long tails or build complex structures to attract females. All of these characters are thought to result from evolution driven by sexual selection. In some cases, sexual selection may have produced cumbersome male morphology or dangerous competitive male behavior that actually reduce male fitness, compared to the fitness that males might have achieved if females did NOT discriminate among them.

There is thus the potential for antagonism between the effects of sexual selection and viability selection, the focus of the previous experiment. Viability selection may successfully oppose sexual selection, preventing the evolution of extreme morphology or behavior. For instance, brightly

colored males may be more attractive to females but may also be more conspicuous to predators. It is interesting to note that birds are generally more colorful than mammals of similar size, suggesting that the greater ability of birds to flee from predators may have tilted the balance toward less camouflaged sexual plumage. Thus, the evolutionary dynamics of sexually selected traits may be quite complicated.

Laboratory sexual selection will be studied by measuring a male-limited component of fitness called *virility*. In biology, the term virility refers to the relative success of males in being chosen as mates by females, when multiple males are striving to mate with the same female(s).

Methods

Each laboratory group will have a population of experimental males that are wild type. The virility of these males will be tested against males that carry the white (*w*) allele on the *X*-chromosome. This experiment allows females to choose between males with red eyes and males with white eyes. We do not have to assume that females will discriminate between males on the basis of their eye color. Whether they do so or not will be determined by the experiment itself. Indeed, this is one of the most important questions that you should answer in your write-up of your laboratory report.

1. Place one wild type male and one white-eye male in each of ten vials. Make sure that you do not expose males of one eye color to more CO_2 than males of the other eye color.
2. After both males have recovered from CO_2, about 10-15 minutes, take 10 wild type virgin females and place each one in each vial. By letting both males recover fully you ensure that neither has an advantage over the other.
3. Carefully watch the females. When a female has mated for more than 30 seconds record the type of male she mated with. It takes male fruit flies more than a few minutes to transfer much sperm, unlike most mammals, so a mounting that only last 10-30 seconds is more likely to be an unsuccessful mating attempt, rather than a successful fertilization.

In order to make the duration of this laboratory reasonable, we will place an arbitrary limit on the period during which you will watch the females choose between males. This will be announced at the start of the class.

The number of matings by wild-type males can be thought of as a binomial random variable. The probability of the wild type male successfully mating is given by,

$$V_{+} = \frac{[\text{number of wild males mating}]}{[\text{total number of matings}]}.$$

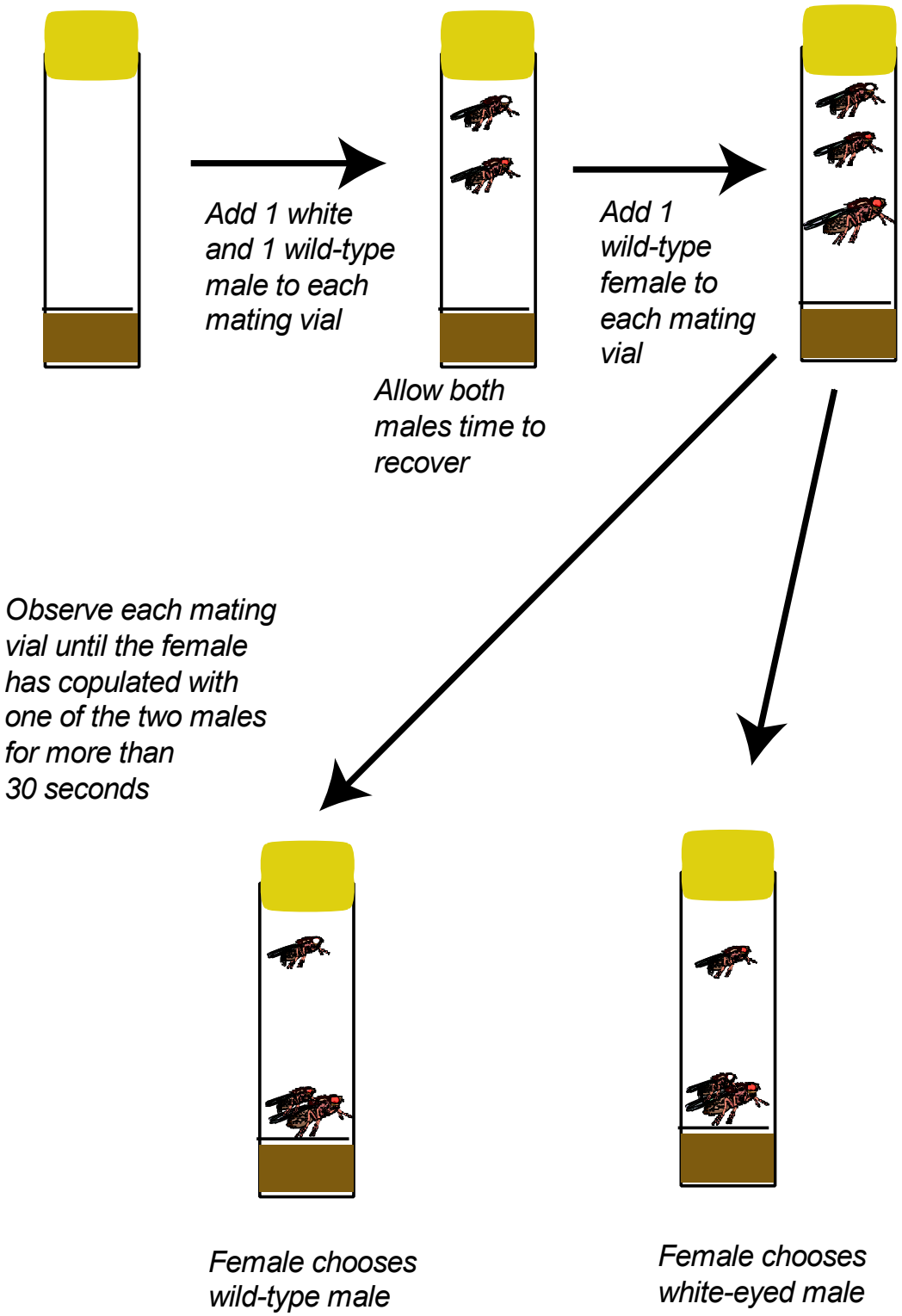
With this definition, the mating success of white-eye males is simply 1 - V_{+}.

Sexual Selection Experimental Results

Male	Number of matings Your experiment	 whole class
Red-eye		
White-eye		
Total		

1

SEXUAL SELECTION EXPERIMENT:



Questions for the Lab Report

1. In your experiment do the females show a preference for *white* or wild type males?

2. Summarize the results of the entire class. Put confidence intervals on the relative mating success of white and wild type males, using R. [Hint: use binom.test]

3. Suppose someone suggested using rates of population growth as a measure of fitness (e.g. taking a population of flies homozygous for the white allele and comparing that to a population homozygous for the wild type allele by measuring their population growth rate). How would virility affect this measure of fitness, assuming that all females eventually mate?

References

Freeman, S. and J.C. Herron. 1998. *Evolutionary Analysis*. Chapter 15.

Appendix

In this appendix we will develop allele frequency recursions for the sex-linked viability model. We must keep track of allele frequencies in the males and females separately. Assume there are two alleles, A_1 and A_2. Let the frequency of the A_1 allele in the females be p_f and in the males p_m. To follow the results refer to the table below.

	Females			Males	
Genotypes	$A_1\,A_1$	$A_1\,A_2$	$A_2\,A_2$	A_1	A_2
Fitness	F_{11}	F_{12}	F_{22}	M_1	M_2
Zygote frequencies	$p_f\,p_m$	$\left(1-p_f\right)p_m + p_f\left(1-p_m\right)$	$\left(1-p_f\right)\left(1-p_m\right)$	p_f	1-p_f
Relative frequency after selection	$p_m p_f F_{11}$	$\left[\left(1-p_f\right)p_m + p_f\left(1-p_m\right)\right]F_{12}$	$\left(1-p_f\right)\left(1-p_m\right)F_{22}$	$p_f M_1$	$\left(1-p_f\right)M_2$
Absolute frequency after selection	$p_m p_f F_{11}/\overline{W}_f$	$\left[\left(1-p_f\right)p_m + p_f\left(1-p_m\right)\right]F_{12}/\overline{W}_f$	$\left(1-p_f\right)\left(1-p_m\right)F_{22}/\overline{W}_f$	$p_f M_1/\overline{W}_m$	$\left(1-p_f\right)M_2/\overline{W}_m$

We assume that the genotypes of zygotes are initially in Hardy-Weinberg proportions, with the frequency of each genotype the product of the frequencies of each allele, keeping track of order. [This means that, if we don't keep track of order, the frequency of the heterozygotes is *twice* the product of the individual allele frequencies.] Since the male and female allele frequencies are possibly different, these Hardy-Weinberg frequencies look different than the usual ones. To convince yourself they are the Hardy-Weinberg proportions, let $p_m = p_f$ and simplify the expressions to the one-locus case.

The next step in the calculations reflects the change in the relative frequencies of the zygotes due to differential survival, as may be seen in the line labeled "relative frequencies after selection". Since these relative genotype frequencies no longer sum to 1, we divide them by the mean fitnesses for each sex to calculate the absolute frequencies after selection. [Now these frequencies do sum to 1.] Since fitnesses are different in males and females, we have different average fitnesses in the males ($\overline{W}_m$) and females ($\overline{W}_f$). These mean fitnesses are given by:

$$\overline{W}_f = \left[p_m p_f F_{11} + \left(\left(1 - p_f\right) p_m + p_f \left(1 - p_m\right) \right) F_{12} + \left(1 - p_f\right)\left(1 - p_m\right) F_{22} \right],$$

and

$$\overline{W}_m = p_f M_1 + \left(1 - p_f\right) M_2$$

The frequencies of the A_1 alleles in the next generation are then,

$$p'_f = \left[p_m p_f F_{11} + \tfrac{1}{2}\left(\left(1 - p_f\right) p_m + p_f \left(1 - p_m\right) \right) F_{12} \right] / \overline{W}_f$$

$$p'_m = p_f M_1 / \overline{W}_m$$

The prime (') symbol indicates the value of the allele frequency in the next generation. The female frequency (p'_f) is simply the frequency of A_1A_1 homozygotes plus half the heterozygote frequency while the male frequency (p'_m) is just the frequency of A_1 males.

Age-Specific Selection

Introduction

The basic theories of natural selection often assume that the life cycle of the organism is discrete. This is the assumption underlying the diagram at the start of the "Natural Selection" experiment's Introduction. In such life cycles, eggs become immature larvae. After a period of preadult growth, development, and maturation, the sexually mature adults mate and produce offspring in a short interval of time. With discrete generations, all the adults either die or completely stop reproducing, leaving the offspring to establish the next generation. Annual plants and "univoltine" insects are just some of the examples of species that have such a well-synchronized life cycle, in which each generation only mates with itself, in one well-defined bout of reproduction.

Many other species, however, have more complex life cycles. You should understand this already, because humans have multiple bouts of reproduction and mating is not confined to individuals of the same age. In fact *Drosophila* too have an extended period of adult life during which reproduction can take place multiple times, with repeated bouts of mating and egg-laying spread out over a number of weeks. In this respect, as in some others, fruit flies are useful analogs of humans. Such populations are said to have *overlapping generations* and *age structure.*

This type of life history greatly complicates the description of natural selection. In computing fitness, we have to consider the survival of adults and the amount of reproduction that takes place at each adult age. In addition, we still have to take viability and mating into account. As we shall see shortly, the fitness of a genotype will not only depend on how long the genotype lives and how many offspring it produces, but also on the timing of production of these offspring. In general, fitness will increase most by having progeny early in life rather than later. This general rule has portentous implications for evolution, particularly with respect to the evolution of aging. Natural selection in this relatively complex context is sometimes called *age-specific selection. A*ge-specific selection requires the most complete accounting of all life-history stages in the determination of fitness. Understanding selection in an organism as simple as a fruit fly or as complex as a human being requires quantitative measurement of the potential for age-specific selection.

Theory of Age-Specific Selection

The description of age-structured populations first requires that the life cycle be divided into a number of equally spaced time intervals. These time intervals are the basis for dividing up the members of the population into defined *age-classes*. Generally the length of the time interval that defines the age-classes is somewhat arbitrary. The choice of time interval usually depends on the type of population and amount of information available for estimating key components of the age-structured life-history. In humans, age-classes are usually defined by time intervals of one or five years. In fruit fly experiments, age-classes are usually defined by days or weeks. In developing the theory of age-specific selection here, we will ignore these particulars.

Suppose the first age class consists of newborns and the older age-classes are formed from individuals which have survived from earlier age-classes. We will represent the number of individuals in each age-class at time t as $n_1(t)$, $n_2(t)$, ..., $n_d(t)$, where d is the last age-class. We will also represent the quantitative probability of surviving from one age-class to the next as P_x, x= 0, 1, 2,...,d-1. P_0 is the probability that newly produced eggs or offspring survive to the first age class. P_d is not defined because by definition no one lives beyond the *dth* age class.

For each age-class after the first, the number in each age-class is given by,

$$n_{x+1}(t+1) = P_x n_x(t), \text{ for } x = 1, \ldots, d\text{-}1. \qquad (1)$$

To calculate the number of individuals in the first age class formulaically, we must first describe the birth process. Let the number of offspring born to a female aged x be m_x. Then the number of these offspring that survive to the first age class, per-female is given by,

$$f_x = P_0 m_x .$$

If we assume, for the sake of simplicity, that there are equal numbers of males and females in each age-class, then the number of progeny produced per-adult (male or female) in age-class x will be $f_x/2$. With all this notation, the total number of individuals in the first age class is given by,

$$n_1(t+1) = \sum_{x=1}^{x=d} \tfrac{1}{2} f_x n_x(t) . \qquad (2)$$

This equation simply adds up all the progeny produced by each age-class. If the young age-classes are sexually immature, f_x may be 0 for all values of x less than the age at which reproductive maturation first occurs. To characterize the fitness of alternative genotypes will requires the collection of information on the age-specific probabilities of survival, P_x, and the age-specific fecundities, f_x. Note that in this discussion we have ignored male fertility, although it could be taken into account by a further elaboration of this model. Usually male fertility does not limit female fertility, particularly in non-monogamous populations. [Almost no species are strictly monogamous, and certainly humans and fruit flies aren't.]

Fitness in Age-Structured Populations

Populations that grow according to eqs. (1-2) have age-structure. An important feature of age-structured populations is that they grow in a consistently exponential pattern after the proportions of the age-classes stop fluctuating. When there is no more fluctuation in age-class composition, we say that a population has reached a stable age distribution. The following figure shows the process by which a hypothetical population living in a constant environment achieves a stable age-distribution. From this figure we see that after some time has passed, all age-groups, as well as the total population, begin to grow at the same exponential rate.

You might think that it shouldn't matter just how a population is distributed into age-classes, but it does. Consider a scenario in which all the students enrolled in the class are dumped on a desert island for the rest of our lives. We will further assume that, while you have enough food and there are no contagious diseases, you will still age, with falling probabilities of survival and falling fecundity as you get older. Unfortunately, you won't have birth control, television, or video games, so your limited recreational opportunities will lead to a high birth rate. Since most of the students in the class will be in their early twenties, your total fertility will be quite high at first. Soon there will be a number of babies. The babies won't be reproductive for a while, but the original group will still be able to reproduce while your children grow up. Once the first group of children grow up, they won't have TV either, so they will be making your grandchildren. The original group will start to get old, and the females among you won't have much fertility after the age of 45. The population will come to consist of a mixture of juveniles, potentially reproductive adults, and post-reproductive adults. With this pattern, we expect a high population growth rate initially, then a fluctuating population growth rate

for a time until the composition of the population settles down to a stable mix.

In the theory of age-structured populations, it is the exponential rate of growth of a hypothetical population composed of just one geonotype that is used as a summary of the fitness of that genotype. Generally, the rate of exponential increase is most sensitive to changes in survival and fecundity early in life. Thus, doubling the fecundity of age-class 2 would have a much more dramatic impact in increasing the rate of growth than would doubling the fecundity in the last age-class. Similar results apply to changes in age-specific survival. This makes intuitive sense. We can expect such a reduction in the importance of each age-specific life-history character as its age of occurrence increases if only because earlier deaths lead to less common expression of later life-history attributes.

The rate of exponential increase exhibited by a genotype can be found by solving a particular polynomial equation. For the example in figure 1 this equation looks like,

$$\lambda^4 - f_1\lambda^3 - P_1f_2\lambda^2 - P_1P_2f_3\lambda - P_1P_2P_3f_4 = 0,$$

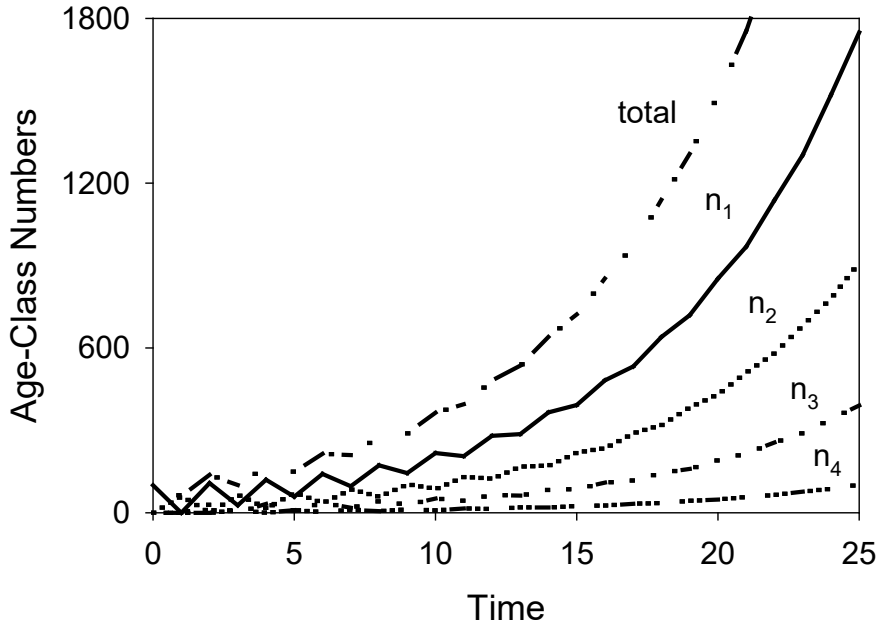


Figure 1 The change in the total population size and four different age-classes. The survival probabilities for P_1, P_2 and P_3 were 0.6, 0.5 and 0.3 respectively. The age-specific fecundities were, 0, 0.9, 1.8 and 0.4. The initial population consisted of 100 1-year olds.

where the largest positive real root of this equation, λ_m, is the rate of exponential increase of the population. In the case of the example in figure 1, $\lambda_m = 1.0441$. This means that the population is increasing by a factor of about 4.4% each time interval.

In this experiment age-specific mortality rates and fertilities will be estimated for a population of wild-type *Drosophila melanogaster*. From this information, you will use the theory developed in the previous sections to estimate a fitness value from these observations.

Methods

Each laboratory group will get their own population of flies. At the start of the experiment, your population will consist of 50 males and 50 females that are less than one week old as adults. These flies will be wild type. .

1. Count out 20 males and 20 females at random and place them in 20 charcoal vials, one male-female pair per vial.

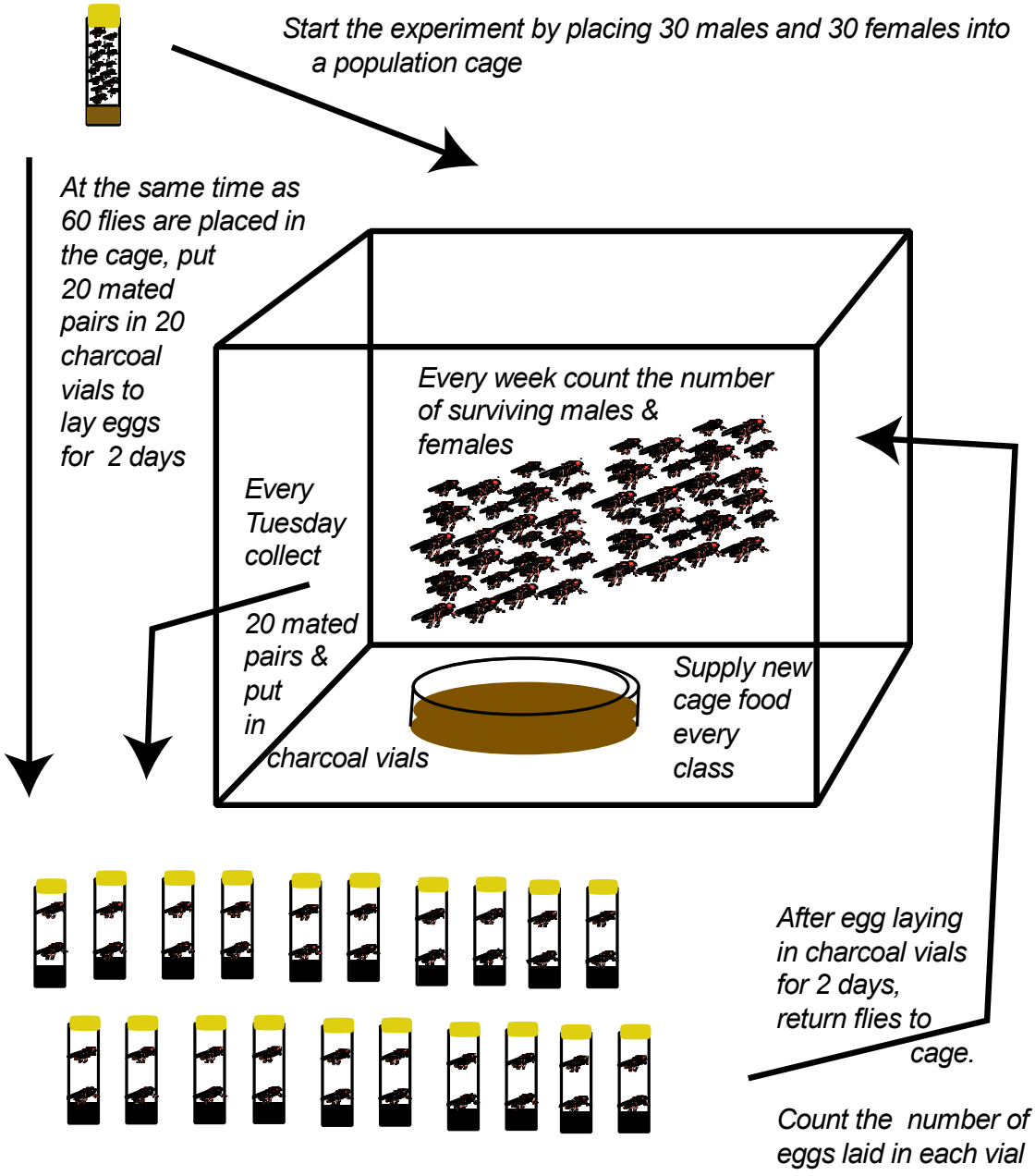
2. Count the remaining flies (initially 30 males and 30 females) and place them in the population cage. After the flies wake up in the cage place a petri dish of ***plain*** food in the cage.

3. After 48 hours remove the females from the charcoal vials and return them to your cages. Replace the petri dish in the cage with a fresh petri dish of food ***plus a large dab of yeast paste***. Count the eggs laid in the charcoal vials over the previous two days.

4. The process described above will be repeated four more times, at weekly intervals. At each weekly interval the total number of surviving males and females should be counted in your population cage and recorded. Food in these cages should be replaced every Tuesday and Thursday, as described above.

AGE-SPECIFIC SELECTION EXPERIMENT:

Once the experiment has started, you will be maintaing a same-aged cohort in a population cage. Every week, you will census the population, set up egg -laying vials, and counting the eggs laid in each laying vial after two days. After the egg-laying, RETURN the flies from the egg-laying vials to the population cage.



Analysis of Growth Rates

For the data collected in this experiment, we will make several assumptions. We will assume that the number of eggs laid by females is equivalent to the number entering the first age-class, that is $P_0 = 1$. The pre-adult age-classes will all be lumped into age-class 1 even though these take more than a week to complete. The survival from the preadult to adult age classes (P_1) will be set to one, because we don't have any good information about survival during this life-cycle, while the fecundity of the pre-adult age-classes is by definition zero. In this experiment, there will be data on five adult age-classes, thus the equation on page 35 will be a sixth-order polynomial rather than a fourth order polynomial.

The largest root of these polynomial equations can be solved with the R function "polyroot." This function takes as its arguments the coefficients of the polynomial starting with the constant term. For instance, using the parameter values in figure 1, the polynomial we must solve looks like, $0 = -0.036 - 0.54\lambda - 0.54\lambda^2 + \lambda^4$. The R commands and results produced for this example are shown below.

```
> polyroot(c(-0.036,-0.54,-0.54,0,1))
[1] -0.07176821-1.349959e-21i -0.48617915+4.940097e-01i
[3] -0.48617915-4.940097e-01i  1.04412651+3.885781e-16i
```

Since the polynomial is fourth order there are four roots. Each root is written as a complex number with the general format, $a+bi$, where $i = \sqrt{-1}$. We can see that the first root has a very small coefficient in front of i. This means that we can ignore the imaginary part of this number. Thus, the first root is –0.072. The second and third roots are imaginary numbers and are complex conjugates, -0.49+0.49i and –0.49-0.49i. The last root is also a real number and is the one positive root we are looking for, 1.044.

Experiment Results Age-Specific Selection

		Week 1	2	3	4	5
# of surviving	males	50				
	females	50				
Fecundity	1					
Vials#	2					
	3					
	4					
	5					
	6					
	7					
	8					

	9					
	10					
	11					
	12					
	13					
	14					
	15					
	16					
	17					
	18					
	19					
	20					

Questions for the Lab Report

1. Compute the age-specific survival probabilities and fecundities for your population. Remember fecundity and survival have been estiomated and the uncertainty in these estimates can be summarized with confidence intervals.

2. Estimate the exponential rate of increase, or fitness, for your population. Since most of the first two weeks of life were spent as a larva or pupa in these populations and are combined into the first age-class, assume $f_1 = 0$ and $P_1 = 1$.

3. Using your own data, find the change in the exponential rate of increase that would result from doubling the female fecundity (i) at two weeks of age (e.g. f_2), AND (as a second calculation) (ii) doubling at five weeks of age (e.g. f_5).

4. What happens to average female fecundity with increasing age? What is this change due to?

References

Charlesworth, B. 1980. *Evolution in age-structured populations*. Chapters 1,3.

What Are Statistics?

Samples and Populations

Studies are largely based on sets of *individual observations* (or just observations). Such observations generally consist of a single measurement or set of measures taken on the "smallest sampling unit" of the study. A set of such observations is referred to as the *sample* under consideration. A set of blood pressure measures obtained from 100 patients is a sample, with each blood pressure measure being an observation. Similarly, if we counted the number of queen ants in each of 500 colonies, the number of queens in each colony is an observation and the entire set of 500 such counts the sample.

The actual measurement taken on each sampling unit is referred to as a *variable*. This variable could be a single measure (like blood pressure), a vector of observations (blood pressure taken a different times after a drug was administered), or even a more complex set of observations (a reconstructed three-dimensional CAT scan).

A third useful concept is the *population*. In statistics, the population is the entire collection of "things" we wish to learn about. Generally, we try to draw a sample from a population in a manner that is representative, such that inferences made from the sample apply to the population. Thus, a sample of 1000 doctors taking one aspirin tablet per day for three years and then being monitored for heart disease may be representative of the population of all doctors, or perhaps all Americans. Of course, the trick is picking a sample that is truly representative of the group for which conclusions are meant to be drawn. It is possible for the sample to also be the population. Such an example may be the weights of all animals in captivity for some species that is extinct in the wild.

Variables

Generally, experiments (and all data) differ in the forms they take. The variables we will consider fall into a number of useful categories:

Measurement Variables
- Continuous
- Discontinuous

Ranked Variables

Attributes

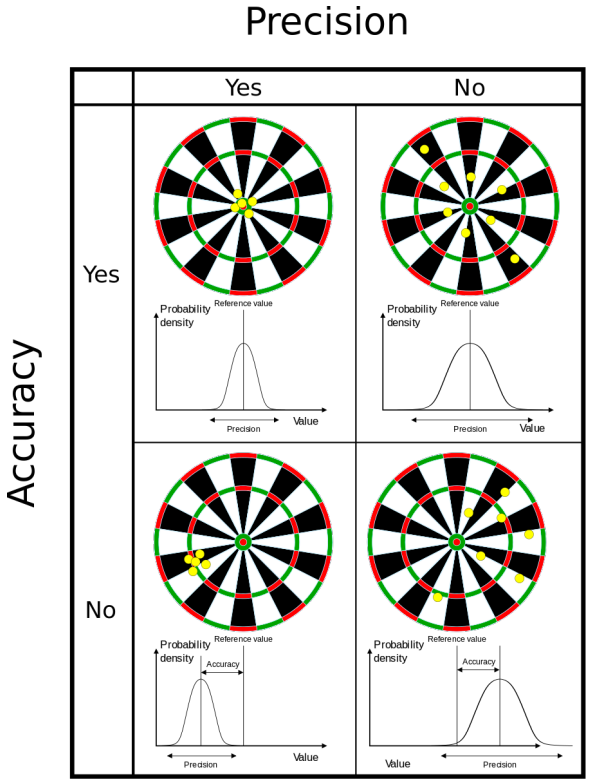
In theory *continuous variables* are measurements that can assume an infinite number of states (perhaps bounded). Classic examples are height and weight -- a person's height or weight can be measured to arbitrary precision depending only on the availability of an appropriate tool for measuring. *Discontinuous variables* on the other hand can only take on fixed values. Examples include the number of sternopleural bristles on a fly (my favorite!), or the number of eggs in a clutch of birds. *Ranked variables* imply order but not scale. One, for example, could record the birth order of a large family and the time between the 6th and 7th offspring's birth could be one minute or five years. The last category of variable is an attribute. *Attributes* are measurements which can only be expressed qualitatively. Attributes can include biological sex, coat color pattern in cats, or blood groups in humans. Attributes are also commonly associated with experimental manipulation. An example of this would be "Nasonia Wasps Treated with Ampicillan (to kill Wolbacia)" versus those "Untreated".

Accuracy and Precision

Accuracy is the closeness of a measure to its true value.

Precision is the closeness of repeated measures to the same value.

A watch that is STOPPED is very precise, but completely inaccurate!



Descriptive Statistics

The technical definition of a statistic is simply *a numerical summary of the data*. We will define several such summary statistics commonly used.

Let $X_1, X_2, \ldots, X_n$ represent a set of N observations from some experiment. X's could include numeric variables or indicator variables indicating discrete outcomes of a trial. Examples of numeric variables are weights of different people (a continuous quantity), or the number of bristles on certain segments of a fly (a discrete quantity). Examples of indicator variables can be success or failure outcomes of trials like determining whether the outcome of a coin toss is a heads or whether a draw from a deck of cards is a heart. Such outcomes can be recorded as 0's or 1's. Consequently, they can also be called binary variables.

Some useful statistics that describe our set of X's are the following:

Sample Mean $$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N}$$

That is, "X bar" is equal to the sum of all the X_i divided by the total number of observations.

Sample Variance $$Var_X = s_X^2 = \frac{\sum_{i=1}^{N} (X_i - \overline{X})^2}{N-1}$$

Sample Standard Deviation $$s_X = \sqrt{s_X^2}$$

The sample standard deviation is a measure of the average deviation from the mean value of the population. For example, is the average height of a population of male college students is 180 cm with a sample standard deviation of 15 cm, this can be thought of as a typical male student is within 15 cm of 180 cm.

If instead of having a single measure for each experimental unit, we took pairs of measures, then we can define some additional useful statistics. Examples of pairs of measures would include measuring both the pulse rate and height of each individual in a study. In this case, data would take the following form: $\{X_1, Y_1\}$, $\{X_2, Y_2\}$, ..., $\{X_n, Y_n\}$. Here X_i is the pulse rate of individual i and Y_i that individual's height. And we can define some additional summary statistics:

Sample Covariance $$Cov_{XY} = \frac{\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})}{N-1}$$

Sample Correlation $$r_{XY} = \frac{Cov_{XY}}{s_X s_Y}$$

The sample correlation is a number between -1 and 1 which measures the strength of the association between two variables in the study. Numbers close to +1 imply the large values of X are strongly predictive of large values of Y (and vice versa), value of -1 imply large values of X are strongly predictive of small values of Y (and vice versa) and intermediate values of r imply the value of X is a poor predictor of the value of Y. We would expect height in centimeters and height in inches on the same set of individuals to be highly correlated (r = 1), on the other hand we expect the height of each person in our class to show little or no correlation with the average time it takes to drive to work for that person (r = 0).

All these statistics can be calculated by hand or with a calculator. This isn't very efficient when one has a lot of data. There are several computer programs that allow these statistics to be calculated easily. We will learn how to use one such program, called R.

Why Do I Want to Use Statistics?

Statistics in and of themselves are not terribly useful. The term statistics is used in a much broader context to refer to a "statistical test of a hypothesis". That is, after we observe the outcome of some experiment, we wish to ask the question: “Is this outcome consistent with chance alone or is there something else going on here?” Generally, scientists set up tests of hypotheses. That is, they may ask if their results are consistent with some predetermined model. If not, they may use their results to reject this model. Alternatively, they may ask if their data differ with respect to an important variable after some experimental manipulation. These are not very intuitive

concepts for many people. In this course, we will motivate statistics with concrete examples.

Coin Toss (The Binomial Distribution)

You want to test the hypothesis that a coin you have is "fair". Thus, following your scientific spirit, you decide to do a coin toss experiment. In a more statistical language, you want to test the null hypothesis (usually designated H_0) that the probability of a head is equal to the probability of a tail, or fifty percent. The alternate hypothesis (usually designated H_1) is that the probability of a head does not equal the probability of a tail. Note that the two hypotheses are mutually exclusive and exhaustive. That is, if H_0 is true H_1 is false and vice versa, and the two hypotheses include all possible outcomes. Formally you would write this as:

H_0 : Prob(Head) = Prob(Tail) = 0.5
H_1 : Prob(Head) ≠ Prob(Tail)

You test this hypothesis by flipping a coin 20 times and each time writing down the outcome. Imagine that you observe 9 heads and 11 tails, and therefore you conclude that the probability of a Head is 45% (9 out of 20). Is this consistent with H_0 or H_1? It is not obvious. Why? What should you conclude? Suppose you had observed only 5 Heads out of 20, what then?

Luckily for you (and all of us), statisticians have given this problem a great deal of thought. They realized that in 20 trials of a coin toss experiment there is some probability of observing 0, 1, 2, ..., 20 Heads out of twenty. These probabilities can be worked out from the elementary rules of probability theory, but will not concern us here. It turns out that your experiment is a special case of the more general probability distribution referred to as the Binomial Distribution:

$$\Pr(S;N,p) = \frac{N!}{S!(N-S)!} p^S (1-p)^{N-S}$$

This equation is read as the Probability of S successes given N trials when the probability of a success equals p. This is the "distribution" of S conditional on N and p. The above case of the binomial distribution applies to the case where we perform **a number of trials** of some experiment and **observe one of two mutually exclusive and exhaustive outcomes** and we wish to test our observed number of "successes" to that expected under an assumed probability of success.

In your experiment, we define a success as a Head that occurs with a probability of 0.5 under the null hypothesis. Below we have plotted the *probability* of seeing exactly the specified number of heads out of 20 tosses (the bars, scale on left). We have also plotted the *cumulative probability* of seeing that number of heads or fewer (the line, scale on right). From the figure, we can see that the probability of observing exactly 9 Heads is about 17% (draw a horizontal line to the scale on the left for the bar at 9 heads). On the other hand, the probability of observing 9 or fewer heads is about 40% (draw a horizontal line to the scale on the right for the line at 9 heads). Thus, your observation of 9 Heads does not appear to violate the null hypothesis. Had you observed 5 Heads the situation would be different. In this case, the probability of observing 5 or fewer Heads is less than 5% (the right y-axis and line in Fig. 1). Thus, it seems unlikely the null hypothesis is true, and the coin is biased.

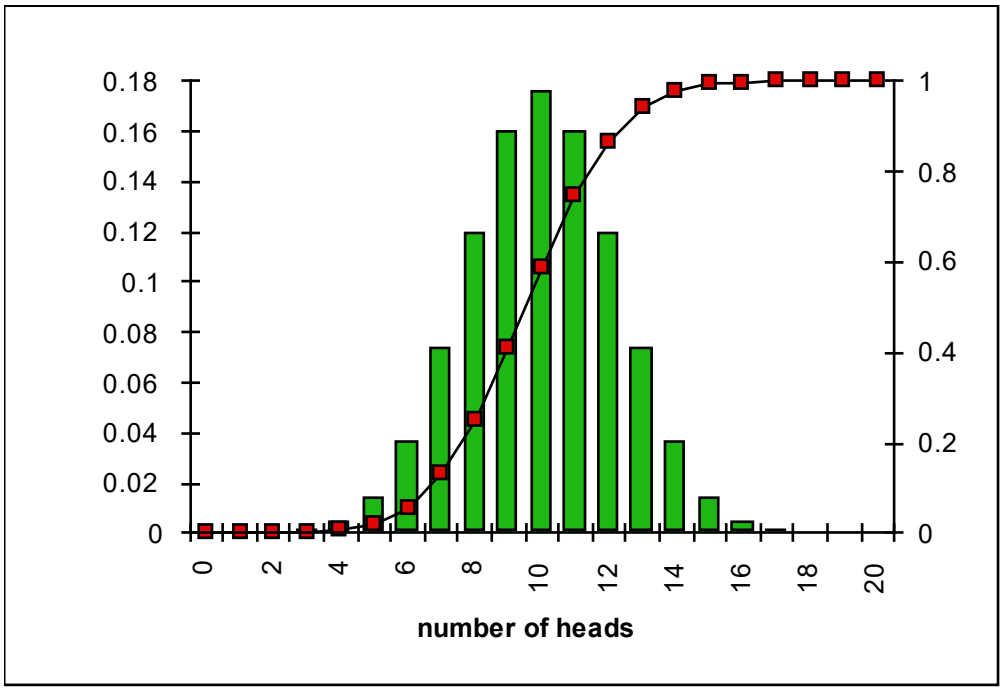


Figure 1: The binomial probability distribution and cumulative distribution, N = 20 and p =0.5.

So what is "statistics" really about? Simple.

1. You set up a null hypothesis and an alternative hypothesis.
2. You record your data and then calculate some "statistic" or summary of the data.
3. You count on a statistician having determined the distribution of your statistic given that the null hypothesis is true.

4. You look and see where the observed value of your statistic falls in that theoretical distribution. You ask if your observed value is likely to have occurred by chance alone under the null hypothesis?

The trick is to pick a null hypothesis (1) and statistic (2) for which the distribution is worked out!

Is sex-ratio influenced by rearing temperature in turtles? (Chi-square Distribution)

In many species of turtles the sex-ratio of offspring hatching from a given nest is affected by the temperatures experienced during development in the nest. Molly, a hypothetical highly esteemed turtle biologist, discovers a new species of turtle, *Chrysemys gobruinsentius*. She decides to carry out an experiment to determine if sex-ratio is affected by rearing temperature. She rears a cohort of turtles at either 16C or 25C and counts the number of turtles of each sex that hatch.

OBSERVED DATA

Temp	**16C**	**25C**	**Total**
Male	50	70	120
Female	50	20	70
Total	100	90	190

Molly's null hypothesis is that sex-ratio is independent of temperature. That is 50/100 is statistically indistinguishable from 70/90. Her alternate hypothesis is that the two ratios are unequal.

H_0: sex-ratios are independent of temperature
H_1: sex-ratios are different

It turns out that there is a convenient statistic, called the Chi-square statistic, which summarizes the information in the table, and whose distribution is known if the null hypothesis is true. The statistic is:

$$\chi^2_{df=x} = \sum_{cells} \frac{(obs_i - exp_i)^2}{exp_i}$$

where *df* is the degrees of freedom in the model, obs_i is the number of observations in each cell and exp_i is the expected number of observations in each cell under the null hypothesis. We will discuss degrees of freedom later, cells refer to a particular combination of sex and rearing temperature (e.g., males hatching at 16C). If *N* is the total number of observations over all cells, then the expected number in each cell under the null hypothesis

can be easily calculated. In the case of Males hatching at 16C this is just the total proportion of males times the total proportion of turtles hatching at 16C times *N*; or

(#Males/N) × (#16C/N) × N = (120/190) × (100/190) × 190 = 63.2

The expected numbers can be calculated similarly for the remainder of the cells. With the resulting table

EXPECTED DATA

Temp	**16C**	**25C**	**Total**
Male	63.2	56.8	120
Female	36.8	33.2	70
Total	100	90	190

The Chi-square statistic can be calculated as the sum of the observed counts minus expected counts squared divided by expected counts over all cells of the table (as per the formula above). In this example, the resulting value of the Chi-square statistic is 15.7. The degrees of freedom are calculated as the total number of cells minus the "number of constraints placed on the expected cell totals". In this case, there are three constraints placed on the expected cell values: the total number of observations in the entire data set, the total number of Males, and the total number of turtles hatched at 16C. That is, once you are given these three values *(i.e.,* conditional upon these three values) all the expected values are completely determined (convince yourself or this). So, in this example, there are 4-3 = 1 degree of freedom (this will generally be the case with 2X2 tables). One minus the cumulative Chi-square distribution looks like this for one degree of freedom. That is, each point on the line can be thought of as the probability of observing a Chi-square statistic larger than that value. The horizontal line shows the 5% level, or the value of the Chi-square statistic for which we only expect to observe larger values if the cell counts are independent five percent of the time.

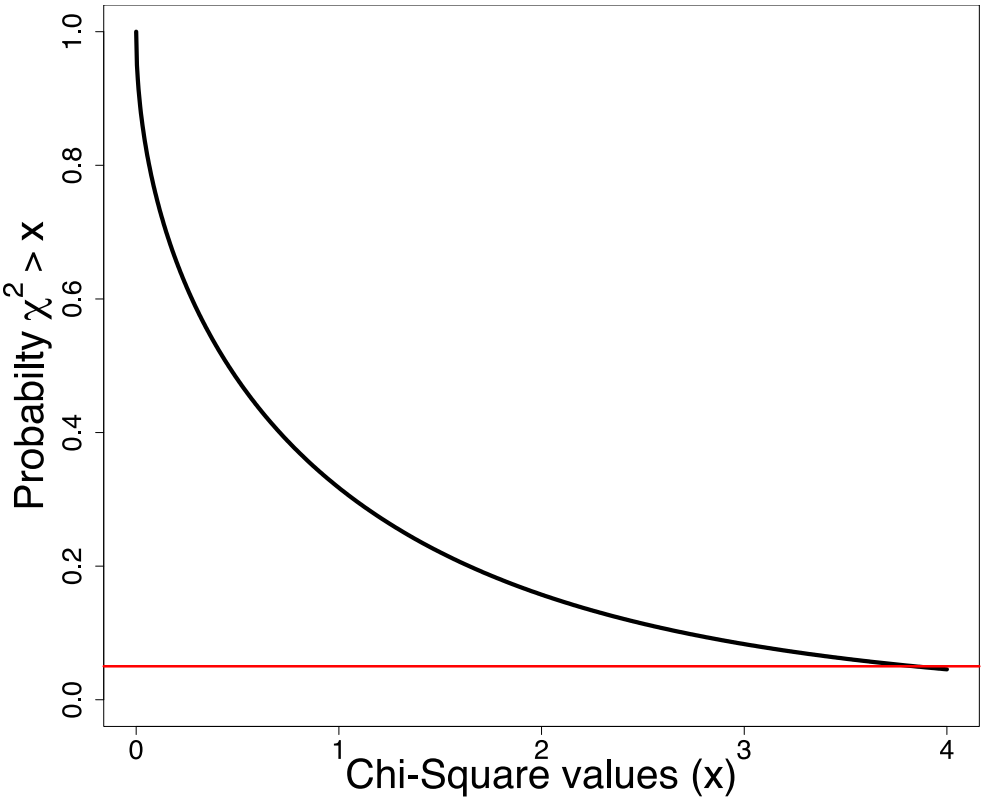


Figure 2: One minus the chi-square cumulative distribution with one degree of freedom. The red horizontal line shows the 5% probability level. Notice that 15.7 would fall far to the right of the figure.

Thus, it is apparent that the probability of observing a number as extreme as 15.7 is quite rare by chance alone (for values as low as six the probability is approaching zero). The implication is that we can reject the null hypothesis that the sex-ratios are independent of rearing temperature as the probability of observing Molly's data is very low under the null hypothesis. Molly is confident that rearing temperature affects sex-ratio in her newly discovered turtle species.

Confidence Interval on a Variance Estimate

The Chi-square distribution is used in a number of different contexts in statistics. For example, it turns out that

$((n-1)\ s^2)/\sigma^2$ is distributed as a Chi-square with *n*-1 degrees of freedom where s^2 is the estimated observed variance of a sample of size n from some population and σ^2 is the (unknown) true variance of the distribution from which a sample is taken. We will state without proof that it follows that

$\left(\frac{(n-1)s^2}{q_2}, \frac{(n-1)s^2}{q_1}\right)$ is a 95% confidence interval on σ^2, the true variance from which the sample was drawn, provided that

$\Pr(\chi^2_{df=n-1} < q_1) = 0.025$ and $\Pr(\chi^2_{df=n-1} < q_2) = 0.975$. This property of the Chi-square distribution can be useful for determining if the variance of a sample differs from a theoretical prediction. The only "trick" required to calculate a 95% confidence interval on a true (unknown) variance is to calculate values for q_1 and q_2. These q's are referred to as *quantiles* of a statistical distribution, and can be thought of as the value of the "x" axis conditional on the value of the "y" axis in cumulative distribution plots such as Figures 2 and 3.

Below is a cumulative Chi-square distribution for degrees of freedom = 99 (i.e., the sample variance estimated from 100 observations). The red horizontal lines are at 0.025 and 0.975. It can be seen from the figure that the a value of q_1 approximately equal to 75 satisfies the first equation ($\Pr(\chi^2_{df=n-1} < q_1) = 0.025$) and a value of q_2 approximately equal to 125 the second equation ($\Pr(\chi^2_{df=n-1} < q_2) = 0.975$). That is, the values of q_1 and q_2 are obtained by taking the value of the x-axis where the red line intersects the cumulative Chi-square distribution for $\Pr(\chi^2_{df=99} < x)$. Later in the course we will use a computer program to obtain these quantiles.

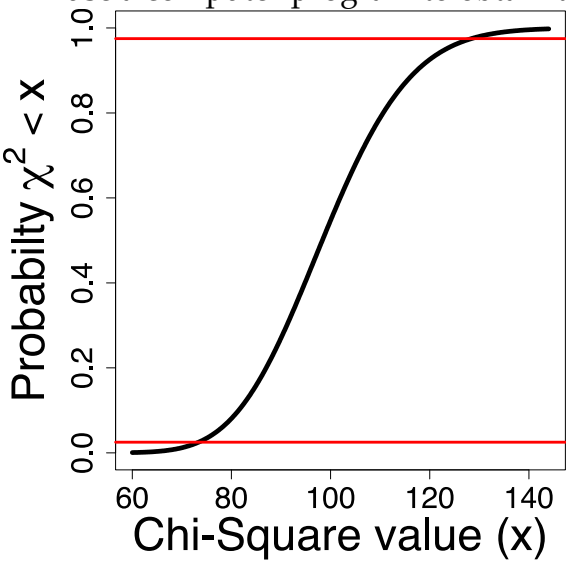


Figure 3: Cumulative Chi-square distribution for degrees of freedom = 99. Horizontal red lines are at 0.025 and 0.975.

Desiccation resistance in flies selected to resist starvation (t-Distribution).

Many organisms have genetic mechanisms that allow them to respond to "stressful" situations (think high temperatures or food shortage as opposed

to a term paper being due tomorrow). One hypothesis regarding the genetics of stress resistance is that there are general mechanisms of responding to stress, such that an organism evolved to counter one stress is often good at tolerating a second "unrelated" stress. In order to test this hypothesis a hypothetical scientist, Parvin, creates 10 evolved populations of *Drosophila*, along with a set of 10 matched controls. Each evolved population and its matched control are derived from a different base population. The selection is carried out by placing flies in vials without food (but with water) until 80% of the flies die, and then allowing the survivors to reproduce. After 25 generations the evolved populations are qualitatively better at surviving starvation then the controls.

In order to examine "cross-tolerance" of her evolved populations Parvin carries out an experiment where both the starvation evolved lines and their controls experience desiccation stress. This is accomplished by placing 500 flies in a bottle in the presence of a desiccant, and then measuring the time in hours until all the flies are dead. Parvin believes that the experiment primarily measures desiccation resistance, as the survival times are an order of magnitude shorter than would be observed in the case of a purely starvation stress. Below are the data she observes.

Survival times in hours

Population	A	B	C	D	E	F	G	H	I	J
Control	8.4	8.1	5.1	7.6	4.7	10.7	5.7	4.1	8.1	6.8
Starvation-resistant	12.4	15.8	11.7	8.6	12.6	11.1	10.5	7.3	7.2	10.8

In Parvin's case the null hypothesis is that the means of the two treatments are the same, and the alternate is that they differ:

H_0: mean desiccation resistance of control populations equals that of the evolved populations
H_1: mean desiccation resistance is different

It turns out there is a convenient statistic, the *t*-statistic, whose statistical properties are well known. The *t*-statistic is:

$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{n_2} + \frac{s_2^2}{n_1}}}$, where X with a bar over it is called "X-bar" and represents the sample mean of each group and s^2 the sample variance of each group, and n is the number of observations in each group (this is a special case of the *t*-statistic for the case in which the variances of the two groups are fairly similar, which is all you have to worry about). This statistic is distributed according to a *t*-distribution with $n_1 + n_2 - 2$ degrees of freedom, and the cumulative distribution looks like this:

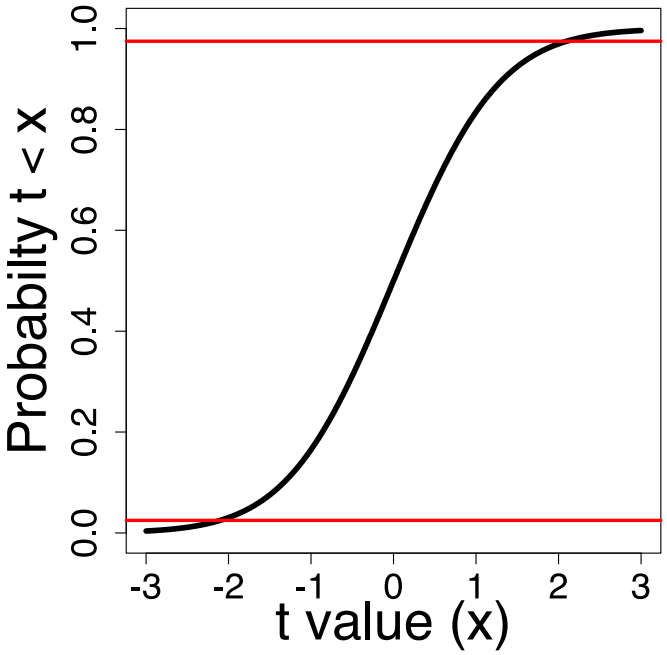


Figure 4 The cumulative *t*-distribution. As before, red horizontal lines are at 0.025 and 0.975.

In Parvin's case the value of the *t*-statistic is -3.69. Less than 1% of the t-distribution falls below the observed *t*-statistic measure in Parvin's case. Thus it appears that Parvin can reject the null hypothesis that selection for starvation resistance results in NO cross-tolerance to desiccation resistance.

What Does it Take to Reject the Null Hypothesis?

By convention, if the value of some statistic is such that fewer than 5% of observations are likely to exceed it by chance alone the null hypothesis is rejected at a "*p*-value" of 0.05. Thus a *p*-value of 0.05 or 5% is equivalent to the statement that only 5% of the time would one observe this data or data

more extreme under the null hypothesis. In the case of the example above we did a two-tailed *t*-test (meaning we just wanted to see if desiccation resistance was different between the two groups). Thus, for the above degrees of freedom and looking at Figure 3 the *t*-statistic would have to be either larger that approximately 2.1 or smaller than -2.1 to be significant (the values of the x-axis, where horizontal lines at 0.025 and 0.975 cross the cumulative distribution).

Confidence Interval on the Sample Mean

Suppose we have a sample of consisting of *n*-observations, x_1, x_2, …, x_n. From these we can easily calculate the sample mean, $\bar{x} = \frac{1}{n}\sum_{i=1}^{i=n} x_i$, and variance $s^2 = \frac{1}{n-1}\sum_{i=1}^{i=n}\left(\bar{x} - x_i\right)^2$. A (1-α) × 100% confidence interval on $\bar{x}$ is equal to $\left(\bar{x} + t_{\alpha/2,n-1}\sqrt{s^2/n}, \bar{x} + t_{1-(\alpha/2),n-1}\sqrt{s^2/n}\right)$, where $t_{a,N}$ is the value of a *t* random variable with *N* degrees of freedom that is greater than $a \times 100\%$ of all such random variables. The value of this *t* variable can be found using R (see next chapter) with the command: `qt(a,N)`.

R: A Language for Doing Statistics

We will use a program called R to help us do statistical calculations and later simulate more complex phenomena. What follows is an introduction to R designed with the information you need to do your problems.

Why R

R is a freely available flexible statistical language. It is an Open Source language, which means that computer programmers anywhere in the world can modify it when a bug is found. R is object-oriented and has a set of powerful graphic tools.

Links

Download R
http://cran.r-project.org/
Introduction to R
http://cran.r-project.org/doc/manuals/R-intro.pdf
A second Introduction to R
http://cran.r-project.org/doc/contrib/Rdebuts_en.pdf
One page R "cheat sheet"
http://cran.r-project.org/doc/contrib/refcard.pdf

Starting and Quitting R

Throughout this introduction commands typed into R will be in a different (Courier) font. Everything in this (Times) font is an explanation of what you are doing.

To get R just download and install R and RStudio from the following two websites:
https://cran.rstudio.com/
https://www.rstudio.com/products/rstudio/download/

Everything in R is an object. q followed by parentheses executes the quit function, whereas q without brackets lists the contents of an object called 'q', which should be the q() function . This will be true of all the functions and objects you encounter.

Try typing "q" without parentheses. Try typing a capital "Q".

> q
> Q
> Q()

There is a differences between asking for the contents of q and executing q as a function. R is also "*case sensitive*" meaning in general *X* does not equal *x*.

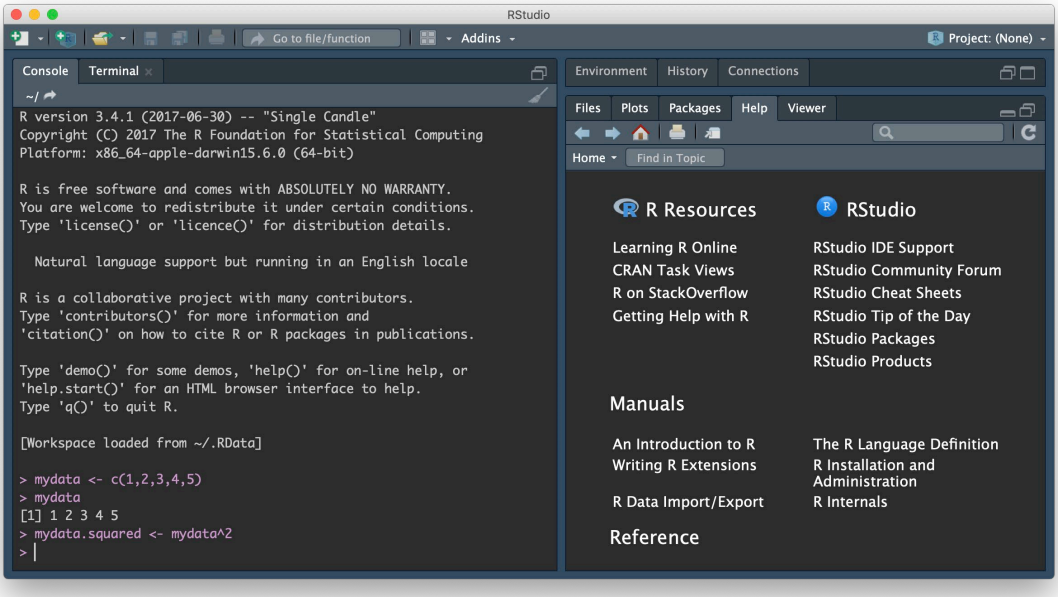
Before proceeding further, view the "Introduction to R" video series (link below). As you watch it, make sure that you follow along in RStudio and perform the same actions as in the video. You will not learn without doing. Be aware that some of the URLs referenced in the video may have changed since the video was uploaded.

Introduction to R:
http://bit.ly/E115L_LearningR

Some Useful Functions

Before continuing, **WATCH THE VIDEO PLAYLIST** above! Now, switch to RStudio and let's look at a few useful functions. First, notice the help tab on the right-hand side (it was in the videos). Next, type the following two commands into the window (omitting the literal '>' signs):

```
> mydata <- c(1,2,3,4,5)
> mydata
> mydata.squared <- (mydata)^2
```



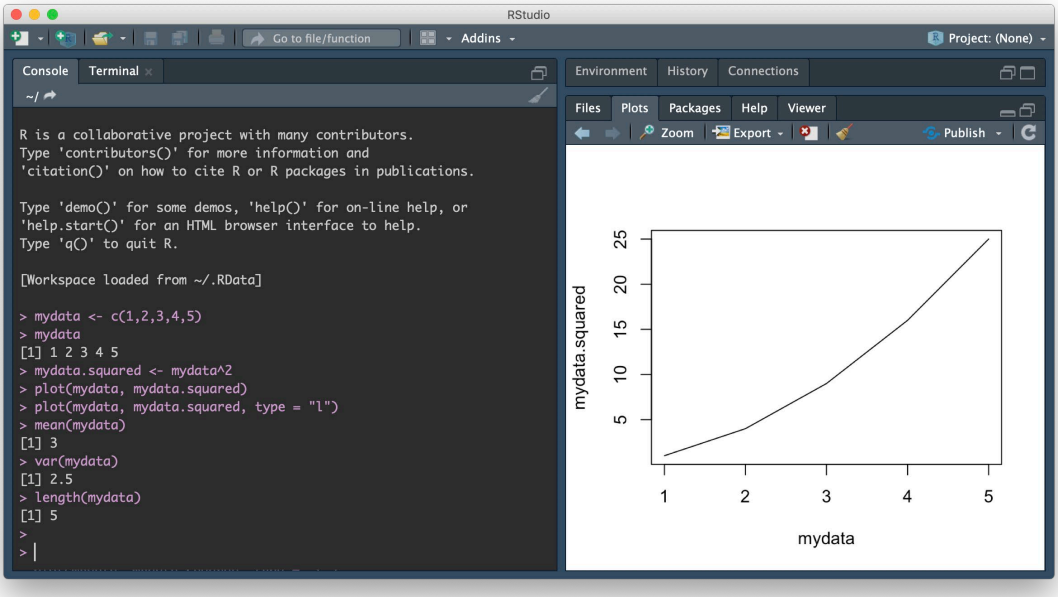
The first line makes an object called mydata which contains the listed set of numbers. Later we will learn how to import data. `mydata` is just a arbitrary label for where we will put the data. The `c(...)` tells **R** to concatenate the number in the parentheses. The arrow tells **R** to put whatever is on the right side into the variable on the left side. `mydata` is a *vector* of observations. In **R** vectors are series of data of one "kind" in one dimension (their length). For example, you cannot have a vector with text and numbers or **R** will convert the numbers into text.

The second line lists the contents of mydata. The third line squares each value in mydata. mydata.squared again is a new variable name where we will put something (in this case the squared values of mydata).

Try typing the following into RStudio:

```
> plot(mydata, mydata.squared)
> plot(mydata, mydata.squared, type="l")
> mean(mydata)
> var(mydata)
> length(mydata)
```

The first two lines make an xy-plot of the data. The plot command is of the form `plot(x-axis, y-axis, options)`. Specifying type = "l" is a line plot. The final three lines give the mean, variance, and number of observations in mydata, repectively. Notice how the plot now replaces the help menu.



Below is a table of commonly used functions and operators

`+`	addition
`-`	subtraction
`*`	multiplication
`/`	division
`^`	exponentiation
`abs`	absolute value
`exp`	exponential (*e* to a power)
`log`	natural log
`log10`	log base ten
`sqrt`	square root
`cor`	correlation between two vectors
`cumsum`	cumulative sum of a vector
`mean`	mean
`median`	median
`min`	minimum
`max`	maximum
`sum`	sum
`var`	variance (or covariance, if given a matrix)

It is useful to see what these functions do to your data. Try some of them out.

A few important notes

1) Write your code in a separate file (not in the console)

2) Annotate your code so that you remember what you did and why

To annotate code, you just add **#** before any line of text, that way R knows that it is not an instruction. For example, type:

```
> This is just an instruction
```

What happened? Now try:

```
> #This is just an instruction
```

Vectors

Vectors are a series of objects (it can be of length 1) in R. They have only one dimension of a certain "length". You can ask R the length of a vector with the command `length()`.

```
> mydata <- c(1,2,3,4,5)
> length(mydata)
```

Note: We can ask if our R object is a vector with `is.vector()`. In R, functions (we will learn more about them) are always followed by a set of parentheses to give arguments for the function.

Vectors can contain many elements. These are some ways to create vectors in R:

```
> v1 <- c(2,4,6,8)
> v2 <- 1:12
> v3 <- 1:4
> v4 <- seq(0,50,1)
> v5 <- rep(7,9)
```

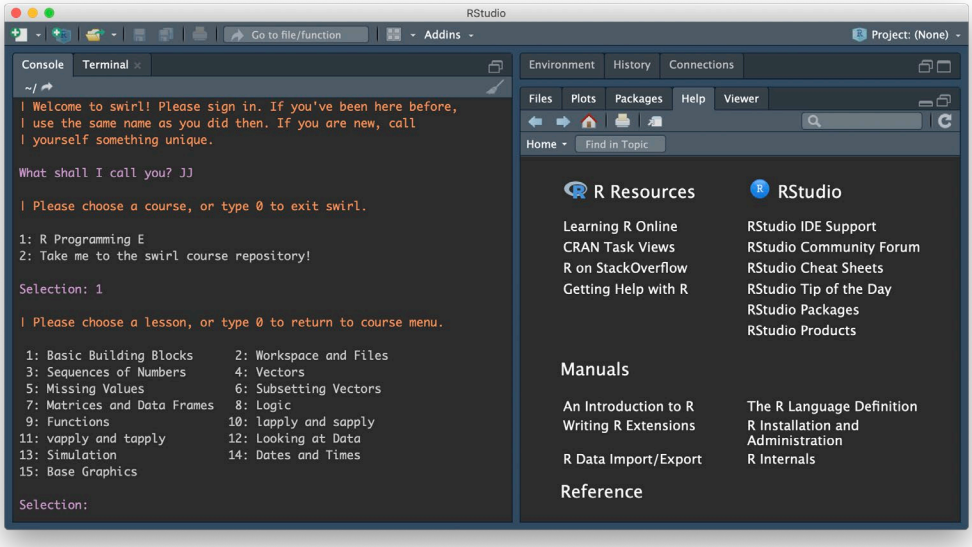
What is each of these functions doing? Could you describe it? What do you think will happen if you type v1+v3?

```
> v1+v3
```

Getting up to Speed with R Using swirl()

By now, you should be able to do a number of basic tasks in R by using RStudio. However, there are a lot of basics that are necessary to understand to continue. And the rest of the course relies heavily on these concepts. Perhaps the most important prep work you'll do for the remainder of the course will be completing the swirl() tutorial. This is an interactive R tutorial *taught in R*! To access it, first open RStudio, and type the following into the Console:

```
> install.packages("swirl")
> library(swirl)
>install_course_github("swirldev", "R_Programming_E")
> swirl()
```



As usual, omit the ">" symbols, as they represent the console prompts. Now, the RStudio console will prompt you for your name and ask you which course you'd like to take (see screenshot above). Enter the number associated with "R Programming E" (it was 1 for me) and press return. For this course, you should complete the whole tutorial except for possibly the "Dates and Times" section. We won't be working on that in class. The tutorials are short, interactive, and very directly relevant to completing the class. At the very least you should complete the tutorials to convince yourself that you know enough to complete the course. I will be reviewing the major concepts in class, but it will be a review. If you haven't reviewed the material ahead of time, it will be hard to learn in class.

Matrices

Often, we will work on matrices instead of vectors. A matrix is a representation of numbers in the form of a table with some number of rows and columns. Generally a row of a matrix will be observations and columns different variables. We will now generate a matrix. First, we need to make an additional vector:

```
> mydata <- 1:5
```

this is shorthand for generating a sequence of numbers from 1 to 5

```
> cbind(mydata, (mydata)^2)
```

cbind stands for "column bind", and will bind columns together to make a matrix. Another function, `rbind`, will bind rows. Say we want to do further work on this matrix. To do so, we must make it an object. How do we do this??

Try:

```
> mymatrix <- cbind(mydata, (mydata)^2)
> mymatrix
```

Try applying some of the functions above to your new matrix. What do min, max, and mean do? What about cor and var?

You can see that many functions don't really work the way you might like them to. It would be useful to have a way to refer to only parts of mymatrix. We can do this fairly easily.

Try the following, what happens

```
> mymatrix[1,1]
```

Note the first number refers to the row number and the second the column number.

```
> mymatrix[3,2]
> mymatrix[,2]
> mymatrix[3,]
> mymatrix[1:3,2]
```

What happens when you apply functions to these "submatrices"??

Now try this:

```
> mymatrix[mymatrix[,2] > 16,]
```

This is pretty complex, but powerful. Do you see what we accomplish by using this command?

What is the variance/covariance matrix associated with the first three rows of `mymatrix`?

Scripting R procedures

Two common complaints about R are: 1) that it is difficult to learn the syntax of the language, and 2) why use a command line interface when we are accustomed to "point and click" in the other programs we routinely use. While the syntax can indeed be difficult to pick up, it more than makes up for this challenge. One advantage of a command line interface is incredible flexibility. If the "correct" procedure or plot you want is not easily generated using point and click type defaults it can be modified at the command line. Sometimes it is painful to do so, but it is at least possible.

More importantly it is possible to save all the commands you used to produce a figure or analysis and then repeat that procedure exactly on the same dataset, the same dataset with a subset of observations changed (say it turned out that 15% of a sample of pH readings making up your dataset were collected with a borrowed instrument that you discovered was imprecise), or an entirely different dataset. If the same set of operations are often applied to different datasets it is possible to even write your own custom function to do this procedure (below). This is very useful the day you want to re-generate a certain *p*-value or figure and you cannot remember the exact sequence of point and click operations that allowed you to do this....

The easiest way to do this is to work in RStudio's script editor. Re-running an analysis merely requires pasting the "code" back into a R window where it is executed one line at a time.

As an example, consider the commands below taken from the earlier examples. I have edited these earlier commands by merely removing the "**>**" prompt. Try pasting them into the R window.

```
mydata <- c(12,19,24,17,15)
mydata.squared <- (mydata)^2
plot(mydata, mydata.squared)
mean(mydata)
var(mydata)
mydata2 <- 1:5
mymatrix <- cbind(mydata2, mydata)
mymatrix[1:3,2]
mymatrix[mymatrix[,2] > 16,]
```

Now let's imagine we discovered the first data point of mydata was not a "12" but a "21" because of a transposition error entering the data. We could correct our mistake easily.

```
mydata[1] <- 21
mydata
```

We want the values of mymatrix for which the values in the second column are greater than 16. What happens if we just re-run the last line of the above script?

```
mymatrix[mymatrix[,2] > 16,]
```

It doesn't work correctly, as we have to re-run all the intermediate steps (such as creating mymatrix) to reflect the changes to mydata. So we can re-run our analysis on mydata by re-running all the lines subsequent to the initial "mydata" assignment. Do this by highlighting those lines and clicking on the "Run" button in the RStudio script editing window.

*In the remainder of the notes I leave out the prompt "**>**" sign so you can directly paste examples into R. When a line of code is indented, under a preceding line, that's an indication that it is a continuation of the previous line, and represents a single line of code.*

Importing

The best way to input data into R is as a "tab" delimited table. If data is saved such that the first line is a set of tab separated names and subsequent lines are the data (each with a row label) then the function `read.table` can be used. For example say you have the following data saved as

"house.data" in the "desktop directory" on your PC (to get the file to your "desktop" go to the file server and drag the appropriate file to the "desktop"). Now in R change your working directory to the "desktop" (under "FILE" -> "CHANGE WORKING DIRECTORY"):

	Price	Floor	Area	Rooms
01	52	111	830	5
02	54.75	128	710	5
03	57.50	101.5	1000	6
04	57.50	131.0	690	6
...				

This data can be read into R using the following command:

```
Housedata <- read.table("house.data")
```

What do you think the following command does?

```
Housedata[,"Floor"]
```

R has another way to refer to parts of a "data.frame" such as "Housedata" using a "dollar sign" notation.

```
Housedata$Floor
Housedata$Floor[1:3]
Housedata$Floor[Housedata$Floor > 115]
```

Applying functions to rows or columns of a dataframe or a list

Lets load the decidedly non-biological example "mtcars" data set into R

```
data(mtcars)
mtcars[1:10,]
dim(mtcars)
```

For each car we have the following variables that are recorded: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. It is obvious what some of these variables are.... others your guess is as good as mine. Say we wanted to calculate the mean and minimum observations for each of the recorded variables

```
mean(mtcars)
min(mtcars)
```

Interestingly the function mean is "smart enough" to know mtcars is a dataframe and calculate the mean for each column, min is not nearly so smart. There is a function "apply: that lets one apply some operation to each row or column of a matrix (or dataframe). The first argument to apply is a matrix or dataframe, the second to apply the operation to rows or columns (rows = 1, columns = 2), and the third the function to be applied. The function can be a built in function taking a single argument (like min), a custom function, or defined within the apply command:

```
apply(X = mtcars, MARGIN = 2, FUN = min)
apply(x = mtcars, MARGIN = 2, FUN = function(x)
     (100*sd(x))/mean(x))
# coefficient of variation
```

It is not difficult to calculate such statistics conditional on the state of one of the other variables. For example say we want the average statistics for each of the 4 cylinder cars.

```
mtcars.4cyl <- mtcars[mtcars[,"cyl"] == 4,]
apply(X = mtcars.4cyl, MARGIN = 2, FUN = mean)
```

In fact, we can calculate statistics of interest conditional on the number of cylinders using the "split" function in concert with the "lapply" function. "split" will split a dataframe by a given variable into a "list" of dataframes, each member of the list being a dataframe subselected by the slitting factor.

{More advanced sub-setting commands are discussed below}

```
mtcars.cyl <- split(x = mtcars, f = mtcars["cyl"])
#  apply the min and mean to each element of the list
     mtcars.cyl


lapply(X = mtcars.cyl, FUN = function(x)
     apply(x,2,min))
lapply(X = mtcars.cyl, FUN = function(x)
     apply(x,2,mean))
#  we much nest the apply within the lapply since each
#  element of mtcars.cyl is a dataframe itself!
```

"tapply" is similar to "lapply" and is useful when tabling results.

```
tapply(X = mtcars$mpg, INDEX = mtcars[c("cyl", "am")],
     FUN = mean)
```

It appears 4 cylinder cars with a radio are your best bet if you want to conserve fuel! (although perhaps this is not significant!)

"tapply", "lapply", and "apply" can be combined in a variety of manners to produce useful results (we will not worry so much about these complex operations). For example:

```
apply(X = mtcars[,c("mpg","disp","hp","cyl","am")],
     MARGIN = 2, FUN = function (x) tapply(X = x,
     MARGIN = mtcars[c("cyl", "am")], FUN = mean))
```

Although the results are readily apparent, I suspect the code is not so transparent. Luckily, nothing this sophisticated with be required to do the problems!

Making your own functions

In many situations you will do something often enough that it is useful to have your own function to simplify this task. In fact many of the functions you use in R are written in R. Let's look at a simple example that cubes a number

```
cube.it <- function(x) x*x*x
cube.it(3)
cube.it(-3)
```

Although this could have been accomplished using -3^3, it is instructive to look at making a function. This is generally of the form

```
yyy <- function(x1, x2, ...){

operations on xi's
final line in the object to be returned
}
```

And is called as

```
yyy(x1,x2,...)
```

The covariance of two data vectors X and Y is defined as

$$Cov(X,Y) = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{Y})}{n-1}$$

So we will make a function called "foo.cov" (we do not want to call is cov or it would replace the built in function for covariance!)

```
foo.cov <- function(x, y){

X.bar <- mean(x)
Y.bar <- mean(y)
Xi.minus.Xbar <- x-X.bar
Yi.minus.Ybar <- y-Y.bar
numerator <- sum(Xi.minus.Xbar * Yi.minus.Ybar)
answer <- numerator/(length(x)-1)
#  length is the number of observations in a vector

# answer we must include the last line so "answer" is
# returned
}
```

We can compare our function that calculates Covariance to the built in R routine

```
XX <- Housedata$Price
YY <- Housedata$Area
XX
YY
foo.cov(XX,YY)
cov(XX,YY)
```

or the same can be accomplished in one line, as:

```
foo.cov(Housedata$Price, Housedata$Area)
```

Graphics

The incredible flexibility of graphics in R, combined with publication quality postscript output is one of the primary reasons that many statisticians migrated to the R statistical language. As with statistical and mathematical operations in R, graphics can also be included in functions (or scripted). As a result very complex figures can be easily recreated if a data-set changed a small amount. Chapter 12 of the R-intro.pdf is a thorough introduction to graphics in R. We will give a much less detailed introduction here. If you really want to read this document, it can be found here: http://bit.ly/R-Intro-pdf

First lets get some data to plot. R has a number of built in data-sets.

```
data(package = base)  # list available datasets
data(faithful)        # load data of intervals between
                      # eruptions of old faithful

faithful[1:10,]
attach(faithful)
```

"eruptions" are the magnitude of observed eruptions and "waiting" is the waiting time between eruptions in minutes.

Lets look at a histogram of eruption waiting times

```
hist(waiting)
hist(waiting, breaks = 30)  # more bins
hist(waiting, breaks = 30, xlim=c(30,110))
# set xaxis limits
hist(waiting, breaks = 30, xlim=c(30,110),prob=TRUE)
# yaxis is now frequency versus counts
help(hist)
```

We can add lines to the histogram that estimate the "density" of eruption intervals (more about what this is later...for now think of it as a smooth curve to the data).

```
lines(density(waiting, bw=1))
```

What does changing "bw" do?
How would you find out other feature of "lines" or "density"?

```
lines(density(waiting, bw=5),col=2)
```

We can , of course make a similar plot for eruptions, try this (note you may have to make the "bw" parameter much smaller as I do below).

```
hist(eruptions, breaks = 30, xlim = c(1,5.5), prob =
     TRUE, col = 3)
lines(density(eruptions, bw=0.1), col = 2)
rug(eruptions)   # I also add the raw observations
```

You can save these plots from RStudio using the "Export" button above them. You can also do so by prefacing the plot commands with a "png" or "pdf" command and suffixing the plot commands with the "graphics.off()" command to plotting is redirected to the "screen".

```
png(filename = "eruptions.ps")
```

```
hist(eruptions, breaks = 30, xlim = c(1, 5.5), prob =
     TRUE, col = 3)
lines(density(eruptions, bw = 0.1), col = 2)
rug(eruptions)   # I also add the raw observations
graphics.off()

pdf(file = "tony1.pdf")
hist(eruptions, breaks = 30, xlim = c(1,5.5), prob =
     TRUE, col = 3)
lines(density(eruptions, bw=0.1), col=2)
rug(eruptions)   # I also add the raw observations
graphics.off()
```

It is interesting that both waiting and eruptions are "bimodal". Is it conceivable that longer waits between eruptions are associated with bigger eruptions?

```
plot(waiting, eruptions)
plot(waiting, eruptions, pch = 16, col = 2)
```

Can you change the "x" and "y" limits and add a title to the plot?

R makes constructing beautiful plots easy. Below is a "fun" example with the faithful data...more intended to "show off" R than teach you how to do this.

```
faithful.m <- apply(faithful, 2, mean)
faithful.sd <- apply(faithful, 2, sd)
fm <- apply(faithful, 1, function(x)
       (x - faithful.m)/faithful.sd)

#  transform both columns to have 0 mean
#  and variance of 1
#  we will learn how to do this later in the course!

image(as.matrix(fm))

# oooh, I don't like these heat colors
# turns out the color spectrum is easily manipulated

my.colors <- c(rgb(r=(30:15)/30,g=0,b=0),
               rgb(g=(15:30)/30,r=0,b=0))

image(as.matrix(fm), col=my.colors)

# or yellow -> blue for those R/G color blind males
```

```
my.colors <- c(rgb(r=(30:15)/30, g=(30:15)/30, b=0),
               rgb(b=(15:30)/30, r=0, g=0))

image(as.matrix(fm), col = my.colors)
```

Stronger yellows are associated with small eruptions and short waiting times, whereas stronger blues are associated with larger eruptions and longer waiting times.

For the data here with only two columns a scatter plot is likely to be more informative, but these image plots are of particular utility when the data is highly dimensional. They have become a very popular way of presenting gene expression patterns for thousands of genes over a number of different treatments. It seems they may also be of utility in other contexts...for example ecological data-sets in which a large number of species each have the same 10 measures.

ggplot

In addition to the already available capabilities of R for graphics, you can install an additional package that makes making beautiful plots really simple (once you get the hang of it). To start with ggplot we first need to install it:

```
> install.packages("ggplot2")
> library("ggplot2")
```

There is a lot of great documentation about ggplot online. For now, th important parts to know are that:

1. ggplot works by adding layers (each added with +)
2. It has "variables" (defined with 'aes') that vary according to the value (usually the columns in your data).
3. And it has parameters like a particular color or shape that are defined outside 'aes'.

Let's try it using the data "iris". This is a famous dataset introduced by the evolutionary biologist Ronald Fisher in his 1936 paper "The use of multiple measurements in taxonomic problems". Ronald Fisher was a man that developed a lot of useful tools and theories in statistics and evolution but unfortunately was also a eugenicist. Luckily for us, this dataset is much more innocent. The iris dataset contains three species of irises (setosa, virginica, versicolor) and four traits measured for each sample. All measurements are in cm.

Ok, let's try ggplot! The first layer just defines the space and overall variables. Each of our columns is one of our variables (x, y, color) and that is why they are defined within the `aes` parenthesis. They will change depending on the value of our columns.

```
> ggplot(data = iris, aes(x = Petal.Length, y =
     Petal.Width,   color = Species))
```

Now let's add some points and save our plot as an object to avoid copying the same many times.

```
> myplot <- ggplot(data = iris, aes(x = Petal.Length,
          y= Petal.Width, color = Species))+
geom_point()+
```

To make things clearer let's add a trend line. Ways of plotting the data (points, lines, boxplots) are 'geom' layers. Layers that summarize the data in some ways are often 'stat' (as in statistics) layers.

```
> myplot + stat_smooth(method= "lm")
```

If we want to have a trend for all species together, we can set the variable for color only in the points. We can change the color of the trend line outside the 'aes' parenthesis.

```
> myplot2 <- ggplot(data = iris, aes(x = Petal.Length, y
              = Petal.Width))+
              geom_point(aes(color = Species))+
              stat_smooth(method="lm",color = "black")
```

Finally, we could change axis and colors (you can call colors by their name in R, or use hexadecimal codes to call any color you want.

```
> myplot2 + xlab("Petal length (cm)")+
     ylab("Petal width (cm)")+
scale_color_manual(values=c("#D00000",
     "#FFBA08", "#3F88C5"))
```

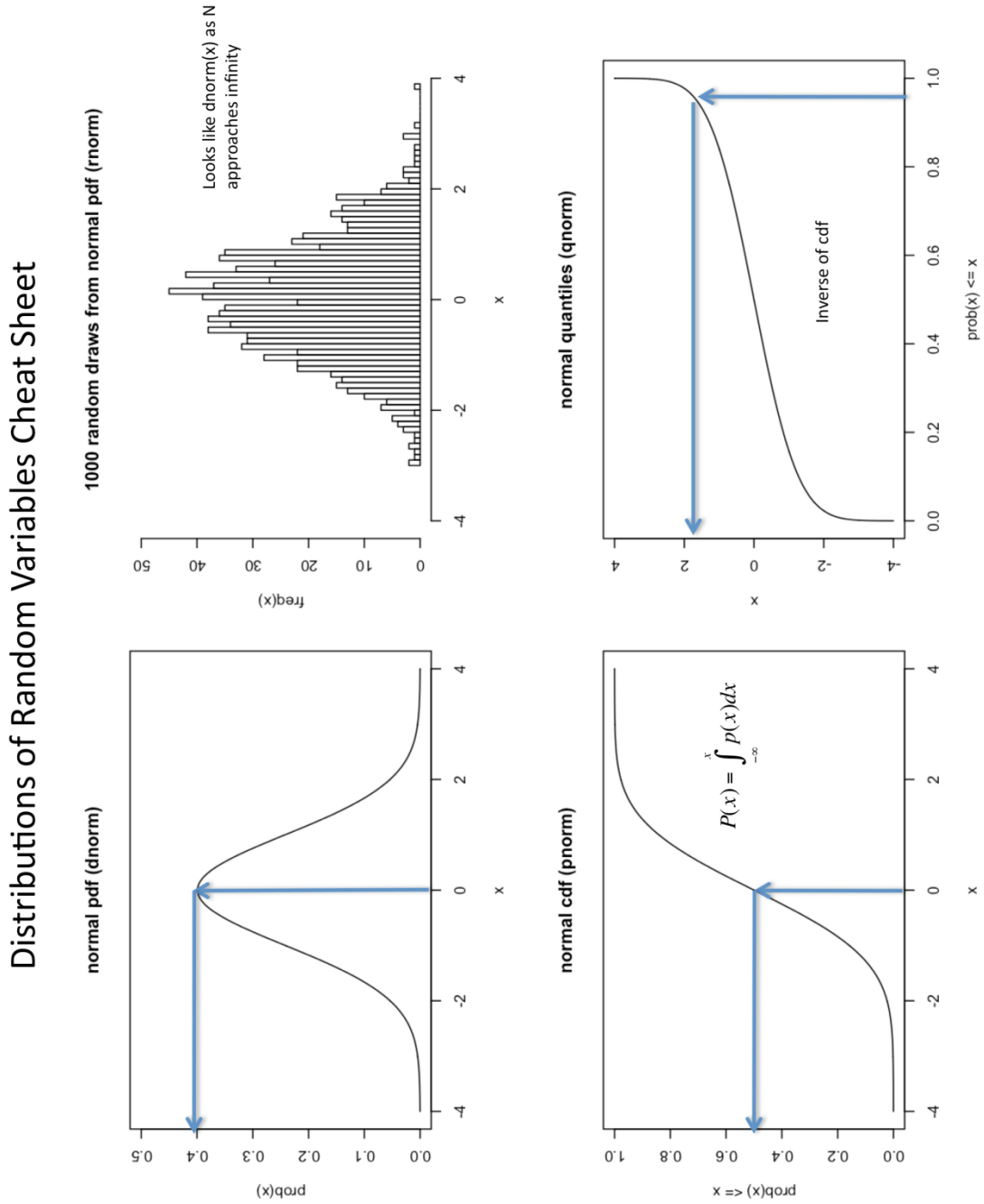
Cleaning up

"ls()" list all the functions in your working directory and "rm()" can be used to remove objects no longer needed.

```
ls() # There is foo.cov and several objects starting
     with "my"
ls(pattern = "my") # Yup, this lists only the "my"
     objects
rm(list = foo.cov)
rm(list = ls(pattern = "my"))
```

When you quit R (`q()`), R will ask if you want to save your work. If you say "n" all variables in your directory will be lost. If you say "y" they will be saved to your working directory as ".RData". Next time you start R from the directing which contains ".RData" all these files will be loaded into your workspace. This can potentially create a great deal of system overhead....so it is worthwhile to clean house occasionally. Or make a number of working directory for different projects!

R as a Set of Statistical Tables



R can supply critical values and other useful information for some of the distributions we discussed earlier. We will examine six of these distributions in more detail.

Statistical Distribution	R name	additional argument in R command
binomial	binom	size, prob
chi-squared	chisq	df
F	f	df1, df2
normal	norm	mean, sd
Student's t	t	df
uniform	unif	min, max

The trick is that each distribution must be preceded by a single letter a d, p, q, or r. This letter tells R that you want either the density, cumulative density (or probability), a quantile, or a random deviate.

Prefix	meaning	first parameter
d	density=probability at point *x*	*x*=the point you want the probability at
p	cumulative density=probability up to that point	*x*=point that you want the probability up to
q	quantile=the value of the statistic which has a probability *p* of an observation less than it	*p*=the desired probability
r	random deviates from that distribution	*n*=the number of deviates

These functions are best understood by going back to a previous example, *Ted the coin toss guy*.

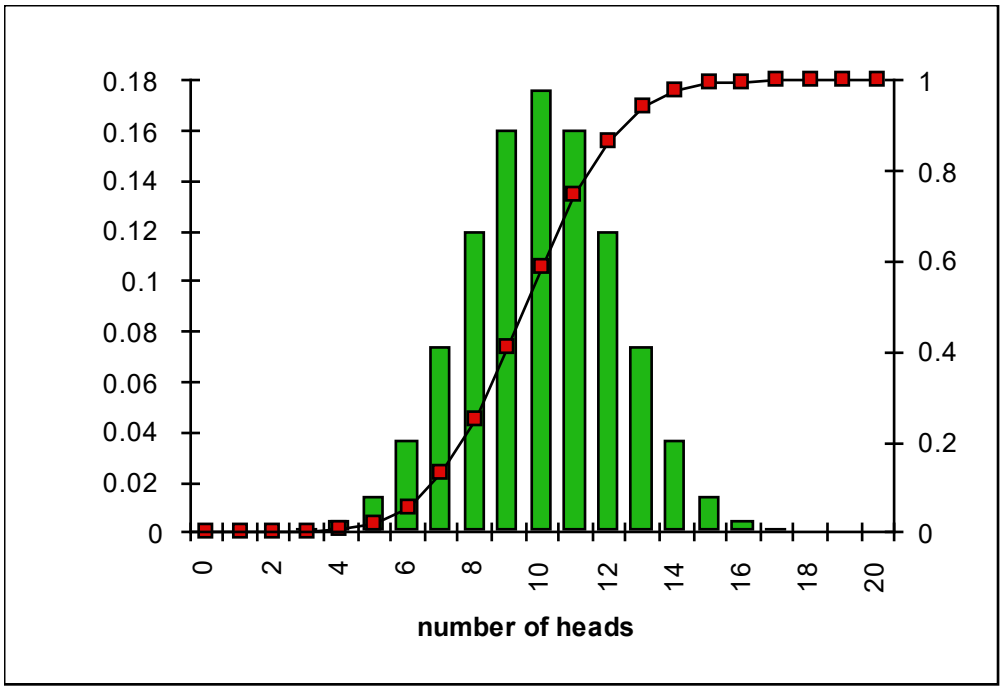


Figure 1. The binomial distribution.

What is plotted in Fig. 1 is the binomial distribution for $N = 20$ and $p = 0.5$. The bars are the density function or the probability at any given point. We could generate the bar at 9 (that is the probability of seeing exactly 9 Heads and 11 Tails) by typing:

```
dbinom(9, size = 20, prob = 0.5)
```

or the probability of 9 or fewer Heads by typing:

```
pbinom(9, size = 20, prob = 0.5)
```

Consider the following vector:

```
x <- 0:20
```

x is now a vector of the numbers 0 through 20

Can you think of a clever way to calculate the probability for every observed number of heads? How about the cumulative probability? How about making a figure? [Hint: Does the first parameter passed to these functions have to be a single number?]

```
> y <- dbinom(x, size = 20, prob = 0.5)
> plot(x, y , type = "l", col = "red")
```

We can also plot continuous pdf's.

```
x <- seq(from = -3, to = 3, by = 0.1)
#  try typing help(seq) to see what seq does
y <- dnorm(x = x, mean = 0, sd = 1)
plot(x = x, y = y, type = "l")
```

We can also use the R statistical table to calculate quantiles.

```
qbinom(p = 0.25, size = 20, prob = 0.5)
qbinom(p = c(0.025, 0.975), size = 20, prob = 0.5)
```

What do these two functions tell us?

Can you now calculate the 2.5% and 97.5% quantiles (q_1 and q_2) associated with a chi-square distribution with 99 degrees of freedom (corresponding to the horizontal red lines on page 50)?

How might we put a 95% confidence interval on the probability of a head given we observe 9 Heads in 20 trials? (see Box 1). Here is a way to "cheat" and let the computer do this for you

```
> binom.test(x = 9, n = 20, p = 0.5)
```

Box 1: Confidence intervals on an observed proportion

We want to establish the underlying binominal probabilities that would result in the observed numbers of successes "or worse" in the observed number of replicates. Therefore if we observe S successes in N trials we want:

Prob($X \leq S$; N, p_{UB}) = 2.5% for the upper bound
Prob($X \geq S$; N, p_{LB}) = 2.5% for the lower bound

Note that Prob($X \geq S$) = 1 – Prob($X \leq S$-1) for integers, Thus we can solve

Prob($X \leq S$-1; N, p_{LB}) = 97.5% for the lower bound. In R we would find a p_{LB}) such that

```
> pbinom(x = S-1, n = N, p = pLB) = 0.975
```

Some Useful Statistical Tests

R has a number of built in statistical tests. You were introduced to the binomial test in the above example. Two other useful built-in functions let you carry out a chi-square test and a t-test. Let's examine the Chi-square test first. First we have to enter the data in the earlier example (*Is sex-ratio influenced by rearing temperature in turtles?*) into R.

```
male.hatch <- c(50, 70)
female.hatch <- c(50, 20)
sex.ratio <- rbind(male.hatch, female.hatch)
sex.ratio
```

Remember that the `rbind` function "glues" rows together to create a matrix. Now let's tell R to do a Chi-square test:

```
chisq.test(sex.ratio)
```

(you may notice that the Chi-square statistic is slightly different from the one we calculated previously. This is due to the Yates' continuity correction. We will not worry about that here.)

Enter the data from "*Desiccation resistance in flies selected to resist starvation*" into R.

```
cntrl <- c(8.4,8.1,5.1,7.6,4.7,10.7,5.7,4.1,8.1,6.8)
strv <-
     c(12.4,15.8,11.7,8.6,12.6,11.1,10.5,7.3,7.2,10.8)
```

We can do a t-test on this data using the `t.test` function

```
t.test(x = cntrl, y = strv)
```

Sometimes we do a t-test on "paired" samples. In the case of the example the data above the data are naturally paired, since each population is derived from a different base population. If instead all the populations used in the experiment were derived from a single base population then the data is not necessarily paired. Paired data is commonly encountered in biological datasets, as often-times an experiment collects observations from a set of individuals before and after some treatment (*e.g.*, blood pressure before and after a patient receives Lipitor®). In the case of paired data the

null hypothesis becomes: the average difference between treatment over matched pairs is zero. And a paired t-test is carried out by typing

```
t.test(x = cntrl, y = strv, paired = TRUE)
```

Notice that performing the t-test and paired t-test yield different test statistics and p-values.

Other Resources

Download "An Introduction to R" under Documentation at:
"https://cran.us.r-project.org/".

Problem Set #1 (10 points):

1. i) Plot a binomial distribution for three different pairs of values of *p* and *N*; ii) plot a Chi-square distribution for three different degrees of freedom, and; iii) plot a *t*-distribution for three different degrees of freedom. For EACH set of distributions, note how the shape of the distribution is affected by changing parameters. You must choose an appropriate range of values for the x-axis so that the at least 99% of the density of the distribution is visible. A common mistake is to plot a range that doesn't capture the shape of the distribution. Doing this **will** lost credit.

2. i) Plot a cumulative probability distribution similar to the one in the coin toss experiment, but for the probability of the number of sixes rolled in 10 rolls of a die (a die has 6 sides!).
ii) If we observe four sixes in 10 rolls, is this die likely to be loaded? If we observe a six four times out ten what is a 95% confidence interval on the underlying probability of rolling a six?
iii) Say instead we had rolled 40 sixes out of 100. Is the die likely to be loaded? What is a 95% confidence interval on the probability of rolling a six?
iv) Say instead we had rolled 400 sixes out of 1000. Is the die likely to be loaded? What is a 95% confidence interval on the probability of rolling a six?
Hint: use `binom.test()` for parts ii-iv.

3. If two heterozygous parents (i.e., both *Aa*) mate, they produce offspring in the expected Mendelian proportions (i.e., 1/4 *AA*, 1/2 *Aa*, 1/4 *aa*). We observe a big *Drosophila* family of 40 offspring of which 7 are *aa*.
i) Construct a 95% confidence interval on the observed proportion of *aa* offspring (7/40). What does this confidence interval tell us about our null

hypothesis? Hint: our null hypothesis is that Mendelian laws have not been violated.
ii) Are 7 or fewer offspring *statistically* consistent with Mendelian expectations? Hint: figure out the probability of observing 7 or fewer offspring in such a family, if Mendelian laws have not been violated.
iii) How few offspring of type *aa* would we have to observe in order to be suspicious that Mendelian proportions are being violated? Is your answer a test of a one-tailed hypothesis or a two-tailed hypothesis?

Problem Set #2 (10 points):

1. Make up a story with new data like "*Is sex-ratio influenced by rearing temperature in turtles?*" for a different problem (unrelated to sex or temperature) that can be tested using a Chi-square analysis. Analyze it in R. State your null and alternative hypothesis and interpret the *p*-value.

2. Make up a non-biological story and data to go with it similar in flavor to "*Desiccation resistance in flies selected to resist starvation*" that requires a *t*-test. Analyze it in R. State your null and alternative hypothesis and interpret the p-value.

3. Input the dataset "`ASC.data.txt`" into R (you will have to load this data into your computer using `read.table`). The first four columns are bristle number measures taken from different parts of the fly calculated in males and females and the remaining columns are molecular polymophisms in a gene region call *achaete-scute* known to effect bristle number development. Rows are different lines we maintain in the lab. It is hypothesized that a subset of the DNA polymorphisms in this region have effects on bristle number.
i) Calculate the mean, variance, standard deviation, and min and max for each column. Hint: While you must use `apply()` for `var()` and `sd()`, the function `summary()` works on columns of matrices by default, and produces mean, min and max.
ii) Calculate the variance/covariance, and correlation matrices associated with this data. Hint: `var()` and `cor()` will produce covariance and correlation between all possible combinations of columns. NB: The covariance between a column and itself is its variance. Confirm this by comparing your answers here to your answers in i) above.
iii) To visualize this data, make at least one scatter plot (using the "plot" command), and at least one histogram. Pick any subsets of the variables you wish to do this, and describe in words the relationships between the variables.

4. In the "*Desiccation resistance in flies selected to resist starvation*" example, is there any evidence that the **variances** in survival times in the control populations versus the starvation-resistant populations differ? Hint: construct a 95% confidence interval on the observed variances (see page 49). You will have to calculate quantiles of the chi-square distribution to estimate q_1 and q_2.

Monte Carlo Simulation

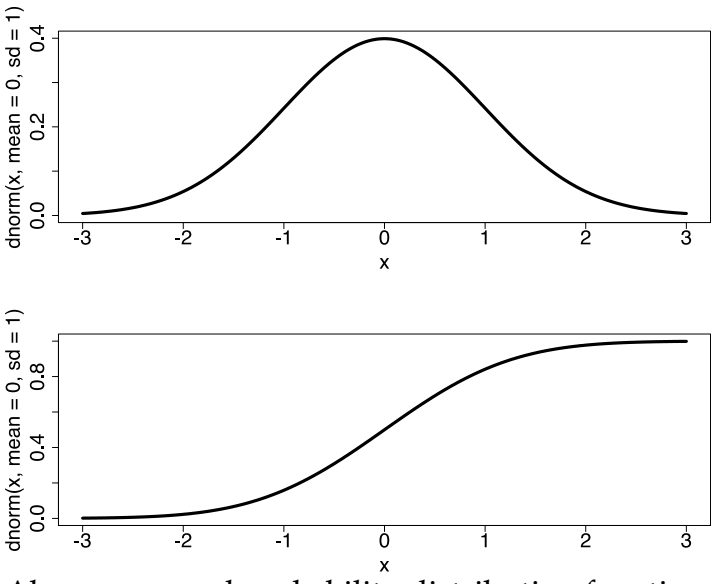
The last chapter focused on the concept of hypothesis testing. We introduced you to a distilled view of how statistics works. Namely:

1. You set up a null hypothesis and an alternative hypothesis.
2. You record your data and then calculate some "statistic" or summary of the data.
3. You count on a statistician having determined the distribution of your statistic given that the null hypothesis is true.
4. You look and see where the observed value of your statistic falls in that theoretical distribution. You determine if your observed values are likely to have occurred by chance alone?

We also introduced you to some statistical distributions used for testing certain types of data. Specifically, you were introduced to the binomial, Chi-square, and t-distributions. In our introduction to R we described a number of other distributions but didn't discuss the circumstances under which these distributions are used. We attempted to plot the distributions whenever possible so that the concept of a test-statistic being "extreme" was visually illustrated.

This is all very convenient. If we lived in a truly tidy universe hypothesis testing would be simple. *But what do we do if we can not write down the distribution of some test statistic under the null hypothesis?*

Lets consider the example below:



Above a normal probability distribution function (i.e., dnorm(x,0,1)) and a normal cumulative probability distribution function (i.e., pnorm(x,0,1)). Note: qnorm(prob,0,1) would give the value of the "x-axis" in the lower panel associated with any given probability ("prob") fed to the qnorm function.

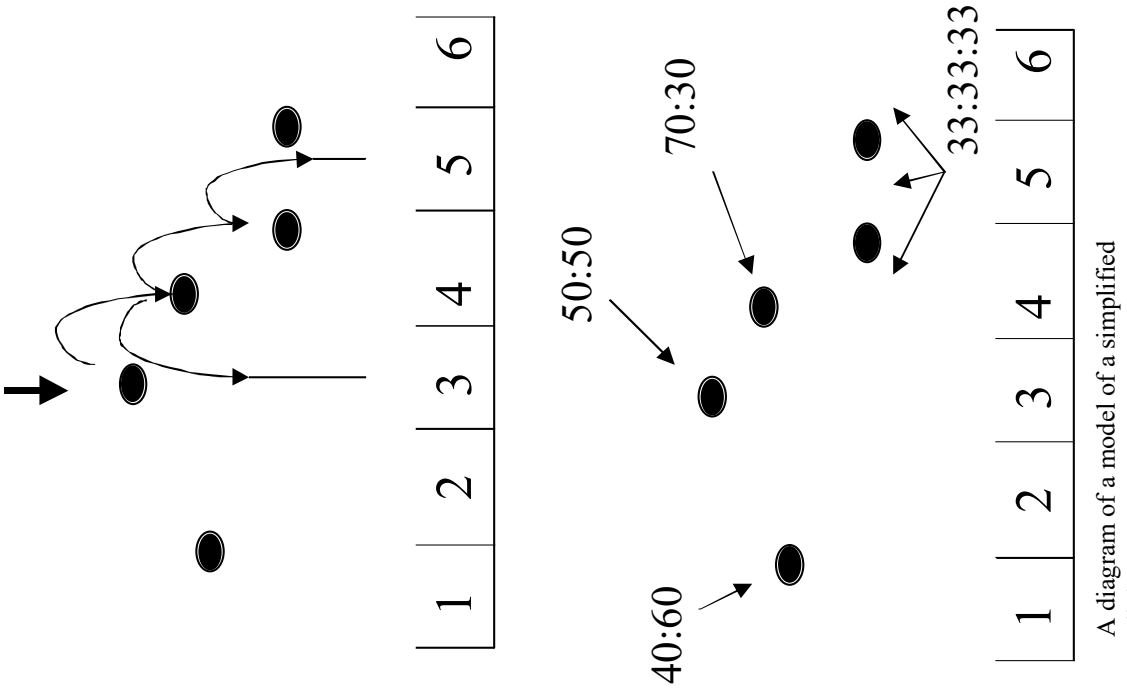
Certain statistics of interest under specific null hypotheses come from KNOWN distributions such as those above. Examples include the *t-statistic* which under the null is distributed as a *t-distribution* and the *Chi-square statistic* which under the null is distributed as a *Chi-square distribution* (we will see this in Lecture 10). Note the distinction between a statistic and a distribution. A statistic is some sort of summary number we can calculate from the data, whereas a distribution is a theoretical construct represented by a function that integrates to one.

Now lets imagine a process which is not nearly so neatly specified by a theoretical distribution. A great example of this is a Japanese game of chance called "pachinko" in which steel balls are fed into a machine. The steel balls collide with various pins while falling and depending on the pins they collide with (and which way they bounce after hitting pins) end up in a different collection bin at the bottom of the game. It's extremely complicated and would be very hard to model statistically. You can find a lot of examples by typing in "pachinko" into YouTube:
https://www.youtube.com/results?search_query=pachinko

For the sake of discussion, we'll briefly consider a similar, but far simpler, game called Plinko, played on a gameshow called the The Price Is Right. In this variant, the player places a disk, rather than steel balls, at the top of the

board, that then bounces off of pins as it falls down to eventually land in one of nine collection bins (see first picture on next page).

Lets say we number the various bins 1 through 9. An interesting statistical question is: Given that we know the rules which determine how balls fall within the Pachinko can we derive the distribution of the frequency that any given ball ends up in each of the 9 bins? Of course, the ability to determine these types of distributions have important applications whenever we can describe the processes (biological or not) that gives rise to the statistic we keep track of. Oftentimes we can determine these distributions by a process calls "Monte Carlo Simulation" even when we can not write down an analytical solution for the probability distribution function. Thus such Monte Carlo simulations have wide applicability in biological situations. Let's simplify our Plinko board even more. Consider a board with only 6 bins and considerably pins. Furthermore, let's force the player to drop the chip directly above the top pin (see second image on the next page).



A diagram of a model of a simplified Plinko game

The Plinko game

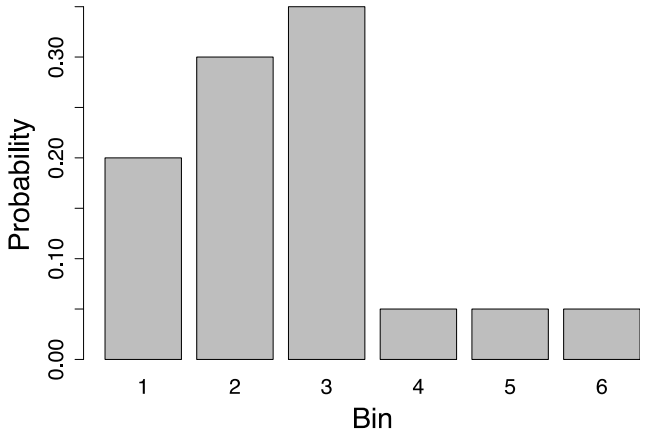
Consider the above process on the next page. What we have here is a Pachinko machine in which we know the probability a ball goes right or left at any given pin.

In this particular machine the probability of ending up in each of the bins is "known"

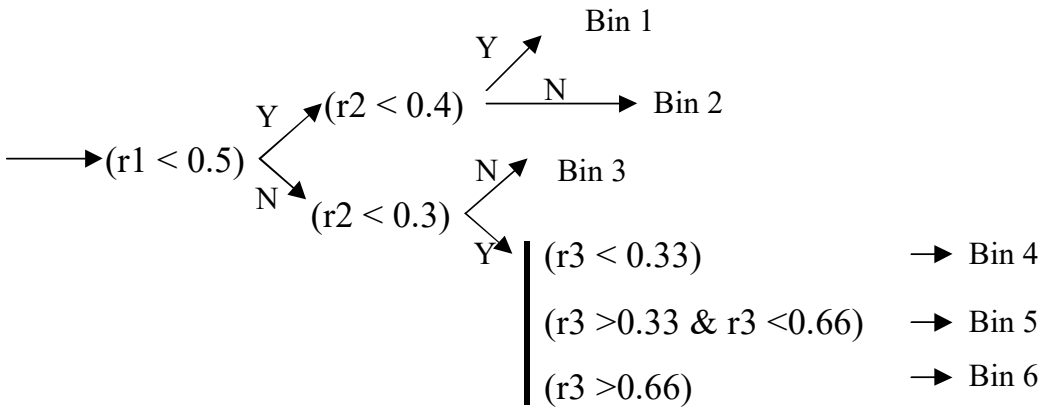
Bin	Probability
1	0.5 * 0.4
2	0.5 * 0.6
3	0.5 * 0.7
4	0.5 * 0.3 * 0.333
5	0.5 * 0.3 * 0.333
6	0.5 * 0.3 * 0.333

With a resultant probability distribution function:

Now say we wanted to run a computer simulation of the process that gives rise to the above distribution, as opposed to directly calculating the resulting pdf.



Consider the following "program" where "r*x*" implies getting a single uniform random variable between zero and one.



The above program would start on the left and through subsequent calls to the random number generator "wind" its way down the Pachinko machine. Each run of the machine would start at the left hand side and end when it reached one of the 6 Bins. Each run can be thought of as a single realization of the process that gives rise to the pdf above – that is it can be thought of as one trial of the Plinko game. By repeating this process a large number of times and keeping track of the result each time we can get the distribution of the resultant statistic (visually just like the barplot above).

In terms of a computer program we would start all the Bins at zero, run the Pachinko trial a large number of times (say one million) and for each trial "increment" the value in a Bin by one if the ball ends up in that bin. In the end we would just count up the number of balls in each bin (i.e., the value

of that bin) and divide by the number of trials to get the probability of each outcome.

Luckily in the modern world we can often simulate on a computer the distribution of some test statistic under the null hypothesis, despite the fact that we cannot write down an analytical solution for that test statistic. We do this using something called a Monte Carlo simulation. This works by recreating on the computer the process that gives rise to the data we are measuring a large number of times. We can then use these results to empirically estimate the distribution of some test statistic we are interested in. We can do this for something as simple as the binomial distribution, as complex as the variance among a set of populations of *Drosophila* that are experiencing random genetic drift, or things much more complex than that.

A re-derivation of the Binomial Distribution (p = 0.5, N = 20)

We wish to carry out a Monte Carlo simulation to "derive" a binomial distribution via computer simulation. The approach we will use to do this is outlined below:

I. Write a function that simulates a single binomial trial comprised of 20 events, each with a probality of success of 50%
II. Run this function 10,000 times, recording the number of success in each trial
III. Compare this simulation to the theoretical distribution

Below is a set of R commands which implements our Monte Carlo simulation of the binomial distribution for prob = 0.5 and size = 20. It looks difficult so we will walk through it line by line following the code.

```
# 1: a function to run a single binomial trial
binom_trial <- function(size, prob) {
  sum(runif(size) < prob)
}

# 2: a function to run many binomial trials
my_rbinom <- function(n, size, prob) {
  replicate(n = n, expr = binom_trial(size = size, prob = prob))
}

# 3: define variables for the simulation
p <- 0.5; trial_size = 20; num_trials <- 1e4

# 4: simulate!
result.mc <- my_rbinom(n = num_trials, size = trial_size, prob = p)
```

```
# 5: table() counts the frequency of each number
# plot() is smart enough to do the "right thing" with a table
# type = 'o' overlaps points on top of a line
plot(table(result.mc)/num_trials, type = 'o', pch = 19, col =
      'gray')

# 6: points adds points to an existing plot
# this plots the theoretical distribution against our Monte Carlo
      simulation
points(x = 0:trial_size, y = dbinom(x = 0:trial_size, size =
      trial_size, prob = p), col = 'red', type = 'o')
```

The first statement (#1) defines a function named `binom_trial()` which simulates a single binomial trial. A binomial trial is composed of observing an event `size` times. Each independent event results in success with a probability of `prob` and failure with a probability of `1-prob`. The function determines success by generating several random numbers (specified by `size`) uniformly distributed between zero and one using the built in `runif()` function. By definition, a random uniform number has a probability of `prob` to be less than `prob`. Consequently, we count each random number less than `prob` as a success. By summing up how many numbers are less than `prob`, we count the number of successes.

Statement #2 is far simpler. It defines a function called `my_rbinom()` that calls our `binom_trial()` function `n` times by using the built in function `replicate()`. Functions like `my_rbinom()` which are mainly designed to call other functions are often referred to as "wrapper functions". This wrapper function effectively runs `n` binomial trials.

In the next line (#3), we define the number of events in a trial (`trial_size`), the probability of success of an event (`p`), and the number of trials (`num_trials`). Then (#4) we call our simulation function and supply the variables from #3 and save the number of successes for each trial to the variable `result.mc`.

In the next line (#5), we tabulate the number of trials resulting in 0 successes, 1 success, 2 successes, etc. using the `table()` function and dividte it by `num_trials` to get a probability. These numbers are passed directly to the `plot()` function, which is smart enough to plot this number appropriately. We specify that the plot is in gray. Finally (#6), we use the `points()` function to plot the theoretical expectation, which we get from the `dbinom()` function.

The plot function has a number of useful options:

- `type = "p" for points, "l" for lines, b for "both" etc`
- `lty = 1,2,` etc. specifies a line type (solid, dashed, etc.)
- `pch = 0, ...,25` specifies the symbol plotted
- `xlab =` "the label you want on the x-axis"
- `ylab =` "the label you want on the y-axis"
- `xlim = c(min, max)` specifies the range of values plotted on the x-axis
- `ylim = c(min, max)` specifies the range of values plotted on the y-axis

Closely related to the plot function are the `lines(x,y)` and `points(x,y)` functions. These function add lines or points respectively to a pre-existing plot. They use the same options as the plot command. We used the `points()` function in the above example to plot the predicted probability densities for the binomial on the same figure as our Monte Carlo simulation.

So why is this so cool?

The important point to understand with this example is that we have used the computer to produce a binomial distribution without working out the probability distribution theoretically or even relying on R to do it for us. In order to do this we simulated a large (10,000) number of samples from a process identical to the one the binomial distribution assumes, but without recourse to the actual binomial distribution. The powerful feature of this approach is that if we then did a coin toss experiment and observed some outcome (e.g., 3 Heads), we could assign a probability to the event happening by chance alone – without knowing anything about the binomial distribution. Obviously, in the case of the coin toss experiment this was a lot of work compared to just using the known distribution. But the Monte Carlo approach can easily be extended to more complex models that are difficult, or perhaps even impossible, to model analytically.

A re-derivation of the *t*-distribution (10 samples versus 10 samples)

We will do another example of a Monte Carlo simulation, this time for the example "*Desiccation resistance in flies selected to resist starvation*". Like the last example, we will first look at the algorithm we will try to write R code for.

I. Write a function that simulates a single t-statistic from a two samples, both with 10 observations. Each sample is drawn from a random normal variable with mean zero and standard deviation one. The formulate for the t-stastic we'll be using is:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{n_2} + \frac{s_2^2}{n_1}}};$$

II. Write a wrapper for the function above and run it 10,000 times;
III. Compare a histogram of our simulated t-statistics to the theoretical expectation.

Below is the R code. We will work through this example line by line.

```
# 1: Function to simulate a t-statistic from two samples
single_rt <- function(size1, size2) {
  x1 <- rnorm(size1)
  x2 <- rnorm(size2)
  ( mean(x1)-mean(x2) ) / sqrt( (var(x1)/size2)+(var(x2)/size1) )
}

# 2: A wrapper to run the single_rt() function many times
my_rt <- function(n, size1, size2) {
  replicate(n = n, expr = single_rt(size1 = size1, size2 = size2))
}

# 3a: Define conditions for a simulation.
num_reps <- 1e4; N1 <- 10; N2 <- 10

# 3b: Run the simulation and obtain simulated t-statistics
t.stats <- my_rt(n = num_reps, size1 = N1, size2 = N2)
# 3c: Get the values of the theoretical t distribution using the
# degrees of freedom related to our simulation.
x_theory <- seq(from = -5, to = 5, by = 0.01) # Where to evaluate
      theory
y_theory <- dt(x = x_theory, df = N1+N2-2)

# 4: Plot the simulation and theory results on the same graph.
hist(t.stats, prob = TRUE, breaks = 25, xlim = c(-5, 5), ylim =
      c(0, 0.41))
lines(x = x_theory, y = y_theory, col = 'red')
```

First (#1) we write a function that simulates two random samples. The only difference between them is one has sample size `size1` and the other has sample size `size2`. Next (#2), we wrap the function from #1 in a function that lets us run it as many times as we want using the built in `replicate()` function.

We now (#3a) set variables to hold the number of control and evolved populations (`N1` and `N2`) and the number of simulations we want to run (`num_reps`). We use the values to both run the simulation (#3b) and to obtain the predictions from theory (#3c).

Finally (#4), we plot a histogram of our simulation and overlay the theoretically predicted probability density. Remember, a histogram

constructed with `prob = TRUE` is an approximation of the probability density of the distribution underlying the data. If the Monte Carlo simulation is working, the observed cumulative distribution should be similar to the theory!

A question that may arise is how did we "know" to seed the Monte Carlo matrix with numbers from a random normal distribution? It turns out that as *n* gets big, the *t*-statistic is distributed according to a *t*-distribution as long as the individual observations are drawn INDEPENDENTLY from the SAME distribution. This claim is explained by something called the Central Limit Theorem. So if I were to define something called:

```
my.distribution <- c(1, 17, 42, 5, 6, 40, 3, 10, 45,
     40, 1, 4, 7, 6)
```

That is, the above numbers occur with equal probabilities then I could have replaced the following two lines

```
x1 <- rnorm(size1)
x2 <- rnorm(size2)
```

in the above example, with

```
x1 <- sample(x = my.distribution, size = size1,
     replace = TRUE)
x2 <- sample(x = my.distribution, size = size2,
     replace = TRUE)
```

as the `sample(x, n, replace = TRUE)` will draw a sample of size *n* with replacement from the vector *x*. Where "with replacement" means after drawing an observation you put it back in the "pile" to potentially be drawn again. Think of it as drawing a card from a deck, replacing the card, shuffling the deck, and drawing again – you could draw the same card twice in a row We will not make use of this observation, but it has very important ramifications for more advanced courses on the design and analysis of experimental design.

What is the distribution in the variance of allele frequencies in a hypothetical drift experiment?

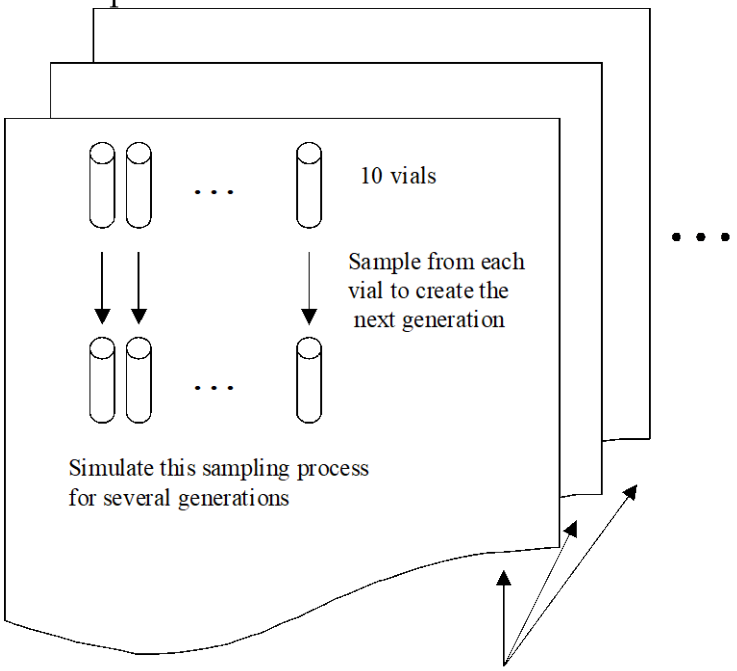
In the last two examples the distribution of the corresponding statistic was known. The next two examples deal with distributions that are either not

known, or more difficult to figure out. Lets consider a random genetic drift experiment, like the one you did in the lab portion of the course.

We will simulate 10,000 replicates each consisting of 10 populations. Below is the algorithm we will employ:

I. First we will "seed" a simulation. We will simulate 10 vials, each starting with 10 copies of the "w" allele and 10 copies of the "wild type".
II. Now to generate the next generation for each vial draw a random binomial deviate of size 20 alleles with the probabily of drawing a "w" allele being equal to its observed frequency in the vial from the preceding generation.
III. Repeat this process for 5 generations. We are now done the Monte Carlo simulation for a single replicate.
IV. Repeat the above process 10,000 times to get the fully replicated simulation.

Or as a picture:



Simulate 10,000 Monte Carlo replicates of this experiment

We start out with 10 populations with initial allele frequency = 0.5, population size = 20 alleles (*i.e*, 10 diploid individuals) and then let them drift for 5 generations. Each population is then completely characterized by 20 copies of the white locus (some fraction WT and some fraction w^-). We repeat this 10,000 times to get our simulation. We will step through this

very careful in the code below. There's no way to sugar coat this. There is a lot of code below and it uses most of the R programming concepts you've leared up until now.

The next page and a half might seem tough at first. But a few things make it not so bad. First, most of the code is actually just comments. Anything following the hashtags (#) is a comment and is ignored by R. But those comments are useful to us, the people reading the code. Whenever you write code, please try to do the same thing. It really makes life easier for others. The second reason the code isn't as bad as it might seem at first is that we've already sketched out our basic approach both in words and in a diagram above. Now all we must do is translate that into code.

If you haven't already, **now is the time to review the learning R videos and to take the interactive R tutorial**. You can review how to run the interactive swirl() tutorial in the section above titled "Getting up to Speed with R Using swirl()". The videos can be found here: http://bit.ly/E115L_LearningR

```
# This function simulates a binomial experiment with sample size of
#   size and a probability of success x. The number of successes is
#   divided by the sample size to get another frequency.
drift_step <- function(x, size) {
  rbinom(n = 1, size = size, prob = x)/size
}

# The goal of this function is to run the drift_step() many times.
# It will run for several different populations each generation (equal
# to the number of frequencies specified by pop_frequencies).
# It will perform num_generations generations of simulation. The
# starting frequencies are specified by the values in pop_frequencies.
drift_sim <- function(pop_size, num_generations, pop_frequencies) {
  # Creates a matrix with num_generations+1 rows and pop_frequencies
  # columns.
  # Each column is a single population. Each row is a single generation.
  # The first row will hold the starting conditions, or generation 0.
  # Each value in the matrix is an allele frequency for a population at a
  # particular generation.
  ans <- matrix(nrow = num_generations+1, ncol = length(pop_frequencies))

  # This assigns the starting allele frequencies to the first row.
  ans[1,] <- pop_frequencies # Generation 0 is row 1.

  # We're going to step through each generation. We keep track of which
  #   generation we're on with the value i.
  for (i in 1:num_generations) {
    # Fill the next generation (i+1) with results by simulating
    # based on frequencies from the current generation (i).
    # The drift_step() function simulates one step of genetic drift
    # using a random binomial deviate. sapply() does its magic by
    # executing the drift_step() function for every value in ans[i,].
    # Finally, we pass the parameter size to drift_step() by giving it
    # sapply().
    ans[i+1,] <-
      sapply(
        X = ans[i,],
        FUN = drift_step,
        size = pop_size
      )
  }
  return(ans)
}

# This function simply applies the mean() and var() function to its argument
# and returns both results with meaningful names. We'll use it later.
summarize_function <- function(x) {
  # Calculate mean(x) and var(x)
  ans <- c(mean(x), var(x))
  # Give names to the results.
  names(ans) <- c('mean', 'var')
  return(ans)
}

# In the schematic figure above, this function first simulates a single
# "sheet". Then it summarizes the generations (ie rows in that "sheet")
# returning a mean and a variance for each generation simulated.
```

```
drift_summary <- function(pop_size, num_generations, pop_frequencies) {
  # Simulate a "sheet". In each "sheet", a column is a population/vial over
  # time and a row is a generation. This is carried out by the
  # drift_sim() function above.
  sim <- drift_sim(pop_size = pop_size, num_generations = num_generations,
      pop_frequencies = pop_frequencies)
  # Summarize the rows (ie generations) with summarize_function(), which
  # we described above. The answer that's returned assigns a row for means
  # and a row for variances, making the columns generations. This
  # effectively rotates the table, but oh well. We'll deal with that below.
  ans <- apply(X = sim, MARGIN = 1, FUN = summarize_function)
  # Label the columns as generations. Remember that apply rotated the
  # generations to be columns.
  colnames(ans) <- paste('gen', 0:num_generations, sep = '_')
  # This will return a matrix with the columns being mean and variance and
  # the rows being generations. This is the reverse of above. The magic
  # comes from the t() function, which swaps rows and columns ("t" is
  # short for the matrix operation "transpose").
  return(t(ans))
}

# Simulate many runs of drift_summary, leading to a different matrix for
# each replicate.
drift.mc <-
  replicate(
    n = 1e4,
    expr =
      drift_summary(
        pop_size = 20, num_generations = 5, pop_frequencies = rep(0.5, 10)
      )
  )

# Set up a plotting area with three rows and two columns. "mfcol" tells
#   the plot() function to fill the plot column-wise.
par(mfcol = c(3,2))
hist(drift.mc['gen_1', 'var',], xlim = c(0, 0.2))
hist(drift.mc['gen_2', 'var',], xlim = c(0, 0.2))
hist(drift.mc['gen_3', 'var',], xlim = c(0, 0.2))
hist(drift.mc['gen_4', 'var',], xlim = c(0, 0.2))
hist(drift.mc['gen_5', 'var',], xlim = c(0, 0.2))
```

In a hypothetical experiment, let's say an experimenter observed a variance in allele frequencies of 0.05 in the first generation after establishment. Based on our Monte Carlo simulation, what is the probability of seeing a variance larger than 0.05 by chance alone? In theory we can estimate this from the histogram we just made. Or we could take a slightly more sophisticated approach:

```
sum(drift.mc['gen_5', 'var',] > 0.05)/1e4
```

In short, we ask for all the independent replications of the entire experiment (10,000 or 1e4) for which the variance was greater than 0.05 and take its

length. This is the total number of Monte Carlo replications for which we observed a larger variance larger than 0.05. Now we just divide by 10,000 to change that to a frequency. You can do something analogous for any time point or statistic you are interested in.

Iterative Solutions to an Equation and Iterative Processes

R contains control structures that allow the user to "loop" through a set of calculations. Below is such an example:

```
x <- 0
for (i in 1:10) {
  x <- x+i
}
print(x)
```

The statement `for (i in 1:10)` sets the variable `i` equal to 1 and then carries out the statement in brackets. The variable `i` is then incremented to 2 and the statement in curly brackets is repeated. This goes on until `i=10` and the statement is carried out for the last time. Verify for yourself that this will result in the variable `x` having a value of 55.

Selection and drift

{Review making your own function on page 64}

In R it is relatively straightforward to write your own functions. I have written two such functions for you to enjoy.

```
drift.select(x,N,w11,w12,w22)
evolve.time(x,N,w11,w12,w22,time)
```

where `x` is a column vector of starting allele frequencies p(A) (for example the starting frequency of the "w" allele), `N` is the population size, `w11`, `w12`, and `w22` are the fitnesses of AA, Aa, and aa respectively, and time is the number of generations to run the drift/selection experiment. These two functions allow you to easily do a Monte Carlo simulation of a small population that incorporates BOTH random genetic drift and natural selection. The easist way to get functions into R is simply to use copy and paste. Then you can view the functions by typing their names with no parentheses

```
# This function takes single starting frequency, and applies
# genetic drift and natural selection to it.
drift.select <- function(x, N, w11, w12, w22) {
  # x is column of starting frequencies
  # N is diploid population size
  # w11, w12, and w22 are fitnesses
  if(x != 0 || x != 1) {
    # drift term
    x <- rbinom(n = 1, size = N, prob = x)/N
    #selection term
    x <- x +
      (x*(1-x) * ( x*(w11-w12)+(1-x)*(w12-w22) )) /
      (w11*x^2 + w12*2*x*(1-x) + w22*(1-x)^2)
  }
  return(x)
}

evolve.time <- function(x, N, w11, w12, w22, time) {
  # x is a vector of starting frequencies
  # N is diploid population size
  # w11, w12, and w22 are fitnesses
  # time is number of generations
  ans <- matrix(nrow = length(x), ncol = time+1)
  ans[,1] <- x
  for(i in 1:time){
    ans[,i+1] <- sapply(X = ans[,i], FUN = function(x)
drift.select(x, N, w11, w12, w22) )
  }
  return(ans)
}
```

You run `evolve.time()` for a given set of starting allele frequencies, population size, fitnesses, and number of generations (time). evolve.time will then just run `drift.select()` for "time" generations. Each generation it generates a new column of allele frequencies and appends it to the starting column, subsequent generations using this new allele frequency to start the next generation.

`drift.select()` calculates a new allele frequency based in part on the fact that a binomial distribution predicts the change in allele frequency over one generation due to genetic drift. The first term is the change due to drift. The second term predicts the change in allele frequency due to the action of natural selection. The equation used in this term can be found in any population genetics or evolution text book.

A really useful property of functions is that they can be simply used without caring a great deal about how they actually work. Here is an example of `evolve.time` in action:

```
# start 75 replicates each at 50% as a column vector
p.start <- rep(0.5, 75)

# run 50 generations at a diploid population size of 25
Aa.most.fit <- evolve.time(x = p.start, N = 25, w11 = 0.9,
w12 = 1, w22 = 0.6, time = 50)
all.fit.equal <- evolve.time(x = p.start, N = 25, w11 = 1,
w12 = 1, w22 = 1, time = 50)

par(mfrow=c(2,1))
plot(0:50, apply(Aa.most.fit, 2 ,mean), xlab = "generation",
     ylab = "mean", ylim = c(0,1), type = "l")
lines(0:50, apply(all.fit.equal, 2 ,mean), col = "red")

Aa.most.fit.var <- apply(Aa.most.fit, 2, var)
all.fit.equal.var <- apply(all.fit.equal, 2, var)
varrange <- range(c(Aa.most.fit.var, all.fit.equal.var))
plot(0:50,
  Aa.most.fit.var, xlab = "generation",
  ylab = "var", ylim = varrange, type = "l")
lines(0:50, all.fit.equal.var, col = "red")
```

First we make a column vector consisting of 75 "0.5's". They represent 75 populations each with a starting frequency of 50%.

Then we run evolve.time for 50 generations. In the first case the heterozyogote is the most fit, in the second case all three genotypes have the same fitness.

Remember apply(*matrix*, 2, mean) will take the mean of each column of *matrix*. The output of evolve.time is a matrix in which each column are 75 replicate evolved populations, and each additional column is a generation of evolution. Therefore the "plot" commands plot either the average allele frequency against time, or the variance in allele frequency against time. I have used the "ylim" parameter and the lines command to put the plots on the same figure.

When the heterozygote is the most fit, what do we expect the allele frequencies to do over time? Why do the means differ when the heterozyogte is most fit compared to when all genotypes have equal fitness? Why do the variances differ?

Problem Set #3 (10 points):

Reminder for all plots: You won't get full credit unless your plots show appropriate x and y axis ranges. For plotting distributions, this means we need to see all or most of distribution. Distributions like the normal have non-zero density from -∞ to ∞, so obviously we can't ask you to plot everything. Also, for binomial distributions with a large size parameter, most of the distribution fits in a narrow window. Even though you can plot the whole distribution, you can plot it for the x range containing most of the density. Of course, never chop off the y range of a distribution!

1. Plot a binomial distribution for the three different pairs of values of p and N you used in problem set #1 *without using* to the theoretical binomial distribution provided by R.

2. Plot a figure similar to the one in the coin toss experiment, but for the distribution of rolling sixes in 10 rolls of a die (a die has 6 sides!) *without recourse* to the binomial distribution. If I rolled 4 sixes is this die likely to be loaded? What is the probability of rolling 4 sixes in 10 rolls, according to your Monte Carlo simulation? Hint: use the "sort" and/or "sum" commands.

3. In the last assignment you made up a story like "*Desiccation resistance in flies selected to resist starvation*" that required a t-test and analyzed it in R. Re-analyze it using a Monte Carlo simulation. Calculate your t-statistic, compare it to your simulated distribution, and explain whether or not you reject your null hypothesis.

4. Use the Monte Carlo simulation for the "*Desiccation resistance in flies selected to resist starvation*" example (p.81) to answer the following questions.

i) What is the distribution under a Monte Carlo simulation of a t-statistic when the standard deviations of the two samples differ by two-fold? Compare this distribution with the theoretical t-distribution. What conclusions can you draw from this comparison?

ii) What is the distribution of the t-statistic when the null hypothesis states that the means and the standard deviations of the two groups vary by two fold (sample sizes still 10)? Compare this distribution with the theoretical t-distribution. What conclusions can you draw from this comparison?

iii) What is the distribution of the t-statistic if the means stay the same, but standard deviations vary by two-fold and the sample size of the first sample is 4 and the second is 12? Compare this distribution with the theoretical t-distribution. What conclusions can you draw from this comparison?

Problem Set #4 (10 points):

1. Is the variation in allele frequency estimates from your drift experiment different from that expected due to chance alone:

i) in the Monte Carlo simulation example?

ii) in a Monte Carlo simulation example with an effective population size equal to that of your experiment?

iii) Is there any evidence for the mean allele frequency differing from that expected by chance alone?

iv) What happens to the variance in allele frequency over time?

2. Run `evolve.time` for the parameters given in the example. Why do the means and variances of allele frequencies differ for the two examples? Run `evolve.time` (500 replicates) of each "experiement" (exp) for the following parameter values:

parameter	**exp1**	**exp2**	**exp3**	**exp4**
N	50	200	50	200
w11	0.8	0.8	1	1
w12	1	1	1	1
w22	0.6	0.6	0.2	0.2
time	50	50	50	50

Explain the differences and similarities in summary statistics and figures generated from these four Monte Carlo simulations using evolutionary arguments. Hint: in your plots, compare experiment 1 with experiment 3, and experiment 2 with experiment 4.

Phylogenetic Inference

We can obtain DNA sequence from a number of individuals, each being a representative of a different species. We often assume that the majority of the DNA sites are evolving in a neutral fashion. Under this assumption the rate at which differences accumulate between species is proportional to the mutation rate, μ. If we further assume that the mutation rate is approximately constant throughout the evolutionary history of the molecule being examined, then the number of differences separating any two sequences is proportional to the amount of time the species have diverged (in generations).

Under the above assumptions we can use the "distance" between sequences to construct a phylogenetic tree representing how and when species have diverged from one another. These phylogenetic trees are used in a large number of contexts in evolutionary biology. They are often constructed using quite sophisticated algorithms. This introduction to phylogenetic trees will use a very simple tree building algorithm. The intent of this approach is to introduce you to the idea of phylogenetic trees and some of their uses.

Constructing a distance matrix

Lets read in some example data (remember to download any files needed to your desktop and change to the proper working directory)

```
examp <- read.table("example.txt", row.names=1)
examp
```

cat	a	c	g	g	t	c	a	t	t
puma	a	c	c	g	t	c	a	t	c
dog	t	t	g	g	a	c	a	t	c
wolf	t	t	g	g	t	c	a	t	c

This is an example of DNA sequence data. Rows represent different species and columns polymorphic DNA sites. These data are obtained from sequencing DNA from the same gene from each of a number of different species and aligning these sequences. For typical genes from "closely related taxa" it is easy to align the DNA by hand or computer as the majority of sites with be monomorphic. In order to carry out a phylogenetic analysis

only the polymophic sites are informative, and as a result we only need work with a fraction of the sequence data. The `row.names` option to the `read.table` function tells R to make the first column of the data the name of each row.

I have written a function which will automatically construct a distance matrix for a set of DNA sequences. Such a matrix measures the number of DNA sequence differences separating all pairs of sequences divided by the total number of polymorphic sites examined. This function can be "loaded" into R using cut and paste on the code below.

```
seq.dist <- function(seq) {
  rs <- nrow(seq)
  cs <- ncol(seq)
  m <- matrix(nrow = rs, ncol = rs)
  for(c in 1:rs){
    for(r in 1:rs){
      x <- sum(apply(seq, 2, function(x) x[c] != x[r]))/cs
      m[c, r] <- x
    }
  }
  ans <- as.dist(m/2)
  attr(ans, "Labels") <- rownames(seq)
  return(ans)
}
```

The function will loop over all species pairs (that is all rows of `seq`) -- this is specified in the the `for(c in 1:rs)` and `for(r in 1:rs)` commands. For each pair of species the function will calculate the total number of sites at which species A and B differ and divide by the total number of sites (i.e., `sum(apply(seq,2,function(x) x[c] != x[r]))/cs`). For each comparison the function will take this value of "divergence" and use it to build a matrix whose [a,b]th element is the divergence between species A and B. I encourage you to attempt to understand how this function works – although it is possible to use it as a "black box".

Use `seq.dist()` to calculate the distance matrix associated with our example.

```
examp.dist <- seq.dist(examp)
```

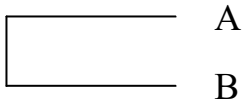
Each element of `example.dist` is the distance between two species. This matrix can now be used to construct a phylogenetic tree.

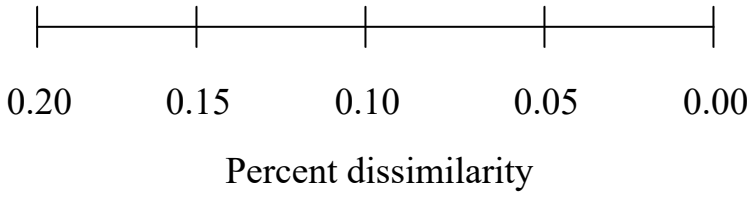
Box 2 UPGMA: Unweighted Pair Group Mean (from B. Gaut)

1. Assume the following distance matrix. The matrix can represent distances between sequences or phenetic distances

Species	A	B	C	D	E
A					
B	0.10				
C	0.28	0.32			
D	0.41	0.39	0.40		
E	0.39	0.41	0.40	0.20	

2. Start by choosing the two most closely related species and start to make a phylogeny by drawing branches between the two species. Take the distance and divide it by 2. For example the branches connecting species A and B should look like this:





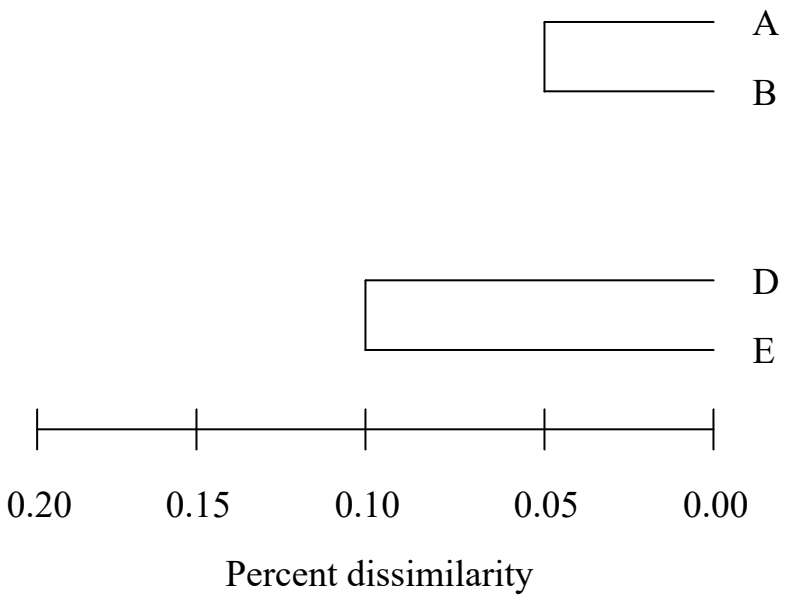
Notice the scale bar on the bottom. Notice also that the horizontal branch length going to B is length 0.05 (0.10/2). The vertical branch length doesn't really mean anything – it's just there to differentiate between the branch leading to A and the branch leading to B.

3. Re-make the dissimilarity matrix by combining species A and B. In the matrix above, species A is 0.28 different from species C and species B is

0.32 different from C; we combine by averaging – e.g., (0.28 + 0.32)/2 = 0.30. If you do this for all species, you will get a matrix like this:

Species	A-B	C	D	E
A-B				
C	0.30			
D	0.40	0.40		
E	0.40	0.40	0.20	

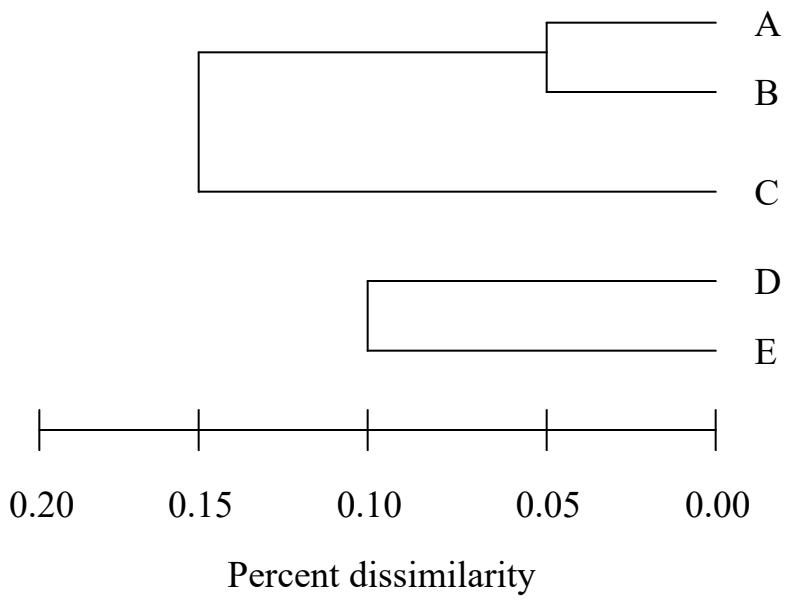
4. Choose the two most similar species from the matrix. The dissimilarity value between D and E is 0.20, and thus we draw branches of 0.20/2 = 0.10 in length. At this point, we only want to connect species D and E to each other; we don't want to connect them to A and B (that will come later). The phylogeny will look like this:



5. Now its time to once again remake the matrix, combining D and E in the process. You should get the matrix:

Species	A-B	C	D-E
A-B			
C	0.30		
D-E	0.40	0.40	

6. Combine A-B and C, as the pair with the least distance on the tree. Here's what the phylogeny should look like:



7. Recompute the matrix, combining A-B with C, and finish the tree.

Making a UPGMA tree in R

UPGMA trees can be conveniently calculated from a distance matrix in R. We will be using the function hclust(), which we will view as a "black box" function that we need to construct pyhlogenetic trees. What it does is equivalent to applying the UPGMA algorithm to the distance matrix generated earlier

```
examp.tree <- hclust(as.dist(0.5*examp.dist), method =
     "average")
```

In short `hclust` is a clustering algorithm, and when this algorithm is used with the method set to "average" it is equivalent to a UPGMA approach. The only other thing we have to do is pre-multiply the distance matrix by 0.5 (i.e, `0.5*example.dist`) so that the resulting tree is scaled to the total distance between sequences. Finally, we can plot the tree.

```
plot(examp.tree, labels = rownames(examp), hang = -1)
```

In this case we use the `labels` option to add our species names to the tree, and a `hang = -1` option to extend branches to the bottom of the tree.

Problem Set #5 (10 points)

1. Plant species often have multiple copies of the catalase genes in their genome. All copies of this gene are descended from one ancestoral copy, but some duplication events occurred prior to the diversification of "grasses" (maize, rice and barley are all grassed), and some duplicated within a specific lineage (for example maize). We define ***homologs*** as copies of a gene that trace their origin back to a *single common ancestor* of all grasses, and ***paralogs*** as copies of a gene that trace their origin back to *different copies present in the common ancestor* of grasses. Load the sequence data `liqing.txt` into R (use something like `liqing <- read.table("liqing.txt",row.names=1)`). Note that you will have to change your "working directory" so R can see liqing.txt. Use UPGMA to make a phylogenetic tree. Numbered genes are duplicate copies of the catalase genes in different plant species. Which copies of genes are likely to be homologs of one another? Which pairs of genes duplicated prior to the speciation events associated with maize, rice, and barley? Which pairs are likely to have duplicated since speciation?

2. Load the sequence data `andi.txt` into R (use something like `andi <- read.table("andi.txt")`). Use UPGMA to make a phylogenetic tree. Rows are different alleles of a gene called *Delta* obtained from *D. melanogaster*, and columns are polymorphic sites. Numbered alleles are all from a single population of wild caught flies from North Carolina; whereas SAM, and STANDARD are strains from elsewhere in N. America. Is there any evidence for "population structure" (different populations evolving independently) in *Drosophila*? Say lines 10, 15, 31, 46, 50, 51, 116, and SAM all had a dark body color phenotype in common. Where is the most parsimonious place in your tree for this body color mutation to have occurred?

3. Load the sequence data sets `peek1.txt` and `peek2.txt` into R (use something like `peek1 <- read.table("peek1.txt",row.names=1)`). Use UPGMA to make a phylogenetic tree for each data set. For each data set rows are different strains of *E. coli* and columns are polymorphic sites. The data set peek1 is for a gene called *mdhs1* and for data set peek2 is for a gene called *fimas1*. These two genes are at different positions in the *E. coli* genome. Bacterial geneticists have often claimed that *E. coli* rarely recombines in the wild. A corollary of this claim is that different genes in *E. coli* should give the same phylogenetic tree since the genome is always inherited vertically as a unit. An alternative hypothesis is that different strains of *E. coli* in nature exchange DNA sequences (that is experience recombination). A corollary of this hypothesis is that different

genes should give different trees. Which hypothesis does your data support? Are there public health implications associated with this result?

April 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	3 Place drift flies in vials	4	5 Empty drift vials Start and finish sexual selection experiment	6	7	8
9	10 Statistics	11	12 Statistics with R	13	14	15
16	17*Census drift flies & transfer *Start age-specific selection *Start age-specific fecundity assay	18	19*Empty drift flies *Count eggs from age experiment, *Start natural selection lab,***SS lab report due 1PM**	20	21	22
23	24 Empty flies from natural selection expt *Start age-specific fecundity assay *Census adult populations	25	26 *Count eggs from age expt Problem Set #1 **due** 4PM	27	28	29

30	**May 1** *Census adult populations *Start age-specific fecundity assay *Census/transfer drift	2	3*Count eggs from age expt *Census selection flies *Empty drift flies	4	5	6

May 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7	8*Census adult populations *Start age-specific fecundity assay	9	10 *Count eggs from age expt Work on PS#2	11	12	13
14	15*Census drift flies and end expt,*Census adult populations, *Start age-specific fecundity assay, Problem Set 2 **due 4 PM**	16	17 Count eggs from age expt Monte Carlo @2:30PM	18	19	20
21	22 Monte Carlo ***Drift lab due 1PM**	23	24 Monte Carlo Problem Set #3 **due 1PM**	25	26	27
28	29 Work on PS #4 ***NS lab due 1PM**	30	31 Work on PS #4 Problem Set #4 **due 4PM**	**June 1**	2	3

4	5 Phylogeny	6	7 Work on PS #5 ***AS lab due 4PM**	8	9	10

June 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
11	12 Problem Set 5 **due 5PM**	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	