MATH 420/620 HW #1 SOLUTIONS

Instructions: A printed copy of your homework is due at **the start of class**. Supplementary electronic files (e.g. R scripts or wxMaxima files) should be emailed to the instructor prior to class with file name format LASTNAME-HWX.EXT (send multiple files in a single ZIP file).

Exercise 1. (15 pts) List three probability distributions (besides the Normal distribution) commonly used in your major field of study, and for each of these describe

(a) the physical/real-world processes that lead to each distribution, or otherwise justify their widespread use;

(b) any useful associations between those real-world processes, the parameters of the distribution, and the mean and/or variance (or similar quantities) of that distribution; and

(c) select plausible parameter values (one set for each distribution) and plot the histogram of a large sample and the corresponding density/mass function.

Solution: Variable. The instructor will put these into a summary table that complements the distribution table discussed in class.

Exercise 2. (15 pts) Neutral Allele's under Wright-Fisher. The Wright-Fisher model is a simplified model of how allele frequencies in a population of N individuals changes from generation to generation. It assumes that, in one time step of the model (i.e., one generation time), the N individuals are replaced by the next generation of N individuals. Assume there are two different types of individuals (e.g., a *normal* type **A** and a *mutant* type **B**) and further assume that the mutant type has the same (*neutral*) fitness (i.e., neither type is more or less likely than the other to contribute offspring to the next generation). In this case, the rule for populating the next generation is to randomly sample from the parent population with replacement, which yields a binomially distributed number of type **B** individuals with probability n/N, where n is the number of **B** individuals in the parent population.

(a) Write down the probability of the **B** type going extinct in the next generation given that n of the N individuals in the parent population are type **B**.

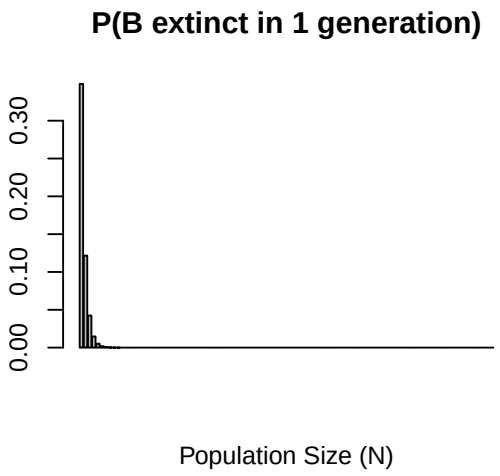
Solution: If we let X be the random variable representing the number of B-type individuals in the next generation, then $X \sim$Binomial(N,n/N) and thus

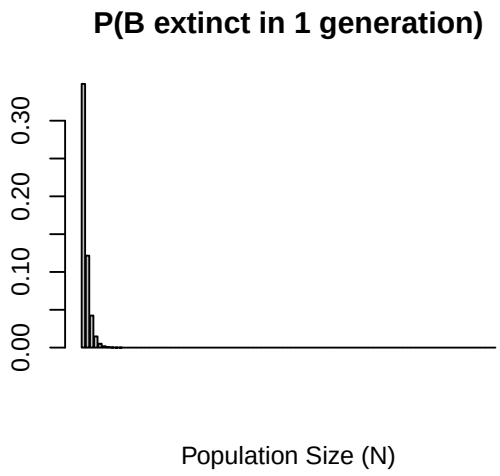
$$\begin{aligned} P(X=0) =& \binom{N}{0}(n/N)^0(1-n/N)^{N-0} \\ =& \binom{N}{0}(1-n/N)^N \\ =& (1-n/N)^N \end{aligned}$$

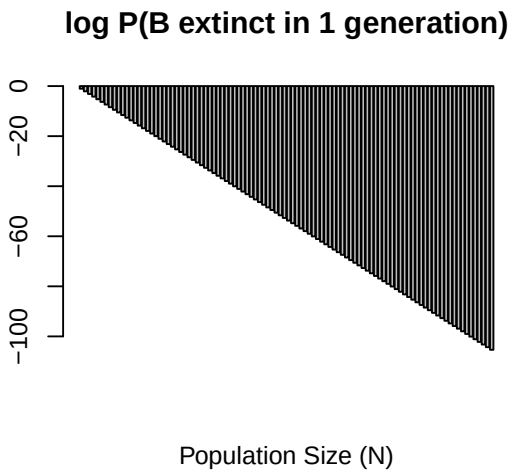
(b) How does population size impact extinction probabilities? Plot this probability (y axis) for the following population sizes (x axis) assuming 10% of the population are mutants, and discuss: $N \in \{10, 20, 30, 40, \ldots, 1000\}$.

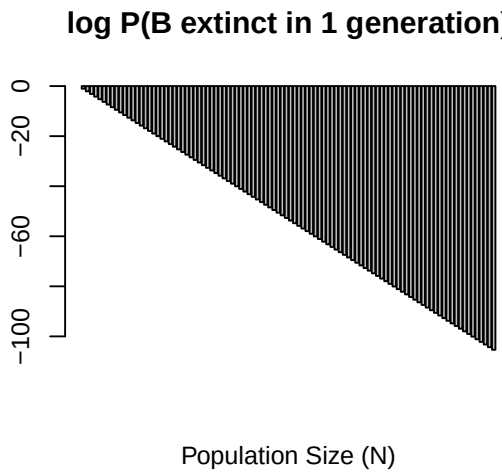
Solution: Since $P(X = 0) = (1 - n/N)^N$, where $n/N = 0.1$ we can calculate these next-step extinction probabilities with the following R code (the left column is calculated directly, the right column was calculated using R's built-in probability mass function for the Binomial distribution).

```
N = seq(10,1000,by=10);
P.extinct = (1-0.1)^N;
# Or alternatively, just use R's built-in probability functions (see ?dbinom)
P.extinct2 = dbinom(0, size=N, prob=0.10)
par(mfrow=c(2,2)) # See http://www.statmethods.net/graphs/bar.html
barplot(P.extinct, main="P(B extinct in 1 generation)", xlab="Population Size (N)")
barplot(P.extinct2, main="P(B extinct in 1 generation)", xlab="Population Size (N)")
barplot(log(P.extinct),main="log P(B extinct in 1 generation)",xlab="Population Size (N)")
barplot(log(P.extinct2),main="log P(B extinct in 1 generation)",xlab="Population Size (N)")
```







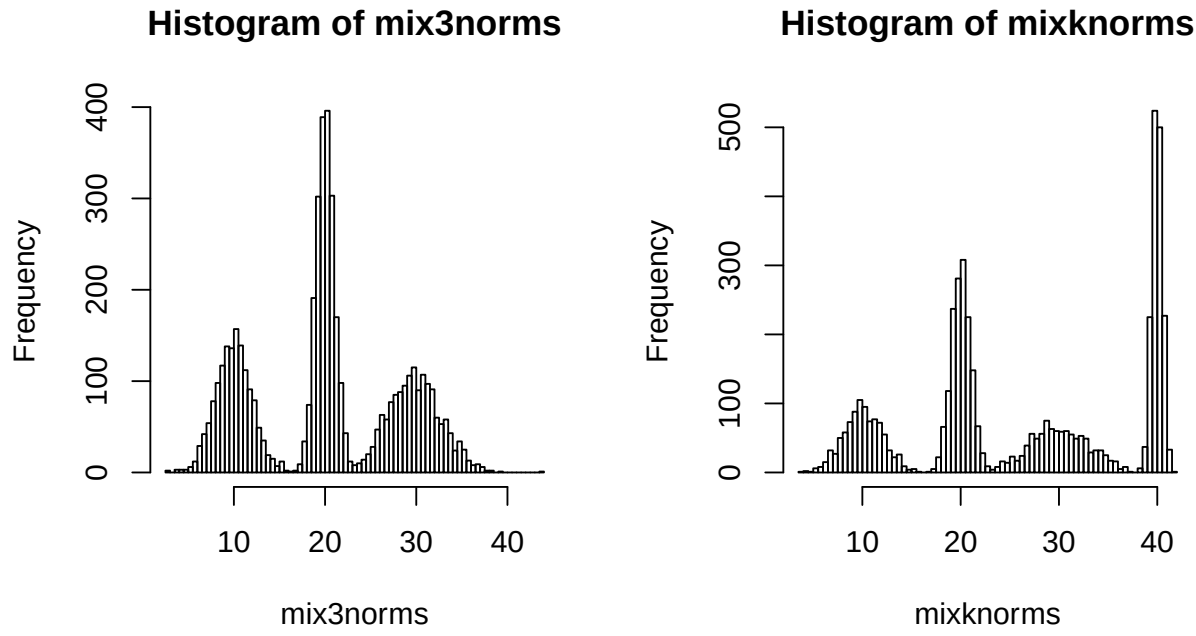


Exercise 3. (10pts) Mixture Distributions.

(a) The R code below generates a random sample from a mixture of two Normal distributions. Modify it generate mixture of three (or more) Normals.
Solution: Here are extensions to 3 Normals or an arbitrary number of normals.

```
par(mfrow=c(1,2))
# Mixture of 3 normals...
rnormmix3 <- function(n, mean1, mean2, mean3, sd1, sd2, sd3, p1, p2, p3) {
  ps=c(p1,p2,p3)/(p1+p2+p3) # This ensures these sum to 1!
  means=c(mean1,mean2,mean3)
  sds=c(sd1,sd2,sd3)
  indx = sample(1:3,n,replace=TRUE,prob=ps)
  return(rnorm(n, mean=means[indx], sd=sds[indx]))
}
# Example:
mix3norms = rnormmix3(5000, mean1=10, mean2=20, mean3=30,
                          sd1=2, sd2=1, sd3=3, p1=0.3, p2=0.4, p3=0.3)
hist(mix3norms,100)

# Mixture of k normals.
rnormmix <- function(n, mean=0, sd=1, p=1) {
  # Check lengths of means, sds and probs to ensure the match!
  if(length(mean) != length(sd) | length(mean) != length(p) ) {
    stop("Error: mean, sd, and p should be the same length!")
  }
  indx = sample(1:length(p),n,replace=TRUE,prob=p)
  return(rnorm(n, mean=mean[indx], sd=sd[indx]))
}
# Example
mixknorms = rnormmix(5000, mean=c(10,20,30,40),sd=c(2,1,3,.5),p=c(.2,.3,.2,.3))
hist(mixknorms,100)
```



(b) The Negative Binomial distribution with rate r and probability p can be viewed as a *compound* distribution (aka a *continuous mixture* distribution) where each observation is drawn from a Poisson whose rate parameter λ is not constant, but instead is sampled from a Gamma distribution, which -- to quote the Wikipedia page https://en.wikipedia.org/wiki/Negative_binomial_distribution as of 5pm on 9/8/2017 -- is parameterized by "shape $= r$ and scale $\theta = p/(1-p)$ or correspondingly rate $\beta = (1-p)/p$". The R code below compares samples from these two distributions using the above parameterization, but there's a problem! **Correct the code, and explain the source of the error.**

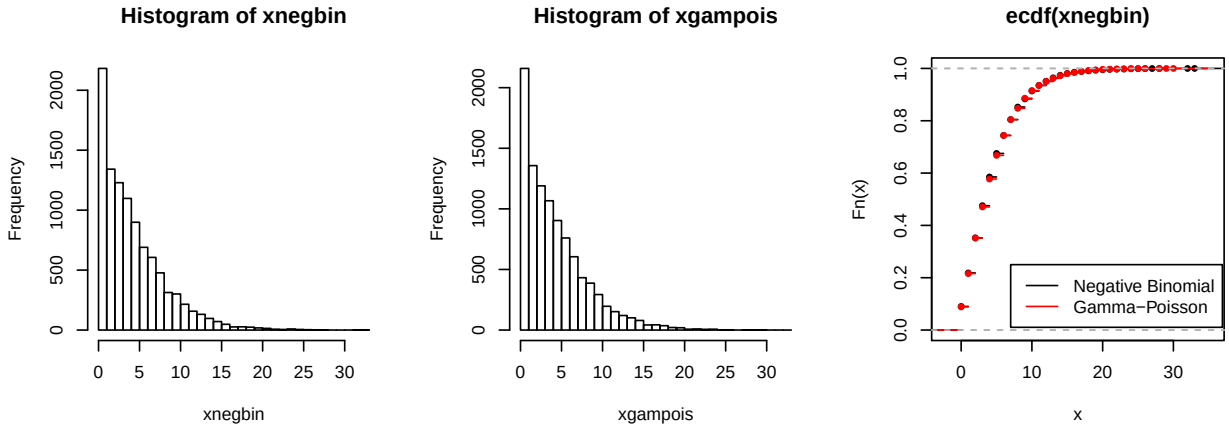
Solution: The problem is that the interpretation of p differs between the implementation of Negative Binomial in R and the version described on the Wikipedia page, resulting in *scale* and *rate* appearing to be mixed up on the Wikipedia page (in the context of Gamma distribution parameterizations, note that $scale = 1/rate$). In R, the documentation for `rnbinom` reads

> The negative binomial distribution with $size = n$ and $prob = p$ has density
>
> $$\Gamma(x+n)/(\Gamma(n)x!)p^n(1-p)^x$$
>
> for $x = 0, 1, 2, \ldots, n > 0$ and $0 < p \leq 1$.
>
> This represents the **number of failures** which occur in a sequence of Bernoulli trials before a target number of successes is reached. The mean is $\mu = n(1-p)/p$ and variance $n(1-p)/p^2$.

Compare that to the wikipedia description where "$p \in (0, 1)$ -- **success probability** in each experiment". Thus, to fix our code swap p and $1-p$, i.e., use $scale = (1-prob)/prob$.

```
rgampois <- function (n, r, prob) {
  rpois(n, lambda = rgamma(n, shape=r, scale=(1-prob)/prob))  # WRONG: scale=p/(1-p)
}
r=2; p=0.3
xnegbin  <- rnbinom(10000, size = r, prob = p)
xgampois <- rgampois(10000, r, p)

par(mfrow=c(1,3)) # Two subplots, arranged in 1 row, 2 columns
hist(xnegbin,breaks = 0:max(xnegbin,xgampois))
hist(xgampois,breaks = 0:max(xnegbin,xgampois))
plot(ecdf(xnegbin)); plot(ecdf(xgampois), col="red", add=TRUE)
legend(7,.25, c("Negative Binomial","Gamma-Poisson"),
          col=c("black","red"), lty=c(1,1))
```



Instructor Provided R Code

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Exercise 3a

rnormix <- function(n, mean1, mean2, sd1, sd2, p1, p2) {
  # n = sample size, ps=c(p1, p2) are the mixing probabilities
  # means = c(mean1,mean2) and sds = c(sd1,sd2) the distributions.
  ps=c(p1,p2)/(p1+p2) # This ensures these sum to 1!
  means=c(mean1,mean2)
  sds=c(sd1,sd2)
  # First pick which distribution each observation comes from...
  indx = sample(c(1,2),n,replace=TRUE,prob=ps)
  # next give the corresponding vector of means and sds to rnorm()...
  return(rnorm(n, mean=means[indx], sd=sds[indx]))
  # see ?rnorm for details.
}

# Example:
fakedata = rnormix(5000, mean1=10, mean2=20, sd1=2, sd2=1, p1=0.2, p2=0.8)
hist(fakedata,50)


# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Exercise 3b

rgampois <- function (n, r, prob) {
  # use a different lambda for each observation
  # parameterized using scale=p/(1-p) as described on wikipedia
  rpois(n, lambda = rgamma(n, shape=r, scale=prob/(1-prob)))
}

r=2
p=0.3
xnegbin  <- rnbinom(5000, size = r, prob = p)
xgampois <- rgampois(5000, r, p)

x11() # try quartz() if this doesn't work on your mac!
par(mfrow=c(1,2)) # Two subplots, arranged in 1 row, 2 columns
hist(xnegbin,breaks = 0:max(xnegbin,xgampois))
hist(xgampois,breaks = 0:max(xnegbin,xgampois))
```