Problem Set #2:

1. *Mass versus density*: a) what is the difference between a probability mass function and a probability density function? b) what is the difference between a random variable and the parameters of a probability distribution?
2. *Convergence Results*: Set up R code to demonstrate the following:
   1. Convergence of binomial distribution to normal distribution for various combinations of N and p.
   2. Convergence of Poisson distribution to normal distribution as lambda parameter grows
3. *Moment matching*: Apply the technique of moment matching to the case of the Gamma distribution where you are interested in modeling the density as a function of its mean, and its skewness. For simplicity, use the shape-rate parameterization of the Gamma. Write out a few lines of R code sketching the resulting density for a fixed mean, under varying scenarios of skewness. This should act as a way to double check your algebraic results.
4. *Coin flipping A*. Suppose you flip a coin with a known bias *p* 5, 10, 50, or 100 times, and you compute the proportion of heads in each trial.
   1. (Monte Carlo analysis): Write some lines of code that simulate this experiment repeatedly using a minimum of 10^4 repetitions (save these values into an array, matrix or data-frame). Use this numerical experiment to place approximate bounds on the expected value of the proportion within an interval with 95% probability.
   2. As the trial size increases, what do you notice about the distribution of this interval?
   3. Based on your answers in ‘a’ and ‘b’, derive a formula for placing bounds this expected value of the proportion.
5. Coin flipping B: Let us say you do NOT know the bias *p* and your goal is to learn about this unknown quantity from data using maximum likelihood.
   1. Write down the maximum likelihood estimator for the parameter *p*.
   2. For each of your simulated datasets from above, apply the maximum likelihood estimator, and plot the results in a histogram (one for each sample size N).
   3. Compare the Monte Carlo standard errors from ‘b’ to the point estimate of standard error from each simulated dataset.
6. Simulating the logistic growth model. For this problem, we will work with the logistic growth model as follows:
7. What happens to dN/dt as N goes to Inf?
8. Where does dN/dt attain its maximum value, and what value does it have there?
9. Plot simulated trajectories overlaid on each other for various choices of r and K.

For simulation purposes, we will need to discretize this model, and initialize a state vector at some value.

Demo script:

r = [some value];

K = [some value];

T = [some value];

N = rep(NA,T);

N[1] = [some value];

for(i in 2:T){

N[i] = N[i-1] + r\*N[i-1]\*(1-(N[i-1]/K))

}

1. Now, add observation error by modifying your base model so that you have a new variable M, where M = N + rnorm(1,0,error)
2. Iteratively repeat step d 10^3 times and overlay each observed trajectory into one plot
3. Instead of observation error, consider that you may have process error to worry about. Simulate 10^3 trajectories where your ‘algorithm’ is:

r = [some value];

K = [some value];

T = [some value];

error = [some value];

N = rep(NA,T);

N[1] = [some value];

for(i in 2:T){

N[i] = N[i-1] + r\*N[i-1]\*(1-(N[i-1]/K)) + rnorm(1,0,error);

}

1. {CHALLENGE}: for the basic case of normal observation error only on a single trajectory, given a known N[1]=1, known K, and known variance, derive a maximum likelihood estimator of r. What is the standard error of this estimate? Hint: work with a *restricted subset* of data-points in the middle, approximately linear, portion of the sigmoidal curve.