# Session 2: Introduction to Probability

## Foundations of Quantitative Ecology (EEOB 8896.11)

Paul J. Hurtado, PhD

Mathematical Biosciences Institute (MBI)

August 28, 2013

# Why Probability?

One answer: **Statistics**.

# Why Probability?

One answer: **Statistics**.

But more importantly...

- Biological processes are noisy! (See Jagers 2010)

# Why Probability?

One answer: **Statistics**.

But more importantly...

- Biological processes are noisy! (See Jagers 2010)
- The fundamental units in biology are *individuals*.
  Thus, *Demographic (intrinsic)* noise is commonplace.

# Why Probability?

One answer: **Statistics**.

But more importantly...

- Biological processes are noisy! (See Jagers 2010)
- The fundamental units in biology are *individuals*.
  Thus, *Demographic (intrinsic)* noise is commonplace.
- Environments are constantly changing!
  Thus, *Environmental (extrinsic)* noise is also ubiquitous.

# Why Probability?

One answer: **Statistics**.

But more importantly...

- Biological processes are noisy! (See Jagers 2010)
- The fundamental units in biology are *individuals*.
  Thus, *Demographic (intrinsic)* noise is commonplace.
- Environments are constantly changing!
  Thus, *Environmental (extrinsic)* noise is also ubiquitous.

In short, **ALL models of living systems are, or are simplifications of, stochastic models.**

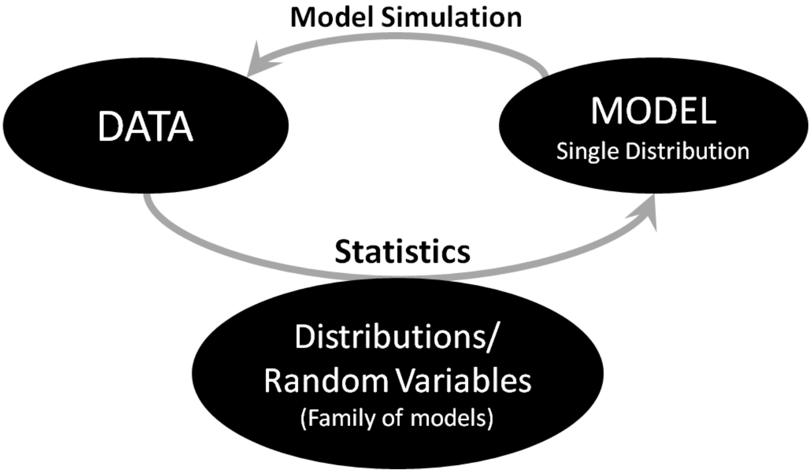
# Why Probability?

So, *how/why* are probability concepts widely used in science?

A few examples...

- Simulation (sampling from distributions, e.g., to mimic data)
- Qualitative properties (expected values, expected deviations)
- Approximation (Law of Large Numbers)
- Deriving relationships (e.g., functional forms) and other models
- Statistics (e.g., Maximum Liklihood = Maximum density!)

# Conceptual Framework



# Simulation

Example: Linear Regression $y = \beta_0 + \beta_1 x + \epsilon$, where $\epsilon \sim N(0, \sigma)$.

```
set.seed(1492); ## ?set.seed or ask me :-)
b0=2; b1=1; sig=2;   y=b0+b1*x+rnorm(length(x),0,sig);

## Error:  object 'x' not found

plot(x,y,pch=19); abline(b0,b1);

## Error:  object 'x' not found

abline(lm(y~x,data=data.frame(x,y)),lty=2)

## Error:  object 'x' not found
```

# Distribution Properties

Mean vs Expected value? Standard Deviation? Moment Generating Function? Conjugate Distributions (Baysian prior & posterior)?

```
x = rbinom(100, 20, p = 0.2)
mean(x)  ## Compare mean(x) vs. E(x)=n*p

## [1] 4.04

sd(x)  ## Compare sd(x)^2 vs. Var(x)=n*p*(1-p)

## [1] 1.693

sqrt(20 * 0.2 * (1 - 0.2))

## [1] 1.789
```

General mathematical results (aka Analytical results) are really powerful, *if* we can find them! They give general answers to our scientific questions, guide biological intuition, and speed up computations.

# Approximation & Deriving Other Models

**Computation:** Gillespie's Stochastic Simulation Algorithm is driven by "coin tosses" (aka *Bernoulli random variables*) – to speed up computations, approximate multiple coin tosses with a single *binomial distribution*.

# Approximation & Deriving Other Models

**Computation:** Gillespie's Stochastic Simulation Algorithm is driven by "coin tosses" (aka *Bernoulli random variables*) – to speed up computations, approximate multiple coin tosses with a single *binomial distribution*.

**Statistical assumptions:** Error distributions may be known (for mechanistic reasons) to be, e.g., Binomial, which can sometimes be approximated by a Normal distribution.

# Approximation & Deriving Other Models

**Computation:** Gillespie's Stochastic Simulation Algorithm is driven by "coin tosses" (aka *Bernoulli random variables*) – to speed up computations, approximate multiple coin tosses with a single *binomial distribution*.

**Statistical assumptions:** Error distributions may be known (for mechanistic reasons) to be, e.g., Binomial, which can sometimes be approximated by a Normal distribution.

**Dynamics:** We might want to ignore the noise, and just look at averages. Ex: Lotka-Volterra-type foodweb models are really useful!

# Approximation & Deriving Other Models

**Computation:** Gillespie's Stochastic Simulation Algorithm is driven by "coin tosses" (aka *Bernoulli random variables*) – to speed up computations, approximate multiple coin tosses with a single *binomial distribution*.

**Statistical assumptions:** Error distributions may be known (for mechanistic reasons) to be, e.g., Binomial, which can sometimes be approximated by a Normal distribution.

**Dynamics:** We might want to ignore the noise, and just look at averages. Ex: Lotka-Volterra-type foodweb models are really useful!

**General Results:** Model approximation (or considering special cases of a model) can yield well understood (approximate) models for which useful, general results already exist!

# Statistics: Maximum Liklihood

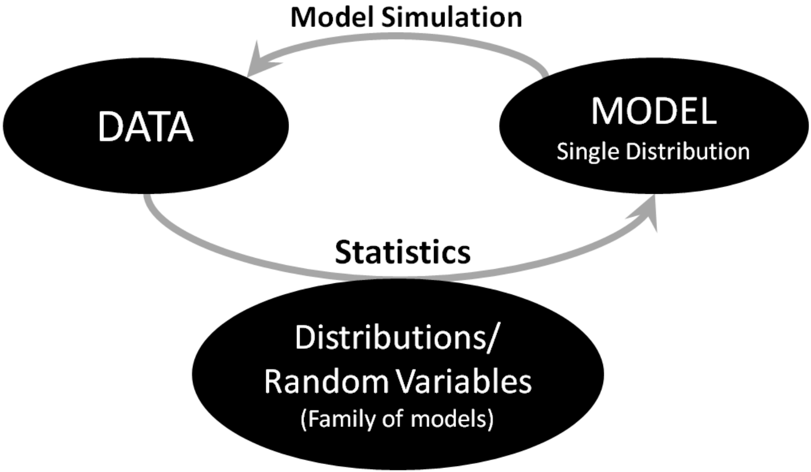
Up to this point, we think of density functions as having fixed parameters $\theta = (\theta_1, ..., \theta_k)$, with arbitrary input value $x$. *Liklihood functions* are **the exact same functions** except the "inputs" are fixed data values $x_1$, ... $x_n$ and our parameters are the arbitrary inputs of interest. Specifically, we want the parameters that maximize our likelihood function value for this particular data set.

# Statistics: Maximum Liklihood

Up to this point, we think of density functions as having fixed parameters $\theta = (\theta_1, ..., \theta_k)$, with arbitrary input value $x$. *Liklihood functions* are **the exact same functions** except the "inputs" are fixed data values $x_1$, ... $x_n$ and our parameters are the arbitrary inputs of interest. Specifically, we want the parameters that maximize our likelihood function value for this particular data set.

To see how all this works, we need to start looking at probability distributions in detail.

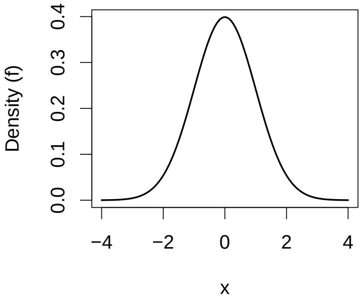
# Conceptual Framework

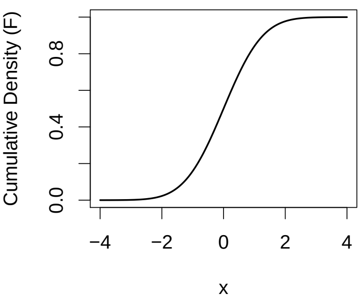


# Distribution & Density Functions

Two ways to think about the Normal distribution (a *continuous* distribution) from the relationship: $F(x) = \int_{-\infty}^{x} f(s)ds$

```
## Standard normal density function f(x) and distribution function F(x)
par(cex = 1.4)
x = seq(-4, 4, length = 200)
plot(x, dnorm(x, mean = 0, sd = 1), type = "l", lwd = 2, ylab = "Density (f)")
plot(x, pnorm(x), type = "l", lwd = 2, ylab = "Cumulative Density (F)")
```

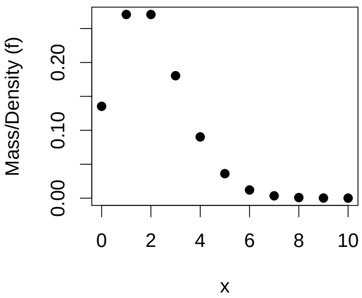


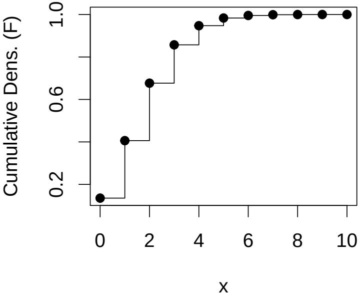



# Distribution & Density Functions

Discrete distributions: replace integrals with sums. $F(x) = \sum_{i=0}^{x} f(i)$

```
## Poisson (mean 2) density and distribution functions
x = 0:10
par(cex = 1.4)
plot(x, dpois(x, lambda = 2), pch = 19, ylab = "Mass/Density (f)")
plot(x, ppois(x, lambda = 2), type = "s", ylab = "Cumulative Dens. (F)")
points(x, ppois(x, lambda = 2), pch = 19)
```



# Exercises

1. Look up which distributions are approximately normal (and for which parameter values), and demonstrate this graphically in R. This may (or may not) be helpful: http://www.math.wm.edu/~leemis/chart/UDR/UDR.html
2. Use the code on previous slides (or your own) and plot the density and distribution functions for these distributions. For each distributiono, do this in a 2x2 figure. In the top row, compare to a normal distribution with the same mean and variance. In the bottom row, do the same but with parameters where the normal approximation fails.
3. For the programmers: Too easy? Automate this with a for loop, or use the lattice or ggplot2 packages for the graphics.