Simulating Data in R

General Strategy

The basic approach to simulating data—making "fake" data—is to (1) take the statistical model, (2) pick some parameter values, and (3) use the random form of the model's probability distribution, with the model embedded inside it, to generate random samples. So for a model $y_i \sim \text{Zink}(\alpha + \beta x_i, \gamma)$, data can be simulated from the model using:

```
a <- 3
b <- 0.5
g <- 12
y <- rzink( n , a + b*x , g )
```

where `a`, `b`, and `g` are the parameters, `x` is a vector of (possibly fake) prediction values, and `n` is equal to the length of `x`, the number of `y` values to generate.

Gaussian Example

Suppose we have a Gaussian regression model, $y_i \sim \mathcal{N}(\alpha + \beta_1 x_i + \beta_2 z_i, \sigma)$. You know the built-in density function for Gaussian values is `dnorm`. The corresponding function to generate random Gaussian values is `rnorm`—this is almost always the case, that the random function is named after the density function, by changing the initial "d" to a "r". To simulate:

```
a <- 7.2
b1 <- 2.1
b2 <- -5
s <- 3
y <- rnorm( length(x) , mean=a + b1*x + b2*z , sd=s )
```

The vector `y` will hold randomly sampled Gaussian values.

Common Distributions

Distribution	Function	Example
Normal	`rnorm`	`y <- rnorm(length(x),mean=a+b*x,sd=s)`
Binomial	`rbinom`	`y <- rbinom(length(x),prob=logit(a+b*x),size=n)`
Poisson	`rpois`	`y <- rpois(length(x),lambda=a+b*x)`
Gamma	`rgamma`	`y <- rgamma(length(x),shape=a+b*x,scale=th)`
Negative binomial	`rnbinom`	`y <- rnbinom(length(x),mu=a+b*x,size=n)`
		–or–
	`rgamma/rpois`	`y1 <- rgamma(length(x),shape=a+b*x,scale=th)`
		`y2 <- rpois(length(y1),lambda=y1)`
Exponential	`rexp`	`y <- rexp(length(x),rate=a+b*x)`
Beta-binomial	`rbeta/rbinom`	`y1 <- rbeta(length(x),shape1=a+b*x,shape2=s2)`
		`y2 <- rbinom(length(y1),prob=y1,size=n)`