Regression analysis: Extensions

Patrick Breheny

February 12

Introduction

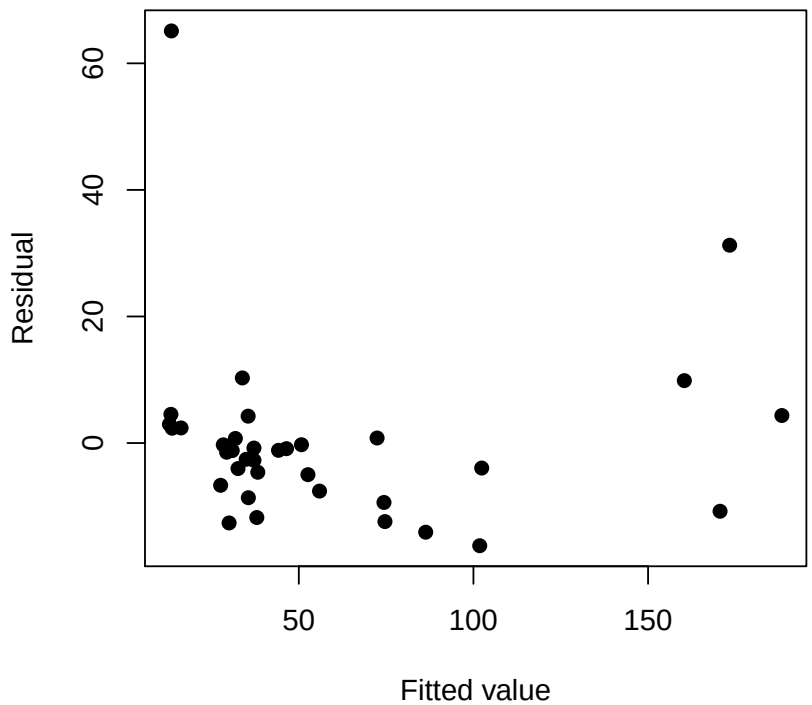
- Last time we explored the basic linear regression model from a Bayesian perspective
- Today, we will look at various ways in which the linear regression model can be extended
- Specifically, we will consider four examples, all of which have some sort of frequentist analog; however, as we will see, the flexibility of the Bayesian approach makes these and further extensions straightforward to implement and interpret

Scottish hill races

- Our first extension is *robust regression*: the normality assumption of OLS linear regression renders it quite sensitive to outliers
- A classic data set in the this literature pertains to hill racing (apparently a somewhat popular sport in Scotland)
- The data set `hills.txt` contains information on the winning times in 1984 for 35 Scottish hill races, as well as two factors which presumably influence the duration of the race:
  - `dist`: The distance of the race (in miles)
  - `climb`: The elevation change (in feet)

Residuals from OLS fit

Fitting a simple linear regression model for `time` assuming additive linear relationships for `dist` and `climb`, we obtain the following residuals:



Using a thicker-tailed distribution

- To reduce the impact of the fit of the model, we can replace the normal distribution with a thicker-tailed distribution
- A natural choice is the t-distribution, which can be implemented by simply replacing the normal likelihood with:

```
time[i] ~ dt(mu[i], tau, nu)
```

- Recall that as $\nu \to \infty$, the t-distribution resembles the normal, but for small ν has considerably thicker tails

Contrasting the two posteriors

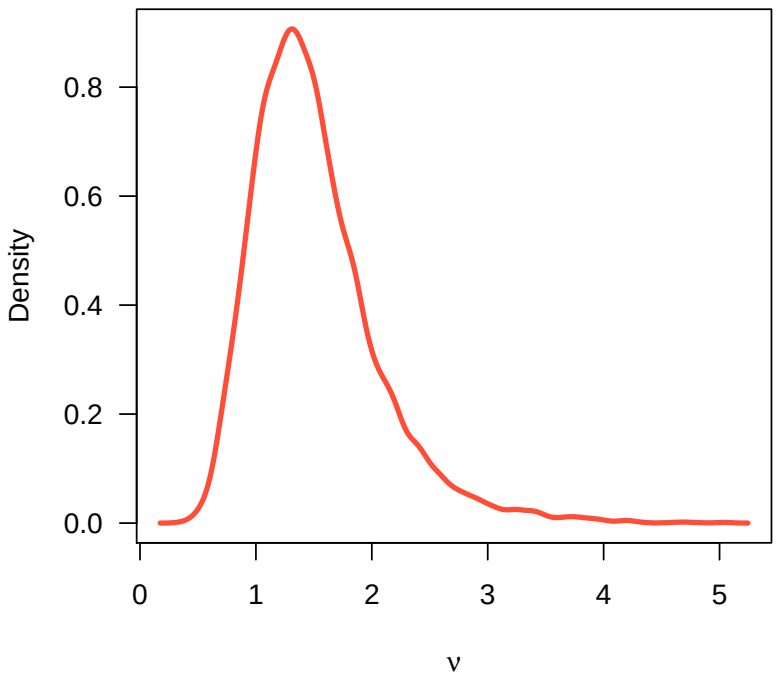
	Climb (100 fit)		Dist. (1 mi)	
	Mean	SD	Mean	SD
Normal	1.11	0.21	6.21	0.62
t_5	0.81	0.15	6.59	0.29

Recall that there are two large outliers in this data set; as they are in some sense downweighted, there is a modest change in the posterior means (the posterior mean for distance goes up, while the one for climb goes down), and a sizeable drop (roughly 2-fold) in the posterior SD

Setting a prior on ν

- Of course, one may ask, why a t_5 distribution?
- Since we do not actually know ν, it would be more reasonable to include ν as a parameter in our model; the only condition is that we must place a prior on it
- A reasonably uninformative prior would seem to be $\nu \propto \nu^{-1}$; *i.e.*, `nu ~ dgamma(.001,.001)` in BUGS

Posterior



Climb (100 ft):

	Mean	SD
Normal	1.11	0.21
t_5	0.81	0.15
$\nu \propto \nu^{-1}$	0.69	0.10

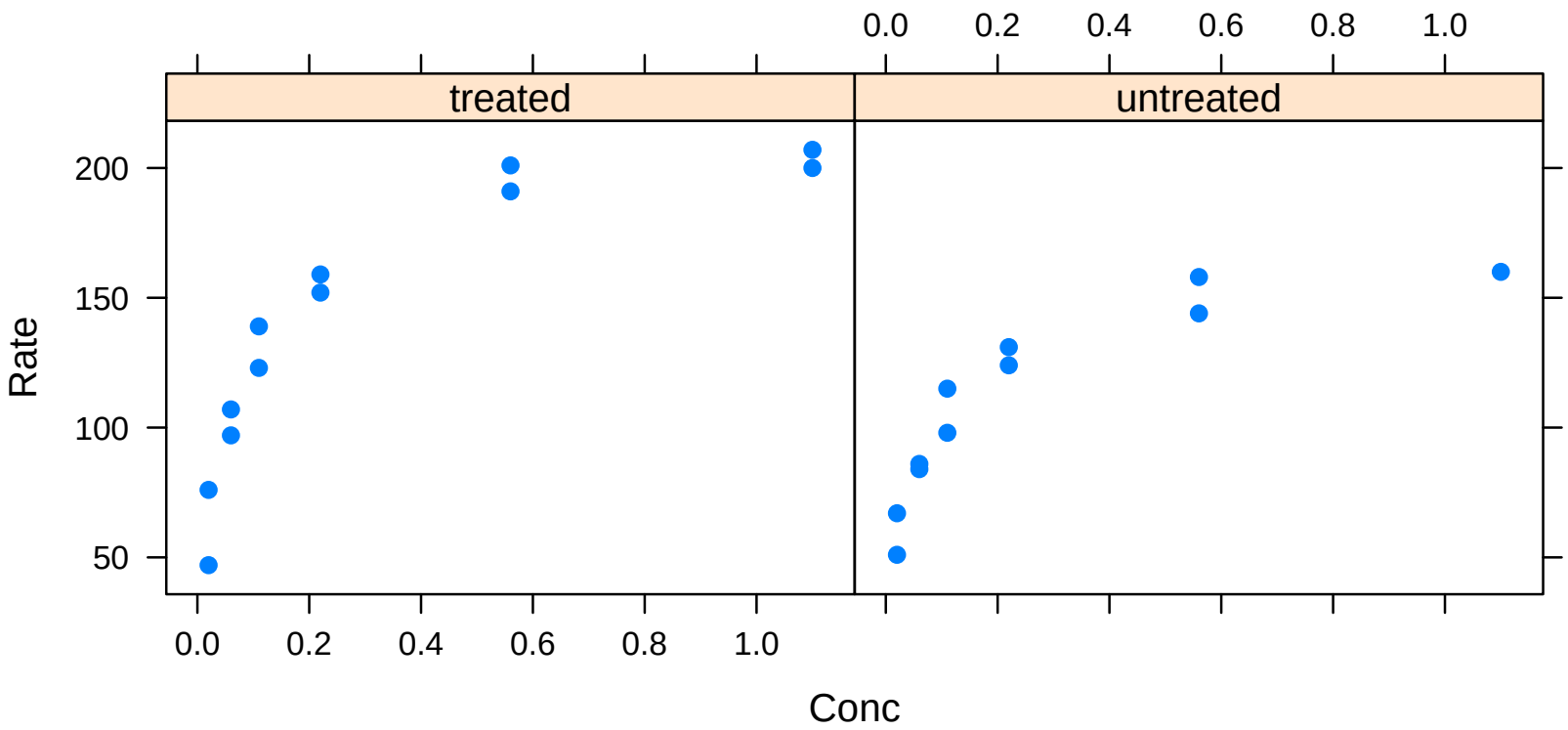
Distance (1 mi):

	Mean	SD
Normal	6.21	0.62
t_5	6.59	0.29
$\nu \propto \nu^{-1}$	6.56	0.24

Puromycin data

- Another desirable extension is the ability to fit a nonlinear model
- Our illustrating data set here is a study of the reaction kinetics of an enzyme called galactosyltransferase
- The recorded variables are `Conc`, the concentration of the enzyme's substrate (in ppm) and `Rate`, the reaction rate (in DPM/min)
- Furthermore, there were two experimental groups: 12 from cells treated with an antibiotic called puromycin, and an untreated control group of sample size 11

Puromycin data: Illustration



MichaelisMenten kinetics

The standard model for the study of these kinds of reactions is the *Michaelis-Menten* model:

$$v = \frac{V_m[S]}{K + [S]},$$

where v is the reaction rate, $[S]$ is the substrate concentration, V_m is the maximum rate, and K is the substrate concentration at which v is one-half V_m

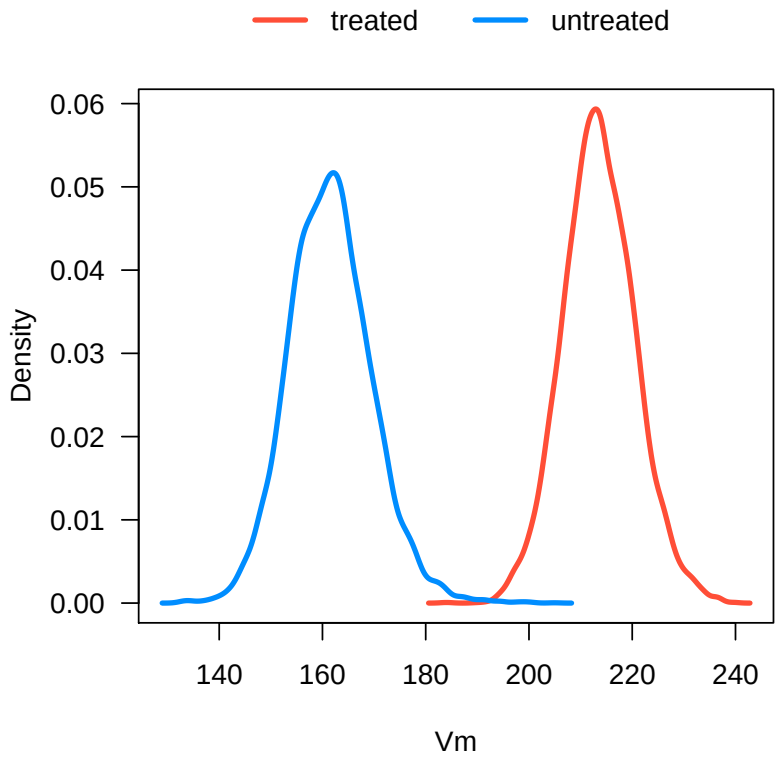
BUGS code

We can implement this model in BUGS as follows:

```
for (i in 1:n) {
  Rate[i] ~ dnorm(mu[i], tau)
  mu[i] <- (Vm[State[i]]*Conc[i])/(K[State[i]]+Conc[i])
}
for (j in 1:2) {
  Vm[j] ~ dunif(0,500)
  K[j] ~ dunif(0,2)
}
tau ~ dgamma(0.001, 0.001)
```

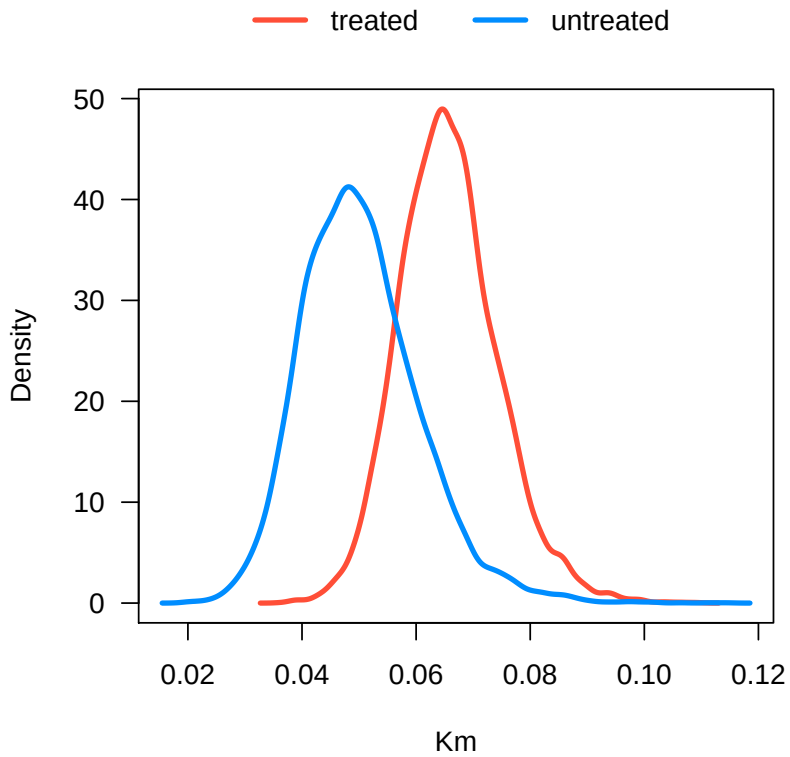
Note the use of "nested indexing" to match observation i up with its correct state

Posterior: V_m



		90% CI	
	Mean	Lower	Upper
Treated	213.6	202.3	225.4
Untreated	161.5	149.0	175.1
Δ	52.0	35.0	69.6

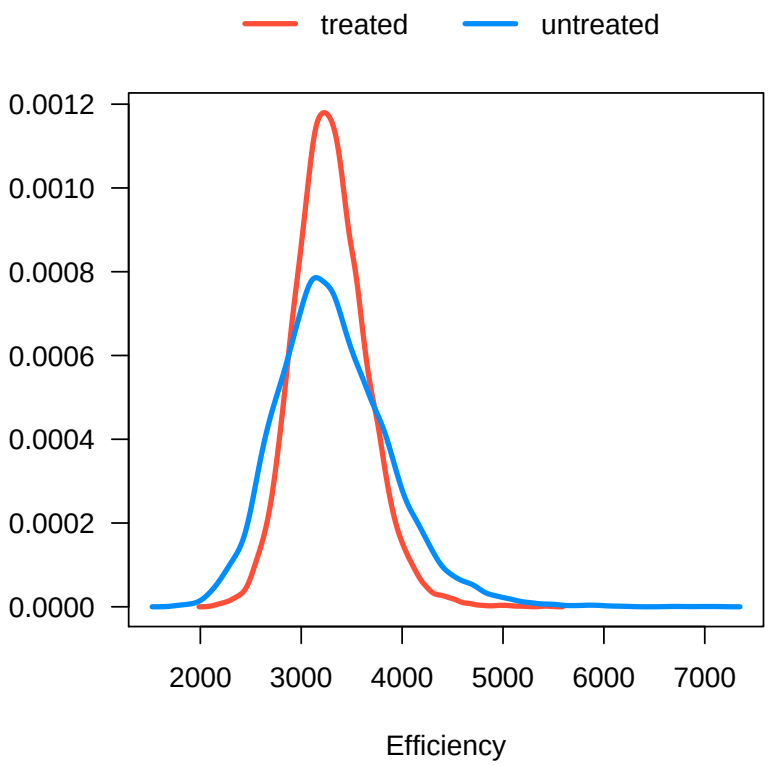
Posterior: K



		90% CI	
	Mean	Lower	Upper
Treated	0.066	0.052	0.081
Untreated	0.050	0.035	0.068
Δ	0.015	-0.006	0.037

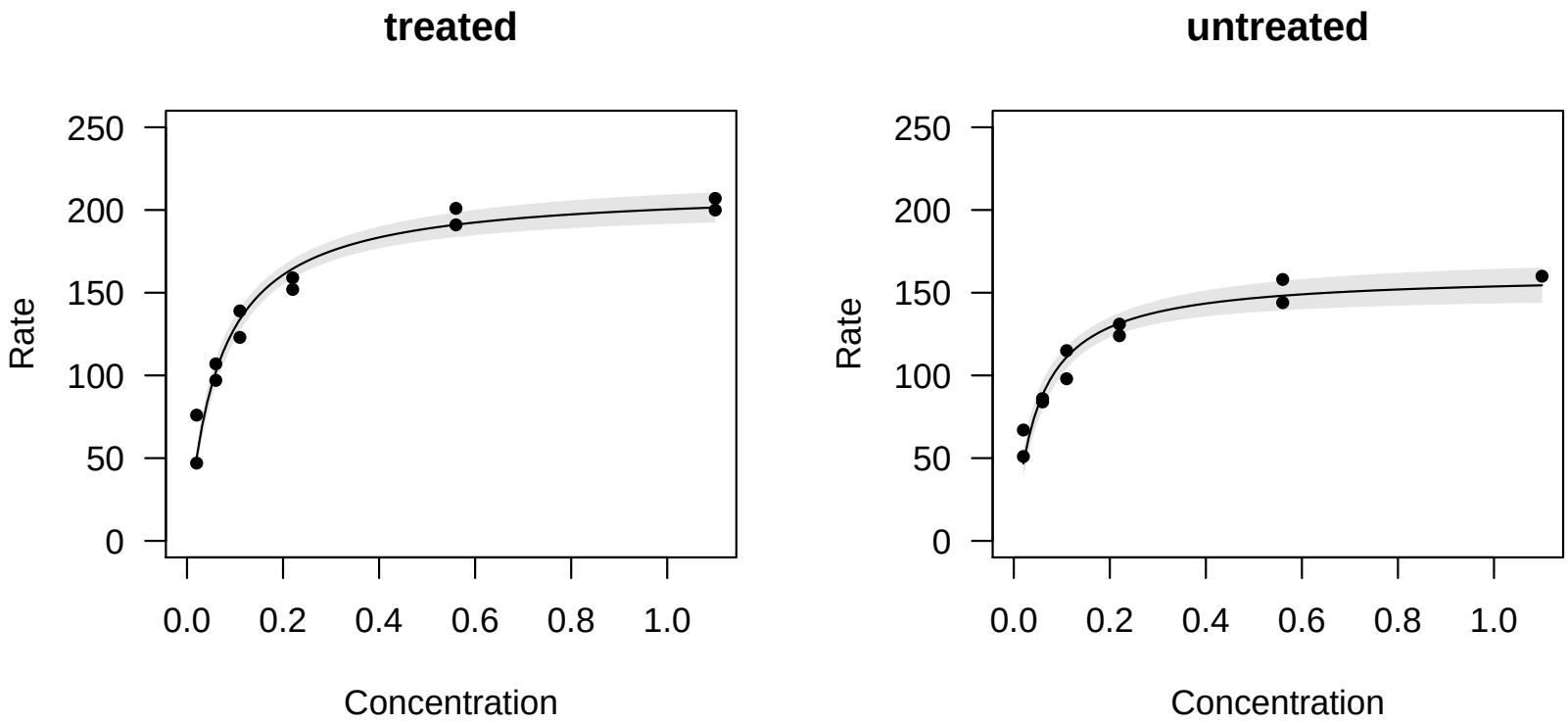
Posterior: Efficiency

A commonly used measure of enzyme efficiency is V_m/K, the slope at $[S] = 0$:



		90% CI	
	Mean	Lower	Upper
Treated	3295	2750	3908
Untreated	3324	2524	4297
Δ	-29	-1163	984

Fitted curves



Overview

- Of course, a very important class of extensions to linear regression are the generalized linear models (GLMs)
- Analogous to frequentist GLMs, in which there is no closed-form solution for $\widehat{\boldsymbol{\beta}}$, we do not generally have conjugate relationships in Bayesian GLMs
- Nevertheless, fitting Bayesian GLMs is straightforward with MCMC; indeed, the MCMC approach is more flexible than the iteratively reweighted least squares approach used in frequentist GLMs, and can even applied to distributions outside the exponential family

Logistic regression

- Perhaps the common specific model in the GLM family is logistic regression:

$$
\begin{aligned}
Y_i|\theta_i &\sim \text{Binom}(n_i, \theta_i) \\
g(\theta_i) &= \eta_i \\
\eta_i &= \mathbf{x}_i^T \boldsymbol{\beta}
\end{aligned}
$$

- In logistic regression, we use the "canonical" link:

$$
\begin{aligned}
\eta_i &= \log\left(\frac{\theta_i}{1-\theta_i}\right) \\
\theta_i &= \frac{e^{\eta_i}}{1+e^{\eta_i}};
\end{aligned}
$$

this function is known as the "logit" of θ_i

Beetle data

- A classic data set used to illustrate logistic regression is the beetle mortality data from Bliss (1935)
- The data (`beetles.txt`) consists of 8 experiments in which beetles were exposed to various concentrations of carbon disulphide (a fumigant) for five hours
- The response variable is the number of beetles killed by the fumigant, with dose being the explanatory variable

Beetle data: summary

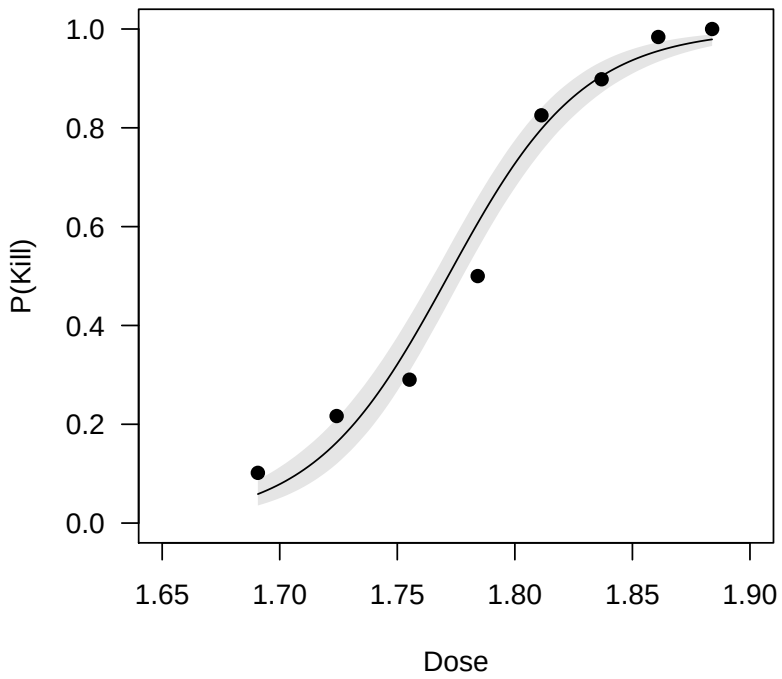
Dose	n	Killed
1.69	59	6
1.72	60	13
1.76	62	18
1.78	56	28
1.81	63	52
1.84	59	53
1.86	62	61
1.88	60	60

Implementation

The logistic regression model is easily implemented in BUGS:

```
## Likelihood
for (i in 1:N) {
  y[i] ~ dbinom(theta[i], n[i])
  logit(theta[i]) <- beta[1] + beta[2]*Dose[i]
}
## Prior
for (j in 1:2) {
  beta[j] ~ dnorm(0, 0.0001)
}
```

Posterior



Posterior odds ratio and 95% CI for a difference of 0.05 units:

Mean	Lower	Upper
5.8	4.5	7.4

Lethal dose quantiles

- A common quantity of interest in a study such as this one is the appropriate dose to kill a certain percent of beetles, typically abbreviated LD_{50} for the median lethal dose
- For a specified percent, this is merely a function of the parameters and thus straightforward to sample from:

$$LD_\pi = \beta_1^{-1}\left\{\log\left(\frac{\pi}{1-\pi}\right) - \beta_0\right\}$$

- For this experiment, we have $LD_{50} = 1.77(1.765, 1.778)$ and $LD_{99} = 1.90(1.89, 1.93)$

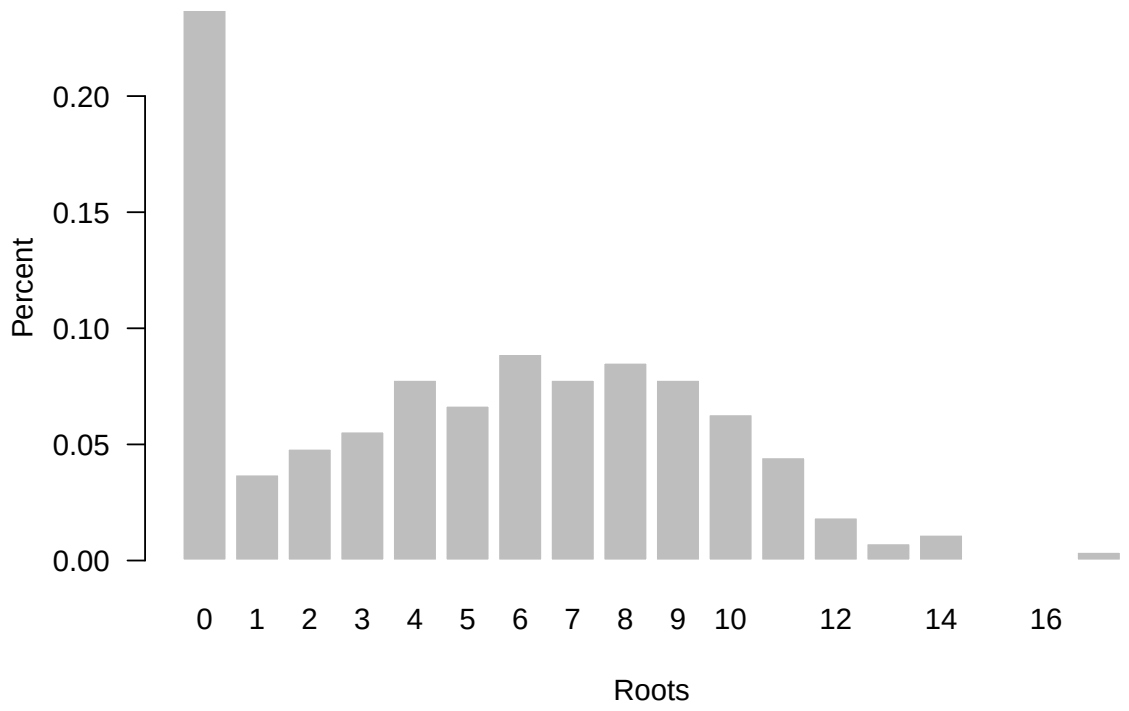
Introduction

- One of the strengths of BUGS and MCMC approaches is the ease with which models can be extended in order to account for the complexities of real data
- We will look at one such example here, in which we extend a Poisson regression model with a mixture distribution to account for an excess of zeros in the data

Rooting dataset

- Our data here come from a horticultural experiment in which apple shoots of the "Trajan" cultivar were grown under various experimental conditions
- The outcome variable is the number of roots produced by the plant
- Two possible explanatory variables are `Photoperiod`, the length of daily exposure to light (in hours) and `Dose`, the soil concentration of a plant growth ctyokinin called BAP

Roots data



Mixture distribution

- The covariates are able to explain some of the mass at zero, but it is clear that there are simply far more zeros in the data set than the Poisson distribution can account for
- One possibility, then, is to assume that Y follows a mixture distribution:

$$Y|\mu,\pi \sim \begin{cases} \text{Pois}(\mu) & \text{with probability } \pi \\ 0 & \text{with probability } 1\text{-}\pi \end{cases}$$

- This sort of model is known as a *zero-inflated Poisson*, or ZIP model

ZIP model

- In the presence of covariates, we would need to include a model for how μ and π depend on the explanatory variables, the most natural model being

$$\log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} \qquad \log\left(\frac{\boldsymbol{\pi}}{1-\boldsymbol{\pi}}\right) = \mathbf{Z}\boldsymbol{\gamma}$$

- This model allows for different effects and even different covariates to be involved with each aspect of the mixture; a simpler and more stable, albeit less flexible, model is to assume

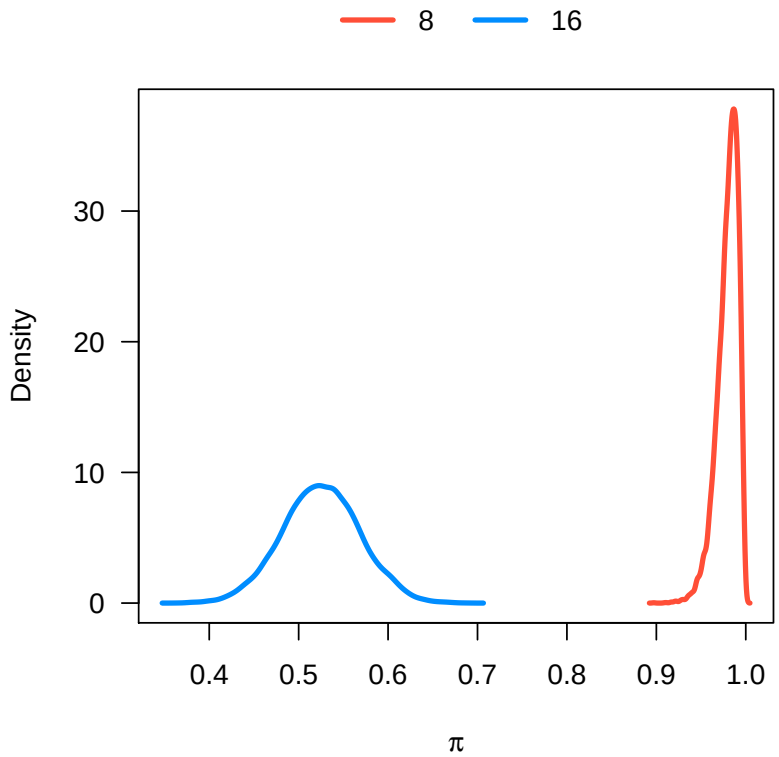
$$\log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} \qquad \log\left(\frac{\boldsymbol{\pi}}{1-\boldsymbol{\pi}}\right) = \tau\mathbf{X}\boldsymbol{\beta}$$

BUGS implementation

Here is an implementation in BUGS taking the first approach, and leaving out Dose for the sake of simplicity

```
## Likelihood
for (i in 1:n) {
  Roots[i] ~ dpois(m[i])
  m[i] <- group[i] * mu[Photoperiod[i]]
  group[i] ~ dbern(pi[Photoperiod[i]])
}
## Prior
for (j in 1:2) {
  mu[j] ~ dgamma(0.5, 0.0001)
  pi[j] ~ dunif(0, 1)
}
```

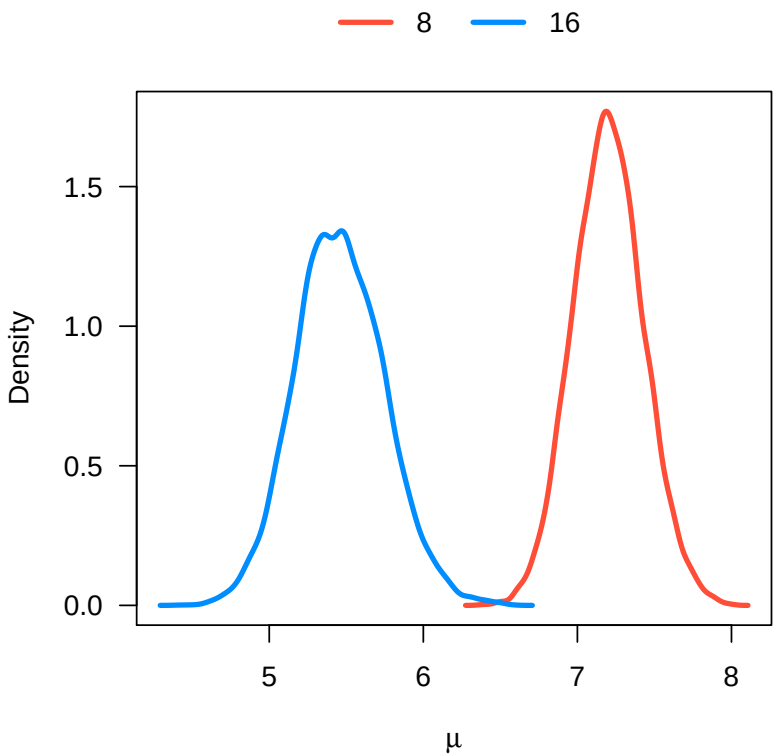
Posterior: π



Posterior means, 90% intervals

	Mean	Lower	Upper
8 hr	0.980	0.957	0.995
16 hr	0.525	0.453	0.598

Posterior: μ



Posterior means, 90% intervals

	Mean	Lower	Upper
8 hr	7.2	6.8	7.6
16 hr	5.5	5.0	5.9
Δ	1.7	1.1	2.3

Note that the actual difference in means in the two groups is much larger than the difference in means for the Poisson portion of the mixture

Distribution