

# Generalized Linear Models

STAT 489-01: Bayesian Methods of Data Analysis \*

Spring Semester 2017

## Contents

<b>1 Multi-Variable Linear Models</b>	<b>1</b>
<b>2 Generalized Linear Models</b>	<b>4</b>
2.1 Illustration: Logistic Regression . . . . .	5
2.2 Multiple Logistic Regression . . . . .	9

**Tuesday 25 April 2017**

– Refer to Chapter 14 Gelman or Chapter 5 of McElreath or Chapter 15 of Kruschke

## 1 Multi-Variable Linear Models

Recall the simple linear regression problem where each observed variable  $y_i$  is associated with one predictor variable  $x_i$  and the sampling distribution is assumed to be

$$y_i \sim N(\mu_i, \sigma^2) \quad (1.1a)$$

$$\mu_i = \alpha + \beta(x_i - \bar{x}) \quad (1.1b)$$

and we perform inference on the parameters  $\alpha$ ,  $\beta$  and  $\sigma$ . In the next two weeks we'll generalize this model to one in which

\*Copyright 2017, John T. Whelan, and all that

the errors need not be normal. But for the time being let's generalize slightly to the common situation where there is more than one predictor variable driving the observations. If we have  $k$  predictor variables, the model becomes

$$y_i \sim N(\mu_i, \sigma^2) \quad (1.2a)$$

$$\mu_i = \alpha + \sum_{j=1}^k \beta_j(x_{ij} - \bar{x}_j) = \alpha + (\mathbf{x}_i - \bar{\mathbf{x}})^T \boldsymbol{\beta} \quad (1.2b)$$

where  $\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$ , and we've defined column vectors  $\mathbf{x}_i$ ,  $\bar{\mathbf{x}}$  and  $\boldsymbol{\beta}$ . Most of the previous calculations go through, with the qualification that some quantities are now vectors or matrices. For example, we have the scalar

$$S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (1.3)$$

the vector

$$\mathbf{S}_{xy} = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(y_i - \bar{y}) \quad (1.4)$$

and the  $k \times k$  matrix

$$\mathbf{S}_{xx} = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (1.5)$$

Completing the square in the likelihood

$$p(\{y_i\}|\{\mathbf{x}_i\}, \alpha, \boldsymbol{\beta}, \sigma, M_1, I) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu_i)^2\right) \quad (1.6)$$

gives

$$\begin{aligned} \sum_{i=1}^n (y_i - \mu_i)^2 &= S_{yy} - 2\boldsymbol{\beta}^T \mathbf{S}_{xy} + \boldsymbol{\beta}^T \mathbf{S}_{xx} \boldsymbol{\beta} + n(\bar{y} - \alpha)^2 \\ &= S_{yy} - \mathbf{S}_{xy}^T \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy} + n(\bar{y} - \alpha)^2 \\ &\quad + (\boldsymbol{\beta} - \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy})^T \mathbf{S}_{xx} (\boldsymbol{\beta} - \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy}) \end{aligned} \quad (1.7)$$

It's convenient to consider the conditional posterior for  $\alpha$  and  $\boldsymbol{\beta}$ , given  $\sigma$ , and then the marginal posterior for  $\sigma^{-2}$ . We assume as before that the prior is uniform in the regression coefficients  $\alpha$  and  $\boldsymbol{\beta}$ , and non-informative in the scale parameter  $\sigma$ . We then have the conditional posterior

$$\begin{aligned} p(\alpha, \boldsymbol{\beta}|\{y_i\}, \{\mathbf{x}_i\}, \sigma, M_1, I) &\propto \exp\left(-\frac{n}{2\sigma^2}(\alpha - \bar{y})^2\right) \\ &\times \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy})^T \mathbf{S}_{xx} (\boldsymbol{\beta} - \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy})\right) \end{aligned} \quad (1.8)$$

Which is a Gaussian in  $\alpha \sim N(\bar{y}, \sigma^2/n)$  and a multivariate Gaussian in  $\boldsymbol{\beta}$  with mean  $\mathbf{S}_{xx}^{-1} \mathbf{S}_{xy}$  and variance-covariance matrix  $\sigma^2 \mathbf{S}_{xx}^{-1}$ . (If we marginalized over  $\sigma$  rather than conditioning on it we'd end up with a multivariate  $t$  distribution.

On the other hand, if we marginalize over  $\alpha$  and  $\boldsymbol{\beta}$ , we get

the posterior for  $\sigma^{-2}$ :

$$\begin{aligned} p(\sigma^{-2}|\{y_i\}, \{\mathbf{x}_i\}, M_1, I) &\propto (\sigma^{-2})^{\frac{n}{2}-1} \exp\left(\frac{\sigma^{-2}}{2}(S_{yy} - \mathbf{S}_{xy}^T \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy})\right) \\ &\quad \times \int_{-\infty}^{\infty} \exp\left(\frac{n}{2\sigma^2}(\alpha - \bar{y})^2\right) d\alpha \\ &\int \exp\left(\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy})^T \mathbf{S}_{xx} (\boldsymbol{\beta} - \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy})\right) d^k \boldsymbol{\beta} \end{aligned} \quad (1.9)$$

Now, all we care about for the marginalization integrals is how many factors of  $\sigma$  they introduce; it's not hard to see that the  $\alpha$  integral is  $\sigma$  times a constant, and each of the  $k$   $\boldsymbol{\beta}$  integrals introduces another factor of  $\sigma$  (since the result is  $\sqrt{\det(2\pi\sigma^2 \mathbf{S}_{xx}^{-1})} \propto \sigma^k$ , so the marginal posterior is

$$\begin{aligned} p(\sigma^{-2}|\{y_i\}, \{\mathbf{x}_i\}, M_1, I) &\propto (\sigma^{-2})^{\frac{n-k-1}{2}-1} \exp\left(\frac{\sigma^{-2}}{2}(S_{yy} - \mathbf{S}_{xy}^T \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy})\right) \end{aligned} \quad (1.10)$$

which says that

$$\sigma^{-2}(S_{yy} - \mathbf{S}_{xy}^T \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy}) \sim \chi^2(n - k - 1) \quad (1.11)$$

This means it's appropriate to define the estimator of  $\sigma^2$  as

$$s^2 = \frac{1}{n - k - 1} (S_{yy} - \mathbf{S}_{xy}^T \mathbf{S}_{xx}^{-1} \mathbf{S}_{xy}) \quad (1.12)$$

Note that when doing computations, it's convenient to write  $\{y_i\}$  as a column vector ( $n \times 1$  matrix)  $\mathbf{y}$ . The tricky thing though is what to call the matrix object corresponding to  $\{\mathbf{x}_i\} \equiv \{x_{ij}\}$ . We'll call it a  $n \times k$  matrix  $\mathbf{X}$ . Then for example we can combine

$\{\mu_i\}$  into  $\boldsymbol{\mu} = \alpha \mathbf{e}_{n \times 1} + \mathbf{X}\boldsymbol{\beta}$  where  $\mathbf{e}_{n \times 1}$  is a column vector of all 1s. Then we can construct for example

$$\mathbf{S}_{xx} = (\mathbf{X}^T - \bar{\mathbf{x}}^T \mathbf{e}_{1 \times n})(\mathbf{X} - \mathbf{e}_{n \times 1} \bar{\mathbf{x}}) \quad (1.13a)$$

To illustrate some of these calculations, consider the Howell1 dataset, but now limit attention to children, so that we expect weight to be correlated with age as well as height:

```
> library(rethinking)
> data(Howell1)
> y = Howell1[Howell1$age < 18, 'weight']
> ybar = mean(y)
> n = length(y)
> # x1 = Howell1[Howell1$age < 18, 'height']
> # x2 = Howell1[Howell1$age < 18, 'age']
> # x3 = Howell1[Howell1$age < 18, 'male']
> X = Howell1[Howell1$age < 18, c('height', 'age', 'male')]
> xbar = colMeans(X)
> k = length(xbar)
> Sxx = t(X - xbar) %*% as.matrix(X - xbar)
> Sxx
      height      age      male
height 1578264.93 111863.7 -16223.93
age      111863.70 670768.8 701807.50
male     -16223.93 701807.5 748472.62
> Sxxinv = solve(Sxx)
> Sxxinv
      height      age      male
height 3.245861e-06 -3.243479e-05 3.048293e-05
age     -3.243479e-05 4.027451e-04 -3.783382e-04
male     3.048293e-05 -3.783382e-04 3.567467e-04
> Sxy = t(X - xbar) %*% (y - ybar)
> Syy = sum((y - ybar)^2)
> s2 = (Syy - t(Sxy) %*% Sxxinv %*% Sxy) / (n - k - 1)
```

```
> s2 = s2[1,1]
> s = sqrt(s2)
> s
[1] 4.684493
> betahat = Sxxinv %*% Sxy
> s2beta = s2 * Sxxinv
> s2beta
      height      age      male
height 7.122871e-05 -0.0007117644 0.0006689319
age     -7.117644e-04 0.0088380300 -0.0083024324
male     6.689319e-04 -0.0083024324 0.0078286183
```

**Thursday 27 April 2017**

```
library(rethinking)
data(Howell1)
y = Howell1[Howell1$age < 18, 'weight']
ybar = mean(y)
n = length(y)
X = Howell1[Howell1$age < 18, c('height', 'age', 'male')]
xbar = colMeans(X)
k = length(xbar)
X_xbar = t(t(X) - xbar)
s2x = colSums(X_xbar^2) / (n - 1)
zX = t(t(X_xbar) / sqrt(s2x))
zX[1:10,]
# or just zX = scale(X)
Sxx = t(zX) %*% as.matrix(zX)
Sxx
Sxxinv = solve(Sxx)
Sxxinv
Sxy = t(zX) %*% (y - ybar)
Syy = sum((y - ybar)^2)
```

```

s2 = (Syy - t(Sxy) %*% Sxxinv %*% Sxy)/(n-k-1)
s2 = s2[1,1]
s = sqrt(s2)
s
betahat = Sxxinv %*% Sxy
betahat
covbeta = s2 * Sxxinv
covbeta
varbeta = diag(covbeta)
sdbeta = sqrt(varbeta)
corrbeta = covbeta/outer(sdbeta,sdbeta)
corrbeta
cov(zX[, 'height'], zX[, 'age'])
plot(zX[, 'height'], zX[, 'age'])
cov(zX[, 'height'], zX[, 'male'])
plot(zX[, 'height'], zX[, 'male'])

```

**Tuesday 2 May 2017**

– Refer to Chapter 9 and 10 of McElreath or Chapter 16 of Gelman

## 2 Generalized Linear Models

Sometimes the linear model  $y_i \sim N(\alpha + \beta^T(\mathbf{x}_i - \bar{\mathbf{x}}), \sigma^2)$  is not likely to even be a good approximation. For instance if the data  $\{y_i\}$  represent discrete quantities like event counts, it doesn't make sense to model them with a normal distribution unless the numbers are large. Suppose for example that they represent the results of Bernoulli trials at a series of predictor variable values  $\{\mathbf{x}_i\}$ , with numbers of trials given by  $\{n_i\}$ . Then a reasonable

model is<sup>1</sup>

$$y_i \sim \text{Bin}(n_i, p_i) \quad (2.1)$$

and the parameter which must be fit is not the mean  $\mu_i$ , but the probability  $p_i$ . Since by definition the model has to give us  $0 \leq p_i \leq 1$ , we can't use a linear model for  $p_i$ . But we know a convenient transformation that changes  $p_i$  into an unbounded parameter: the log-odds-ratio

$$\lambda_i = \ln \frac{p_i}{1 - p_i} \quad (2.2)$$

This is also known as the logit function or the inverse logistic function. The logistic function is its inverse:

$$p_i = \frac{e^{\lambda_i}}{1 + e^{\lambda_i}} = (1 + e^{-\lambda_i})^{-1} \quad (2.3)$$

we can then complete the model in terms of parameters  $\alpha$  and  $\beta$  such that

$$\ln \frac{p_i}{1 - p_i} = \alpha + \beta^T(\mathbf{x}_i - \bar{\mathbf{x}}) \quad (2.4)$$

or

$$p_i = [1 + \exp(-\alpha - \beta^T(\mathbf{x}_i - \bar{\mathbf{x}}))]^{-1} \quad (2.5)$$

This is an example of a generalized linear model (GLM). We can write it as

$$y_i \sim \text{Dist}(\mu_i) \quad (2.6a)$$

$$g(\mu_i) = \alpha + \beta^T(\mathbf{x}_i - \bar{\mathbf{x}}) \quad (2.6b)$$

the inputs are the parametrized sampling distribution  $\text{Dist}(\mu_i)$  (in this case a Bernoulli distribution, if we specialize to  $n_i = 1$ ) and the *link function*  $g(\mu_i)$  (in this case the logit function).

---

<sup>1</sup>Note that we can also treat each trial separately, in which case  $n_i = 1$ , and we allow for the possibility that the same  $\mathbf{x}$  value may appear more than once on the list of  $\{\mathbf{x}_i\}$ .

Another example would be where  $y_i$  represents a number of counts with effectively no upper limit. Then we might want to define the sampling distribution as Poisson  $y \sim \text{Pois}(\mu_i)$ . Now  $\mu_i$  is a continuous parameter which can be between 0 and  $\infty$ . A convenient way to map that onto the real line would be to use a logarithmic link function

$$\ln \mu_i = \alpha + \beta^T(\mathbf{x}_i - \bar{\mathbf{x}}) \quad (2.7)$$

## 2.1 Illustration: Logistic Regression

Return to the Bernoulli trials model where

$$y_i \sim \text{Bin}(1, p_i) \quad (2.8a)$$

$$p_i = [1 + \exp(-\alpha - \beta(x_i - \bar{x}))]^{-1} = \frac{e^{\alpha + \beta(x_i - \bar{x})}}{1 + e^{\alpha + \beta(x_i - \bar{x})}} \quad (2.8b)$$

The sampling distribution is then

$$p(\{y_i\}|\{x_i\}, \alpha, \beta, I) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1-y_i} = \prod_{i=1}^n \frac{e^{[\alpha + \beta(x_i - \bar{x})]y_i}}{1 + e^{\alpha + \beta(x_i - \bar{x})}} \quad (2.9)$$

Since there are only two parameters, we could in principle just plot the likelihood or posterior. It's tempting to start with the MAP solution, but although it's simple to get the analytical form of the maximum likelihood equations, they don't have a closed form solution for  $\alpha$  and  $\beta$ . A little bit of differentiation and algebra shows them to be

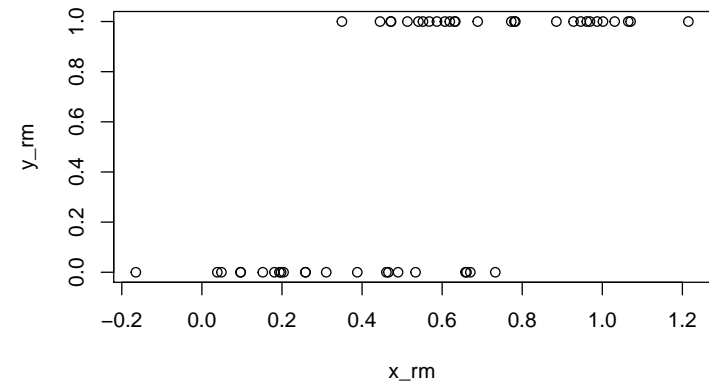
$$n\bar{y} = \sum_{i=1}^n \frac{1}{1 + e^{-\alpha - \beta(x_i - \bar{x})}} = \sum_{i=1}^n p_i \quad (2.10a)$$

$$S_{xy} = \sum_{i=1}^n \frac{x_i - \bar{x}}{1 + e^{-\alpha - \beta(x_i - \bar{x})}} = \sum_{i=1}^n (x_i - \bar{x})p_i \quad (2.10b)$$

In practice, it'll be easier just to run an MCMC to get a sample from the joint posterior.

We illustrate this using a specific example taken from Messenger et al, *Physical Review D* **92**, 023006 (2015). Fifty simulated gravitational wave signals were added to simulated detector noise, and five different pipelines performed searches and reported a detection or non-detection for each signal. These results are available at <https://dcc.ligo.org/LIGO-G1500977/public>, and I've converted them slightly for our purposes in [http://ccrg.rit.edu/~whelan/courses/2017\\_1sp\\_STAT\\_489/data/](http://ccrg.rit.edu/~whelan/courses/2017_1sp_STAT_489/data/). In particular, I've taken the signal amplitude  $h_0$  and converted it to a logarithmic scale via  $x = \log_{10}(10^{25}h_0)$ . The idea is to take the 50 found/missed results and use them to interpolate an efficiency curve as a function of signal strength. Here's what the found/missed results look like:

```
> data = read.table('found_rm.dat')
> x_rm = data[,1]
> y_rm = data[,2]
> n_rm = length(y_rm)
> xbar_rm = mean(x_rm)
> plot(x_rm, y_rm)
```



We can use Stan to sample from the posterior. You might think we'd have to define each probability  $p_i$  as a separate parameter, but it turns out that Stan has a `bernoulli_logit` distribution defined which takes as its parameter the log odds ratio:

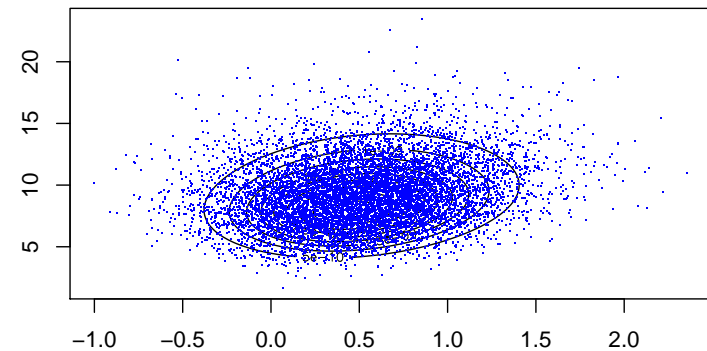
```
> library(rstan)
> rstan_options(auto_write = TRUE)
> options(mc.cores = parallel::detectCores())
> modelString = "
+ data {
+   int<lower=0> n;
+   int<lower=0,upper=1> y[n];
+   vector[n] x;
+   real xbar;
+ }
+ parameters {
+   real alpha;
+   real beta;
+ }
+ model {
+   y ~ bernoulli_logit(alpha + beta*(x-xbar));
+ }"
> DS0 = stan_model(model_code=modelString)
```

In principle `xbar` and `n` should be calculable from the data, but we pass them in because we already know how to calculate them in R. Now run the chains:

```
> rmFit = sampling(object=DS0,
+                 data=list(y=y_rm,x=x_rm,
+                           n=n_rm,xbar=xbar_rm),
+                 chains=3, warmup=500, iter=4000,
+                 seed=20170502)
> rm_alpha = extract(rmFit,'alpha')$alpha
> rm_beta = extract(rmFit,'beta')$beta
```

We want to plot our analytically-calculated posterior for comparison, so we extract the range of alphas and betas from the posterior samples. (Otherwise we could do this by trial and error.)

```
> alphamin = min(rm_alpha)
> alphamax = max(rm_alpha)
> betamin = min(rm_beta)
> betamax = max(rm_beta)
> Ngrid = 101
> ones = rep(1,Ngrid)
> alphas = seq(from=alphamin,to=alphamax,length.out=Ngrid)
> betas = seq(from=betamin,to=betamax,length.out=Ngrid)
> alphagrid = outer(alphas,ones,'*')
> betagrid = outer(ones,betas,'*')
> loglike = matrix(rep(0,Ngrid*Ngrid),ncol=Ngrid)
> for (i in 1:n_rm){
+   lami = alphagrid + betagrid * (x_rm[i]-xbar_rm)
+   loglike = loglike + y_rm[i] * lami
+   loglike = loglike - log(1+exp(lami))
+ }
> likelihood = exp(loglike)
> contour(alphas,betas,likelihood)
> points(rm_alpha,rm_beta,pch='.',col='blue')
```



We see that the sample drawn with Stan does agree with the analytic posterior for these data.

Of course,  $\alpha$  and  $\beta$  are not so easy to interpret; what we really want to learn about is the function

$$p(x; \alpha, \beta) = \text{logistic}(\alpha + \beta[x - \bar{x}]) \quad (2.11)$$

where as usual

$$\text{logistic}(\lambda) = \frac{e^\lambda}{1 + e^\lambda} = (1 + e^{-\lambda})^{-1} \quad (2.12)$$

For instance, an interesting quantity is  $x_{50}(\alpha, \beta)$ , the  $x$  value at which the probability drops to one-half:

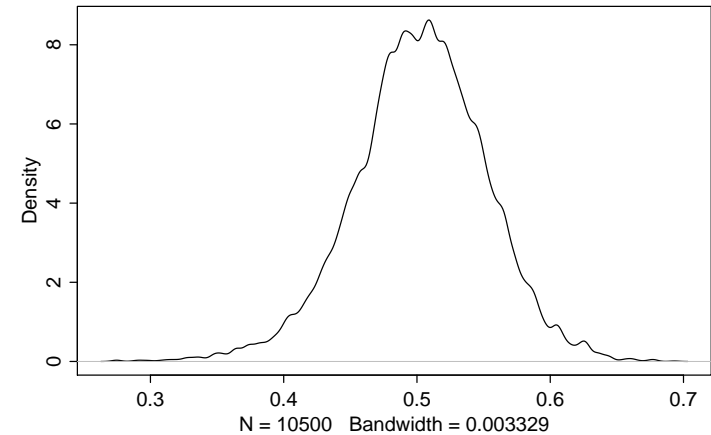
$$0.5 = p(x_{50}; \alpha, \beta) = \text{logistic}(\alpha + \beta[x_{50} - \bar{x}]) \quad (2.13)$$

Since  $0.5 = \text{logistic}(0)$ , we can solve for

$$x_{50} = \bar{x} - \frac{\alpha}{\beta} \quad (2.14)$$

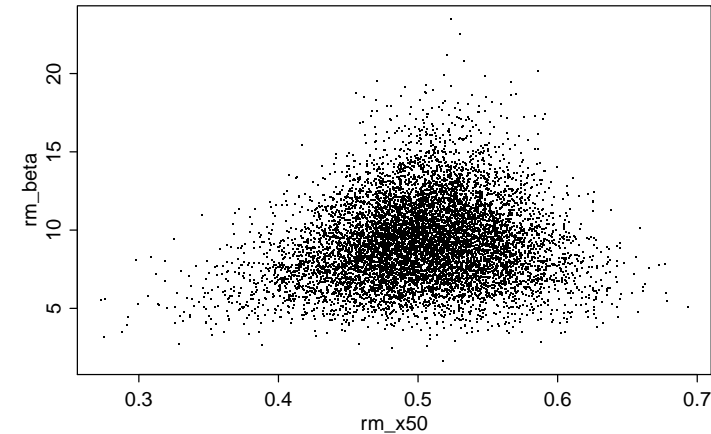
Since we have a sample from the multidimensional posterior, there's no difficulty in converting the  $\alpha, \beta$  sample into an  $x_{50}$  sample:

```
> rm_x50 = xbar_rm - rm_alpha/rm_beta
> library(rethinking)
> dens(rm_x50)
```



or even an  $x_{50}, \beta$  sample:

```
> plot(rm_x50, rm_beta, type='p', pch='.')
```



This  $x_{50}$  is arguably a better parameter than  $\alpha$ . In terms of  $x_{50}$  and  $\beta$ , the model is

$$p(x; x_{50}, \beta) = \text{logistic}(\beta[x - x_{50}]) \quad (2.15)$$

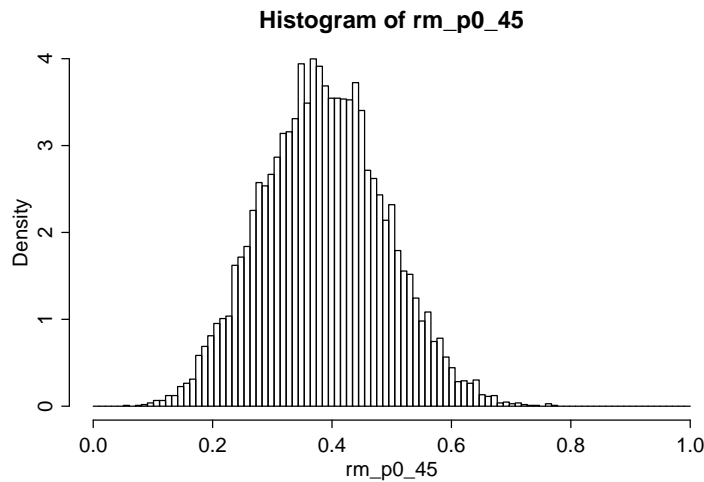
$x_{50}$  determines where the efficiency curve turns over, and  $\beta$  determines how steeply it does so. That also makes sense in light

of the posterior on these quantities; the higher  $\beta$ , the steeper the curve is, and the better-constrained  $x_{50}$  is.

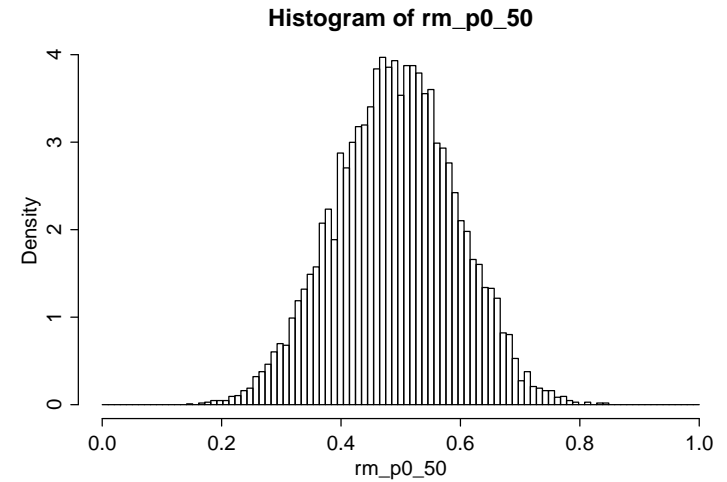
Note that changing these parameters could cause us to make different choices about the prior on the parameters. Rather than  $p(\alpha, \beta|I) = \text{const}$  we might choose  $p(x_{50}, \beta|I') = \text{const}$  or even  $p(x_{50}, \beta|I') \propto \beta^{-1}$ .

We can also extract from the search the efficiency at specified  $x$  values:

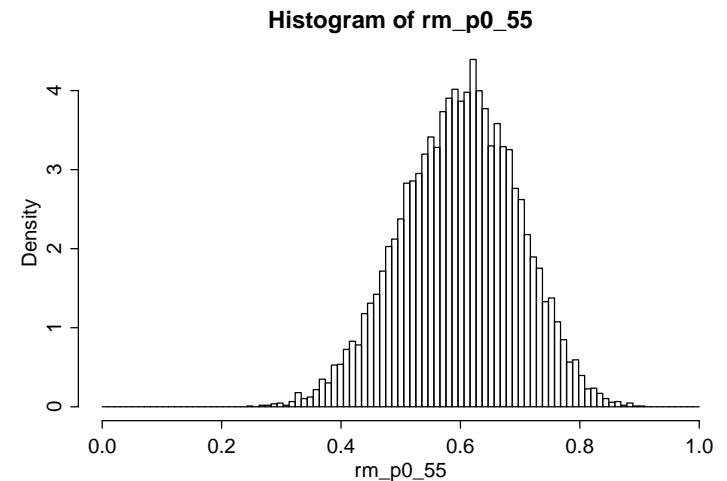
```
> rm_p0_45 = logistic(rm_beta*(0.45-rm_x50))
> hist(rm_p0_45,breaks=seq(0,1,length.out=100),freq=FALSE)
```



```
> rm_p0_50 = logistic(rm_beta*(0.50-rm_x50))
> hist(rm_p0_50,breaks=seq(0,1,length.out=100),freq=FALSE)
```



```
> rm_p0_55 = logistic(rm_beta*(0.55-rm_x50))
> hist(rm_p0_55,breaks=seq(0,1,length.out=100),freq=FALSE)
```



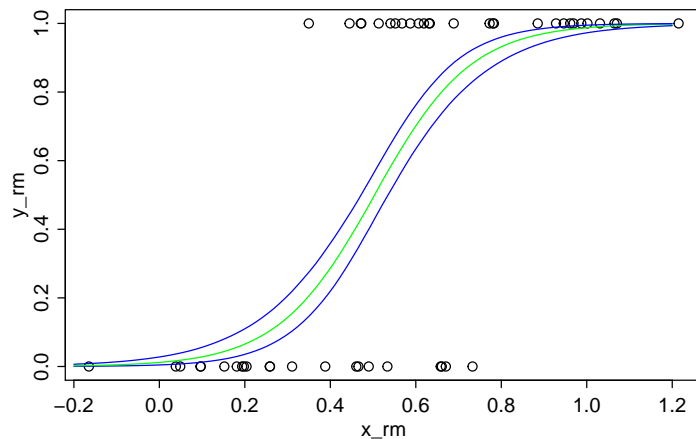
One “deliverable”, which appears in the paper and presentation cited, is to take a series of  $x$  values, calculate percentiles of the  $p(x; x_{50}, \beta)$  posterior at each  $x$ , and then plot those:



```

> xlist = seq(from=0,to=15,length.out=Nx)
> px_25 = rep(0/0,Nx)
> px_50 = rep(0/0,Nx)
> px_75 = rep(0/0,Nx)
> for (l in 1:Nx) {
+   px = logistic(rm_beta*(xlist[l]-rm_x50))
+   px_25[l] = quantile(px,.25)
+   px_50[l] = quantile(px,.50)
+   px_75[l] = quantile(px,.75)
+ }
> plot(x_rm,y_rm)
> lines(xlist,px_25,col='blue')
> lines(xlist,px_50,col='green')
> lines(xlist,px_75,col='blue')

```



Thursday 4 May 2017

– Refer to Chapter 6 of McElreath

## 2.2 Multiple Logistic Regression

```
data = read.table('found2_rm.dat')
```

```

x1_rm = data[,1]
x2_rm = data[,2]
y_rm = data[,3]
n_rm = length(y_rm)
x1bar_rm = mean(x1_rm)
x2bar_rm = mean(x2_rm)
plot(x1_rm,x2_rm)
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
modelString = "
data {
  int<lower=0> n;
  int<lower=0,upper=1> y[n];
  vector[n] x1;
  vector[n] x2;
}
parameters {
  real x50;
  real beta1;
  real beta2;
}
model {
  y ~ bernoulli_logit(beta1*(x1-x50) + beta2*x2);
}"
DSO = stan_model(model_code=modelString)
rmFit = sampling(object=DSO,
  data=list(y=y_rm,n=n_rm,
  x1=x1_rm,x2=x2_rm),
  chains=3, warmup=500, iter=4000,
  seed=20170504)
rm_x50 = extract(rmFit,'x50')$x50
rm_beta1 = extract(rmFit,'beta1')$beta1

```

```

rm_beta2 = extract(rmFit,'beta2')$beta2
plot(rm_x50,rm_beta1,type='p',pch='.')
plot(rm_x50,rm_beta2,type='p',pch='.')
plot(rm_beta1,rm_beta2,type='p',pch='.')
cor(rm_beta1,rm_beta2)
effmodelString = "
data {
  int<lower=0> n;
  int<lower=0,upper=1> y[n];
  vector[n] x1;
  vector[n] xeff;
}
parameters {
  real xeff50;
  real beff;
  real gamma;
}
model {
  gamma ~ normal(0,1);
  y ~ bernoulli_logit(beff*(xeff-xeff50) + gamma*x1);
}"
effDS0 = stan_model(model_code=effmodelString)
effrmFit = sampling(object=effDS0,
  data=list(y=y_rm,n=n_rm,
  x1=x1_rm,xeff=(x1_rm+x2_rm)),
  chains=3, warmup=500, iter=4000,
  seed=201705041)
effrm_xeff50 = extract(effrmFit,'xeff50')$xeff50
effrm_beff = extract(effrmFit,'beff')$beff
effrm_gamma = extract(effrmFit,'gamma')$gamma
Nx = 100
xlist = seq(from=-0.2,to=1.2,length.out=Nx)
px_25 = rep(0/0,Nx)

```

```

px_50 = rep(0/0,Nx)
px_75 = rep(0/0,Nx)
for (l in 1:Nx) {
  px = logistic(effrm_beff*(xlist[l]-effrm_xeff50))
  px_25[l] = quantile(px,.25)
  px_50[l] = quantile(px,.50)
  px_75[l] = quantile(px,.75)
}
plot(x1_rm+x2_rm,y_rm)
lines(xlist,px_25,col='blue')
lines(xlist,px_50,col='green')
lines(xlist,px_75,col='blue')

```