Logistic Regression

Edps 590BAY

Carolyn J. Anderson

Department of Educational Psychology

ILLINOIS

Spring 2018

Overview

- Logistic regression
- MLE estimation: GMAT data
- Bayesian estimation: GMAT data
- Various computational options:
  - thinning.
  - multiple cores (faster).
  - miscellaneous
- Quasi and complete separation.
- HSB data?

As a GLM

- Random component: The response variable is **dicotomous**, coded $Y_i = 1$ or 0. The distribution of Y_i is Binomial and we are interesting in probability that $Y_i = 1$, i.e., $Pr(x_i)$.
- Systematic component: A linear predictor such as

$$b_0 + b_1 x_{1i} + \ldots + b_k x_{ki}$$

  The explanatory or predictor variables may be quantitative (continuous), qualitative (discrete or ordinal), or both (mixed).
- Link Function: The log of the odds that an event occurs, otherwise known as the logit:

$$\text{logit}(Pr(y_i = 1)) = \log\left(\frac{y_i = 1}{1 - Pr(y_i = 1)}\right)$$

Putting this all together, the logistic regression model is

$$\text{logit}(\pi(x_i)) = \log\left(\frac{\pi(x_i)}{1 - \pi(x_i)}\right) = b_0 + b_1 x_{1i} + \ldots + b_k x_{ki}$$

Some Uses of Logistic Regression

- Identify "risk" factors for certain conditions (diabetes, divorce, etc.)
- Descriptive discriminant analysis–describe differences between groups.
- Adjust for "bias" (propensity score analysis/matching).
- Predict probabilities.
- Classify individuals.
- Discrete choice.
- Social network analysis.
- Pseudo-likelihood estimation.
- . . . and many others

Interpreting logistic regression models

The model

$$\text{logit}(Pr(y_i = 1)) = b_0 + b_1 x_i$$

Equivalently,

$$Pr(y_i = 1) = \frac{\exp[b_0 + b_1 x_i]}{1 + \exp[b_0 + b_1 x_i]} = \frac{1}{1 + \exp[-(b_0 + b_1 x_i)]}$$

We'll be using GMAT data (from Johnson & Wichern text) where y is admission to medical school as a function of GMAT scores and college GPA.

- Response: $Y_i = 1$ if the student is admitted, and 0 if the student was denied to wait listed (borderline).
- Explanatory: x_{1i} = student GMAT score and x_{2i} is college GPA.

MLE Estimation

```
logreg.mle ← glm(admit ∼ gmat, data=med, family=binomial)
```

```
                Deviance Residuals:
      Min           1Q       Median       3Q          Max
  -1.5996      -0.5439      -0.2154   0.4038       2.4350
                          Coefficients:
               Estimate  Std. Error  z value      Pr(>|z|)
(Intercept) -17.397141    3.758928   -4.628  3.69e-06 ***
       gmat   0.033609    0.007386    4.551  5.35e-06 ***
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Dispersion parameter for binomial family taken to be 1
Null deviance:  111.533 on 84 degrees of freedom
Residual deviance:  60.106 on 83 degrees of freedom
AIC: 64.106
Number of Fisher Scoring iterations:  6
```

Interpretation

The odds of admission given GMAT score $= \text{gmat}_i$ are

$$\frac{Pr(admission|\text{gmat}_i)}{Pr(deny|\text{gmat}_i)} = \exp[(-17.3971 + 0.0336(\text{gmat}_i))]$$

and when $\text{gmat}_i + 1$,

$$\frac{Pr(admission|(\text{gmat}_i + 1))}{Pr(deny|(\text{gmat}_i + 1))} = \exp[(-17.3971 + 0.0336(\text{gmat}_i) + 1)]$$

The ratio is the odds ratio:

$$\frac{Pr(\text{admit}|\text{gmat}_i + 1)/Pr(\text{not admit}|\text{gmat}_i + 1)}{Pr(\text{admit}|\text{gmat}_i)/Pr(\text{not admit0}|\text{gmat}_i)}$$

$$= \frac{\exp[(-17.3971 + 0.0336(\text{gmat}_i + 1))]}{\exp[(-17.3971 + 0.0336(\text{gmat}_i))]} = \exp(0.0336)$$

Interpretation

Our odds ratio equals $\exp(0.0336) = 1.034171$

the odds admision given $gmat_i + 1 = 1.0342$ odds given $gmat_i$

In words, the odds of admission is 0.0336 times the odds of admission for one point score lower on GMAT. Note that the standard deviation of GMAT is 81.52235. Using this,

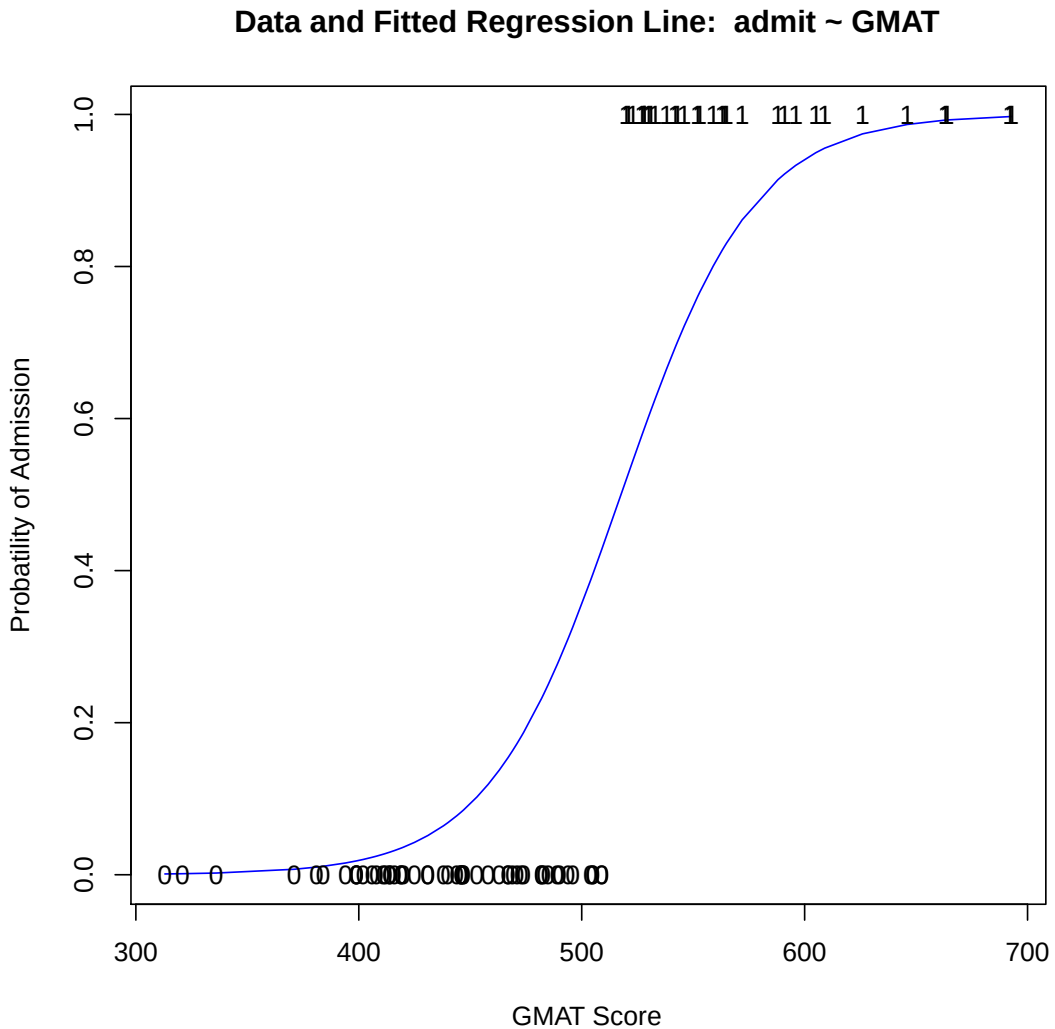
the odds admision given $gmat_i + 81.52235 = 15.47384 \exp[$ the odds given $gmat_i]$

In words, the odds of admission is 15.47 times the odds of admission for one standard deviation lower on GMAT.

Can also say "the probability of admission is higher the greater a student's GMAT score"
or "it is more likely than a student will be admitted if they have a higher GMAT score"

You should NOT say "the probability of admission is 1.03 times higher the greater a student's GMAT score"

Fitted Probability of Admission



DataList

```
dataList ← list(
                y=med$admit,
                gmat=med$gmat,
                n=length(med$admit)
)
```

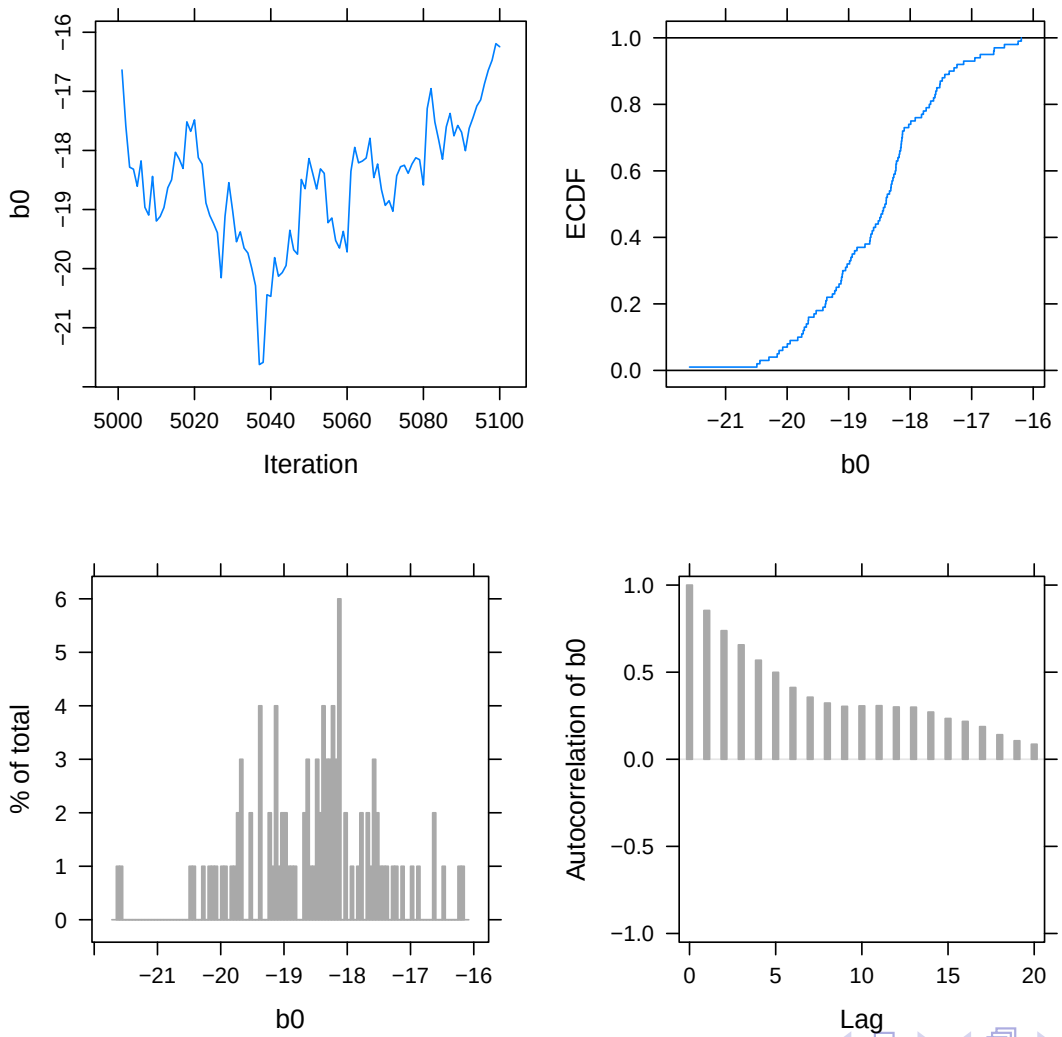
JAGS Model

```
logreg1 <- ``model {
              for (i in 1:n) {
                  y[i] ~ dbern(p[i])
                  p[i] <- 1/(1 + exp(-eta[i]))
                  eta[i] <- b0 + b1*gmat[i]
              }
              b0 ~ dnorm(0,1/1000)
              b1 ~ dnorm(0,1/1000)
           }''

writeLines(logreg1,con="logreg1.txt")
```

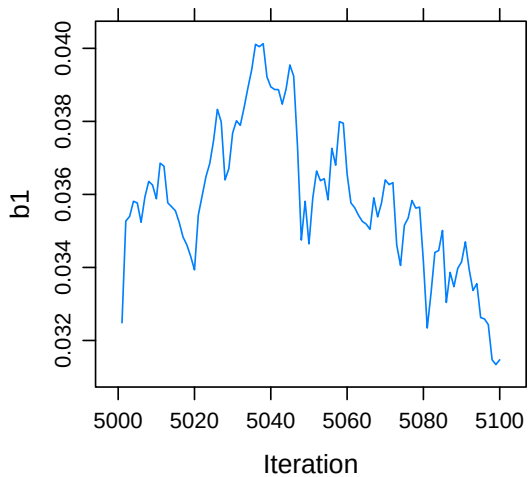
runjags to check model set up

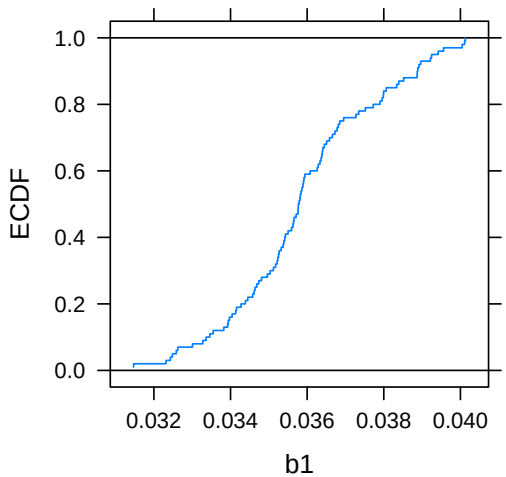
```
start1 ← list("b0"=1,"b1"=-.04)

logreg1.chk ← run.jags(
                model=logreg1,
                sample=100,
                data=dataList,
                inits=start1,
                monitor=c("b0","b1", "dic"),
                n.chains=1
                )


print(logreg1.chk)
plot(logreg1.chk)
```

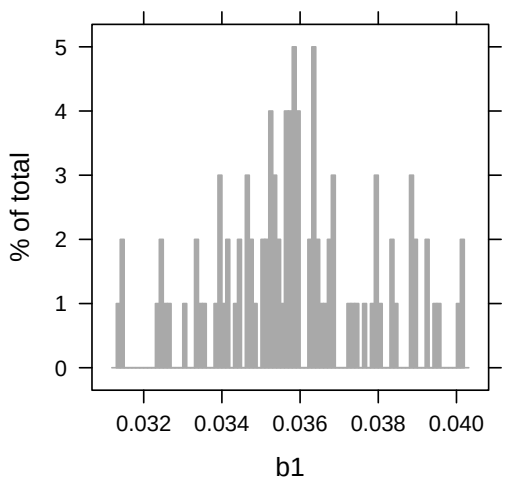
From Checking Set-Up

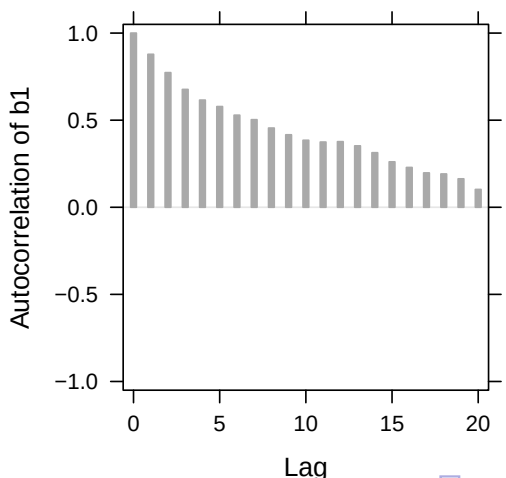


From Checking Set-Up









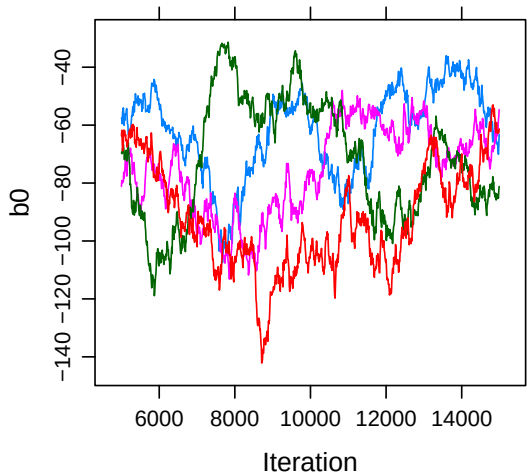
Now Run to Convergence

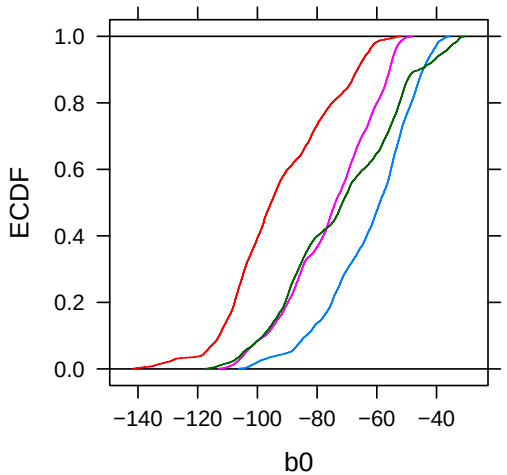
```
start ← list(
                    list("b0"=1,"b1"=-.04),
                    list("b0"=-7,"b1"=.04),
                    list("b0"=5,"b1"=-.02),
                    list("b0"=-1,"b1"=-0.2)
                    )
```

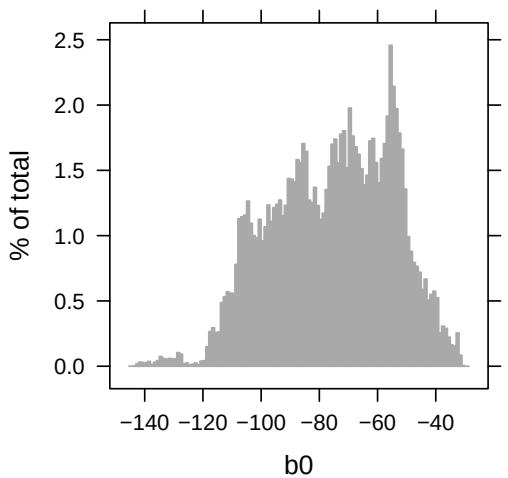
Now Run to Convergence

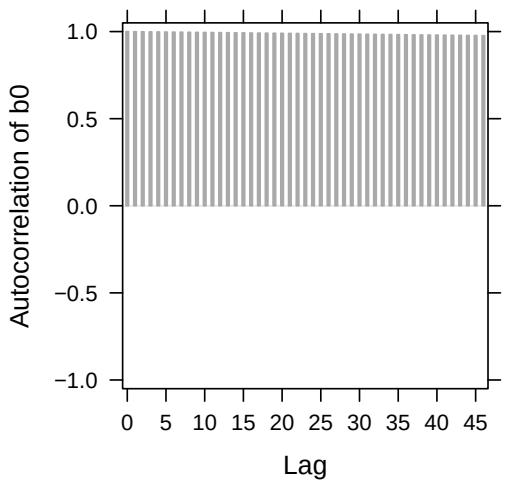
```
logreg1.jags ← run.jags(
               model=logreg1,
               data=dataList,
               inits=start,
               monitor=c("b0","b1","dic"),
               n.chains=4
               )

print(logreg1.jags)
plot(logreg1.jags)
```

Looks Pretty Bad

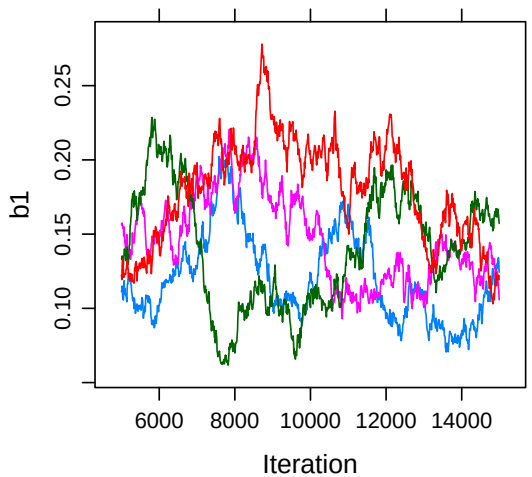


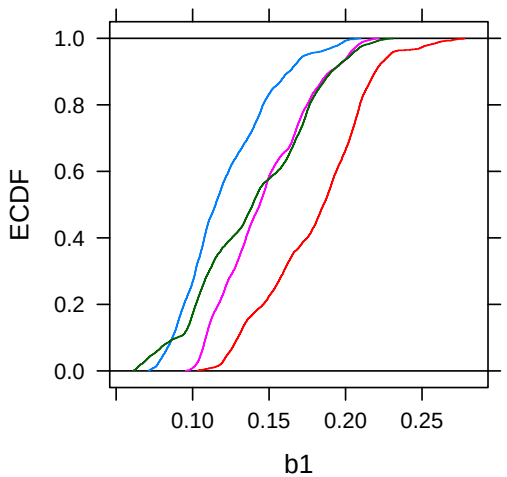


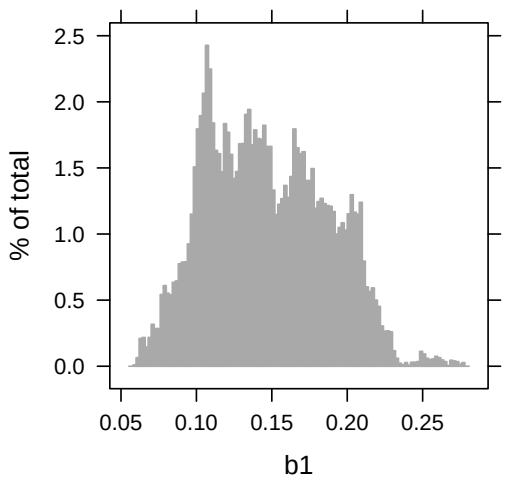


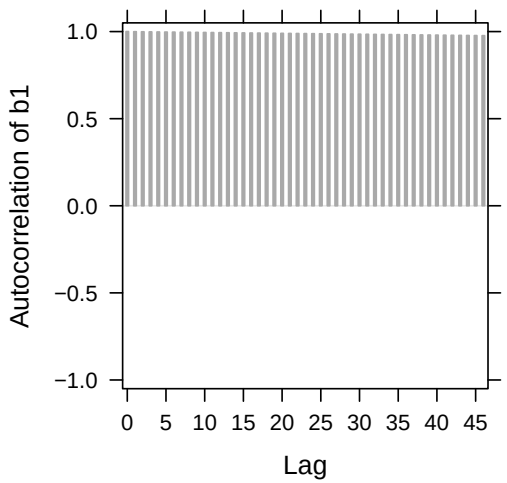


Looks Pretty Bad









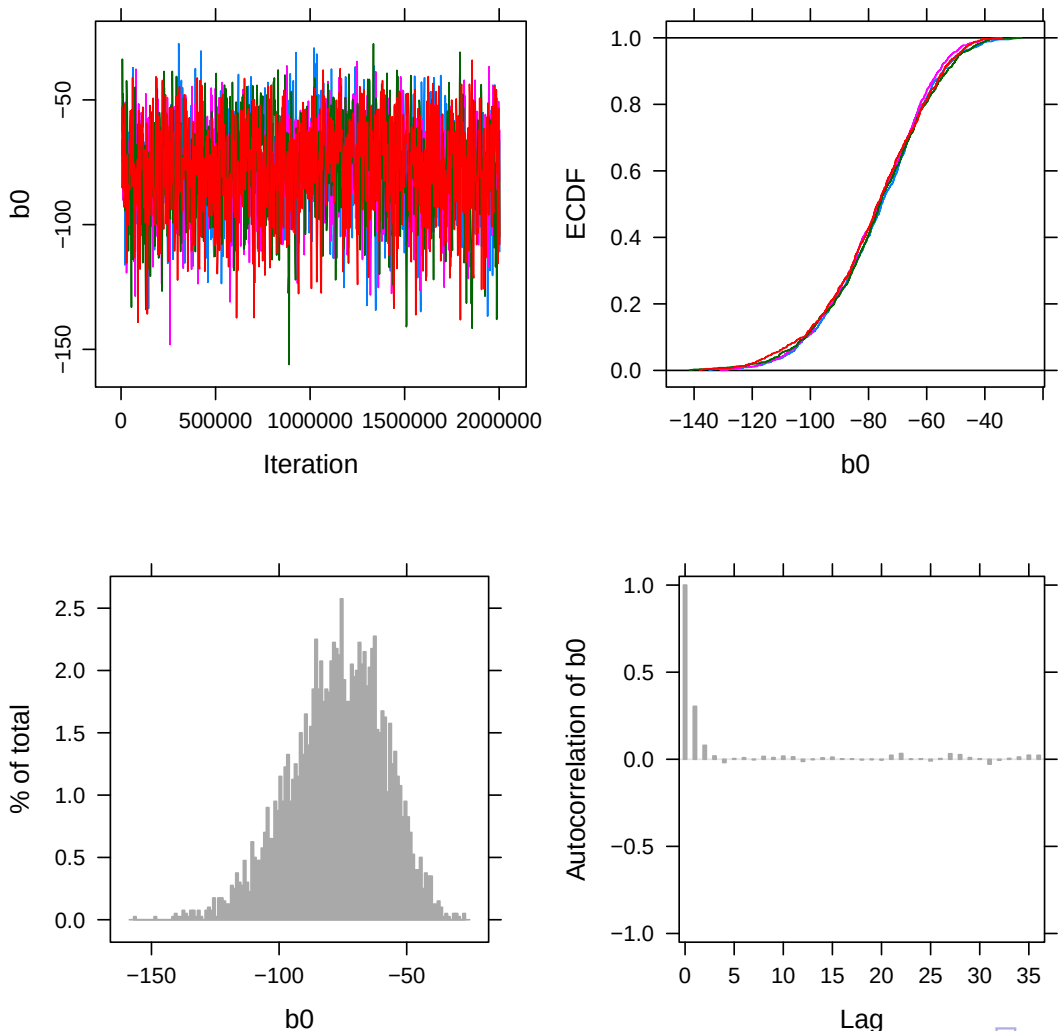
Fix & Speed up

```
# So I don't get warning message (if I want to use
parallel)

start1 ← list("b0"= 1,"b1"=-.04, "b2"=0,
       .RNG.name="base::Wichmann-Hill", .RNG.seed=23)
start2 ← list("b0"= 7,"b1"= .04, "b2"=-.5,
       .RNG.name="base::Marsaglia-Multicarry",
       .RNG.seed=97)
start3 ← list("b0"= 5,"b1"=-.02, "b2"=0.5,
       .RNG.name="base::Super-Duper", .RNG.seed=3351)
start4 ← list("b0"=-1,"b1"=-0.2, "b2"=.2,
       .RNG.name="base::Mersenne-Twister",
       .RNG.seed=7531)
```

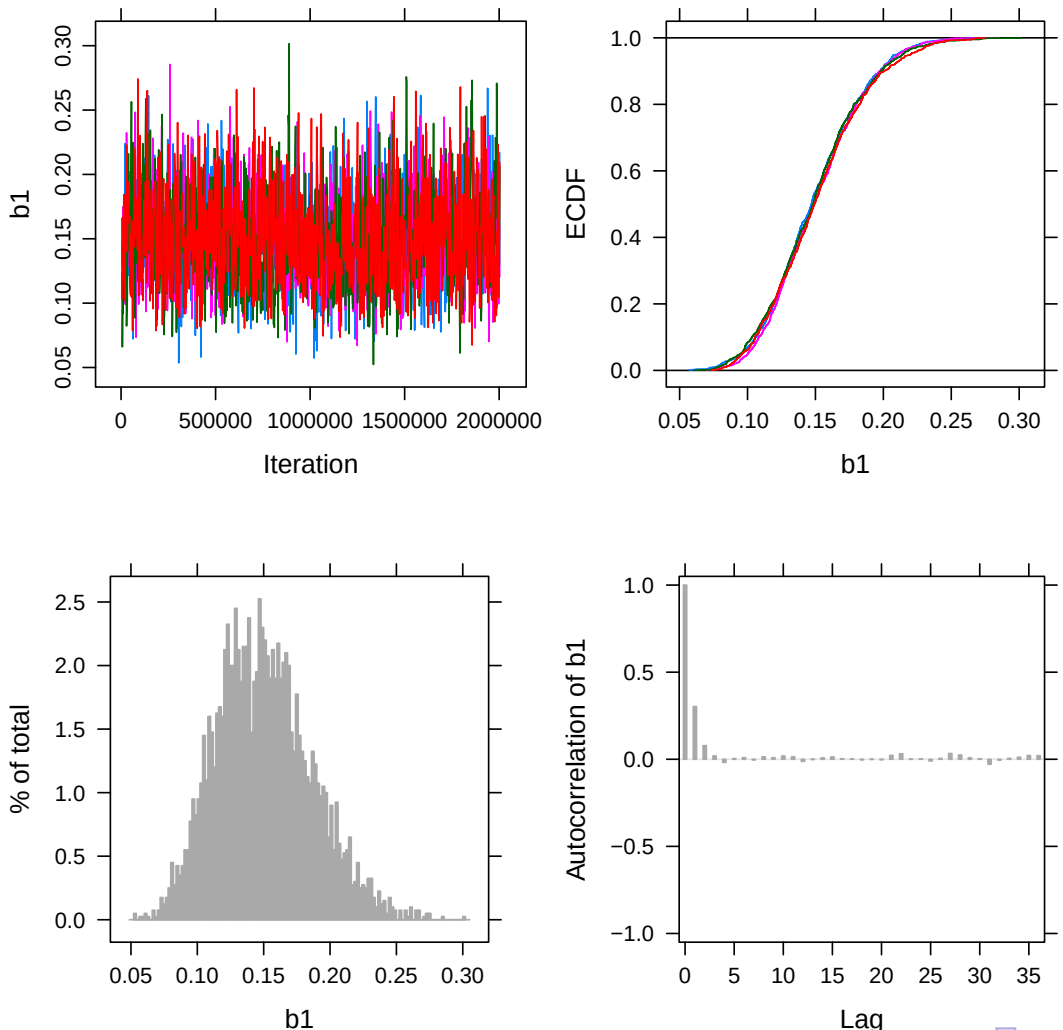
Fix (and speed up)

```
logreg1.faster <- run.jags (
        method="parallel",
        summarise=FALSE,
        plots=FALSE,
        model=logreg1,
        data=dataList,
        inits=list(start1,start2,start3,start4),
        sample=100000,
        monitor=c("b0","b1"),
        n.chains=4,
        thin=20
)
add.summary(logreg1.faster)
plot(logreg1.faster)
```

Better


b0
−50
−100
−150
0
500000
1000000
1500000
2000000
Iteration
ECDF
1.0
0.8
0.6
0.4
0.2
0.0
−140
−120
−100
−80
−60
−40
b0
% of total
2.5
2.0
1.5
1.0
0.5
0.0
−150
−100
−50
b0
Autocorrelation of b0
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35
Lag

Better


b1
Iteration
0
500000
1000000
1500000
2000000
0.05
0.10
0.15
0.20
0.25
0.30
ECDF
b1
0.0
0.2
0.4
0.6
0.8
1.0
% of total
b1
0.0
0.5
1.0
1.5
2.0
2.5
Autocorrelation of b1
Lag
−1.0
−0.5
0.0
0.5
1.0
0
5
10
15
20
25
30
35

Statistics

JAGS model summary statistics from 400000 samples (thin = 20; chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD	Mode
b0	-114.49	-76.158	-42.598	-77.383	18.763	–
b1	0.082886	0.14817	0.22246	0.15056	0.036439	–

	MCerr	MC%ofSD	SSeff	AC.200	psrf
b0	0.38809	2.1	2338	0.88931	1.0006
b1	0.00075426	2.1	2334	0.88931	1.0008

Total time taken: 6.9 minutes

. . . It took a long time because of large number of iterations

Model Evaluation for Logistic Regression

These are my favorites, which can be used with Bayesian as well as MLE: Receiver Operating Characteristic (ROC), Classification tables, and the Concordance index.

Suppose we have a simple model

$$\text{logit}(\widehat{Pr}(y_i = 1)) = \hat{b}_o + \hat{b}_1 x_i$$

Let π_o be a cut-point or cut-score and $\widehat{Pr}(y_i = 1)$ be a predicted probability of the model. The predicted response is

$$\hat{y}_i = \begin{cases} 1 & \text{if } \widehat{Pr}(y_i = 1) > \pi_o \\ 0 & \text{if } \widehat{Pr}(y_i = 1) \leq \pi_o \end{cases}$$

Classification Table:

		Predicted	
		$\hat{y}_i = 1$	$\hat{y}_i = 0$
Actual	$y = 1$	correct	incorrect (false negative)
	$y = 0$	incorrect (false positive)	correct

Classification Table

We're more interested in conditional proportions and probabilities:

		Predicted		
		$\hat{y}_i = 1$	$\hat{y}_i = 0$	
Actual	$y = 1$	n_{11}/n_{1+}	n_{12}/n_{1+}	n_{1+}
	$y = 0$	n_{21}/n_{2+}	n_{22}/n_{2+}	n_{2+}

$$
\begin{aligned}
n_{11}/n_{1+} &= \text{proportion } (\hat{y} = 1 | y = 1) = \text{“sensitivity”} \\
n_{22}/n_{2+} &= \text{proportion } (\hat{y} = 0 | y = 0) = \text{“specificity”}
\end{aligned}
$$

$$
\begin{aligned}
p(\text{correct}) &= p(\hat{y} = 1 \ \& \ y = 1) + p(\hat{y} = 0 \ \& \ y = 0) \\
&= p(\hat{y} = 1 | y = 1) p(y = 1) + (\hat{y} = 0 | y = 0) p(y = 0) \\
&= (\text{sensitivity}) p(y = 1) + (\text{specificity}) p(y = 0)
\end{aligned}
$$

GMAT Example

- Let the cut-score equal $\pi_o = .50$.
- Compare $p(\hat{y} = 1)$ and classify i as $Y = 1$ if $p(\hat{y} = 1) > \pi_o$, otherwise classify i as $Y = 0$.
- Tabulate the results

		Predicted		
		$\hat{y} = 1$	$\hat{y} = 0$	
Actual	$y = 1$	25	6	31
	$y = 0$	6	48	52

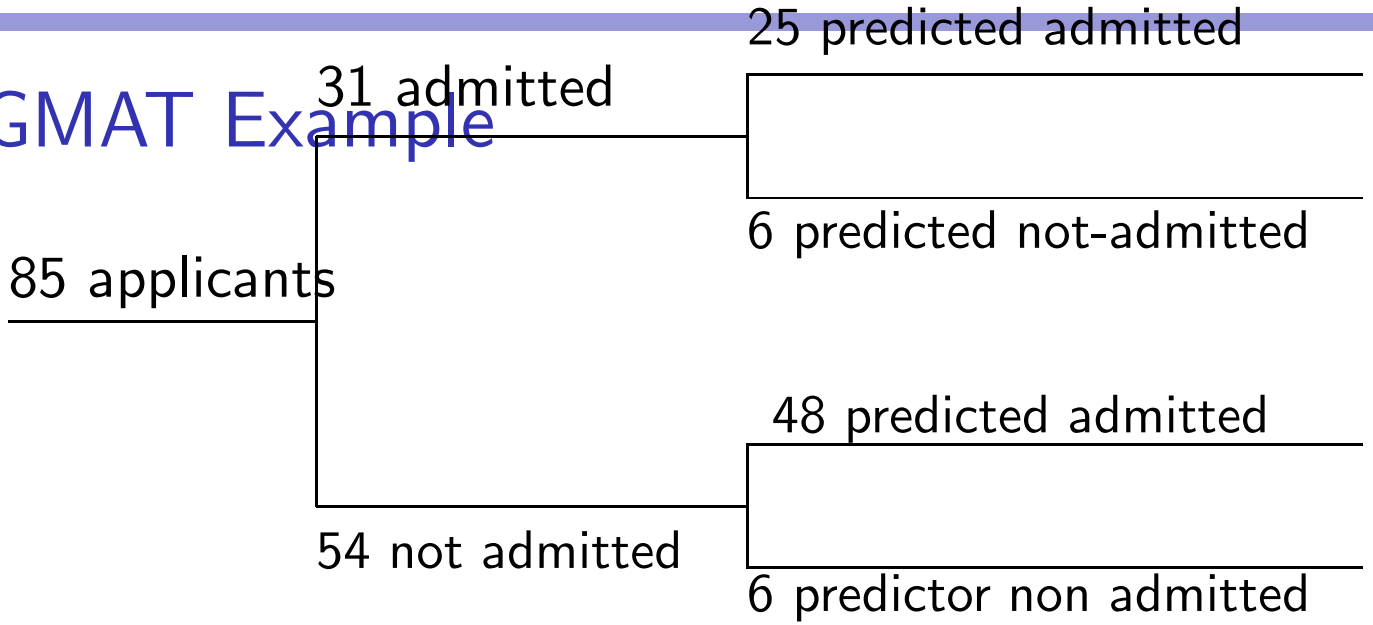
- The Conditional proportions:

$$\begin{aligned} \text{Sensitivity} &= 25/31 = .81 \\ \text{Specificity} &= 48/52 = .92 \end{aligned}$$

- The proportion correct $= (25 + 48)/85 = .86$

GMAT Example

Sensitivity= .81%



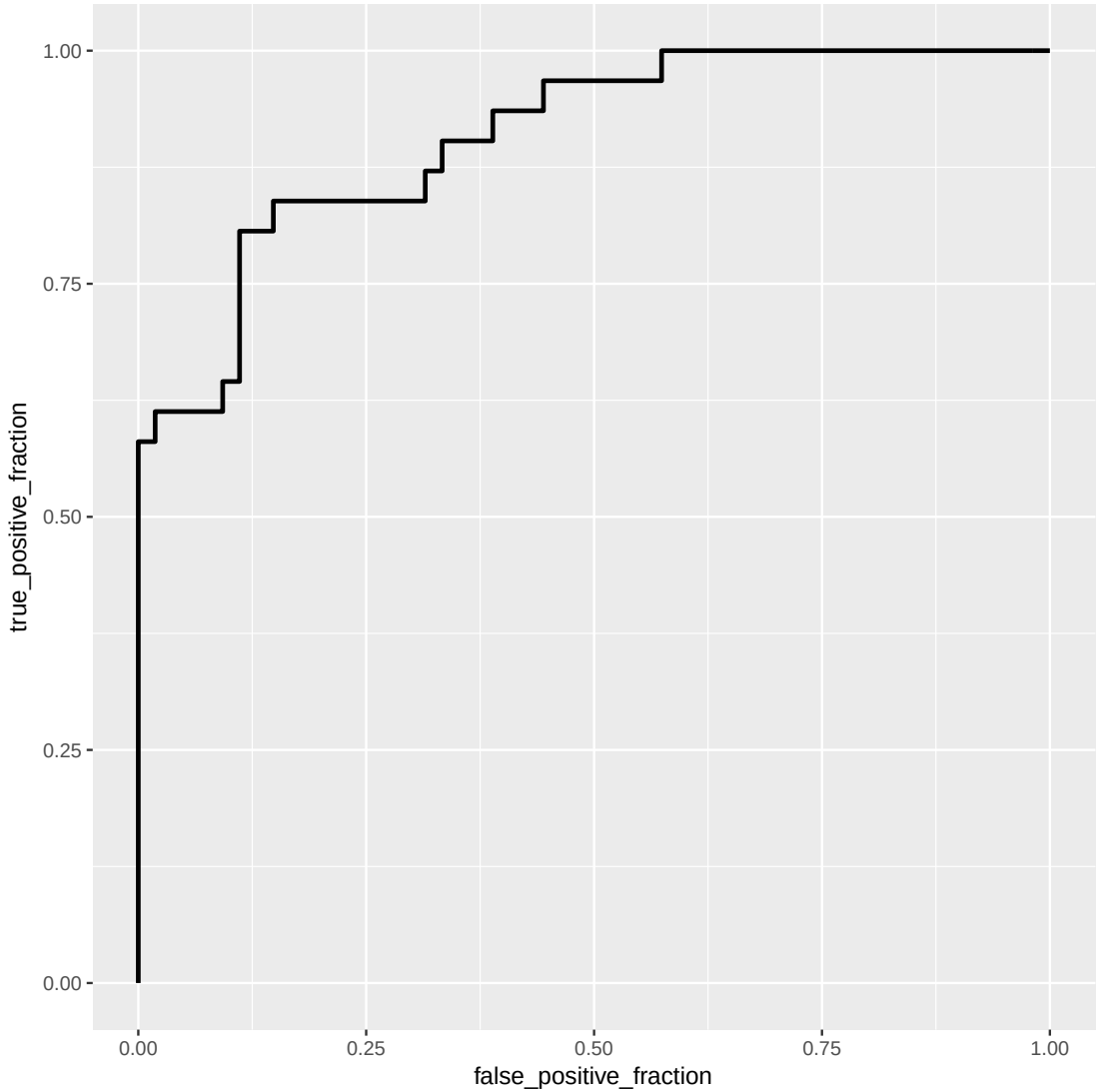
Specificity= .92%

Percent correct = .86%

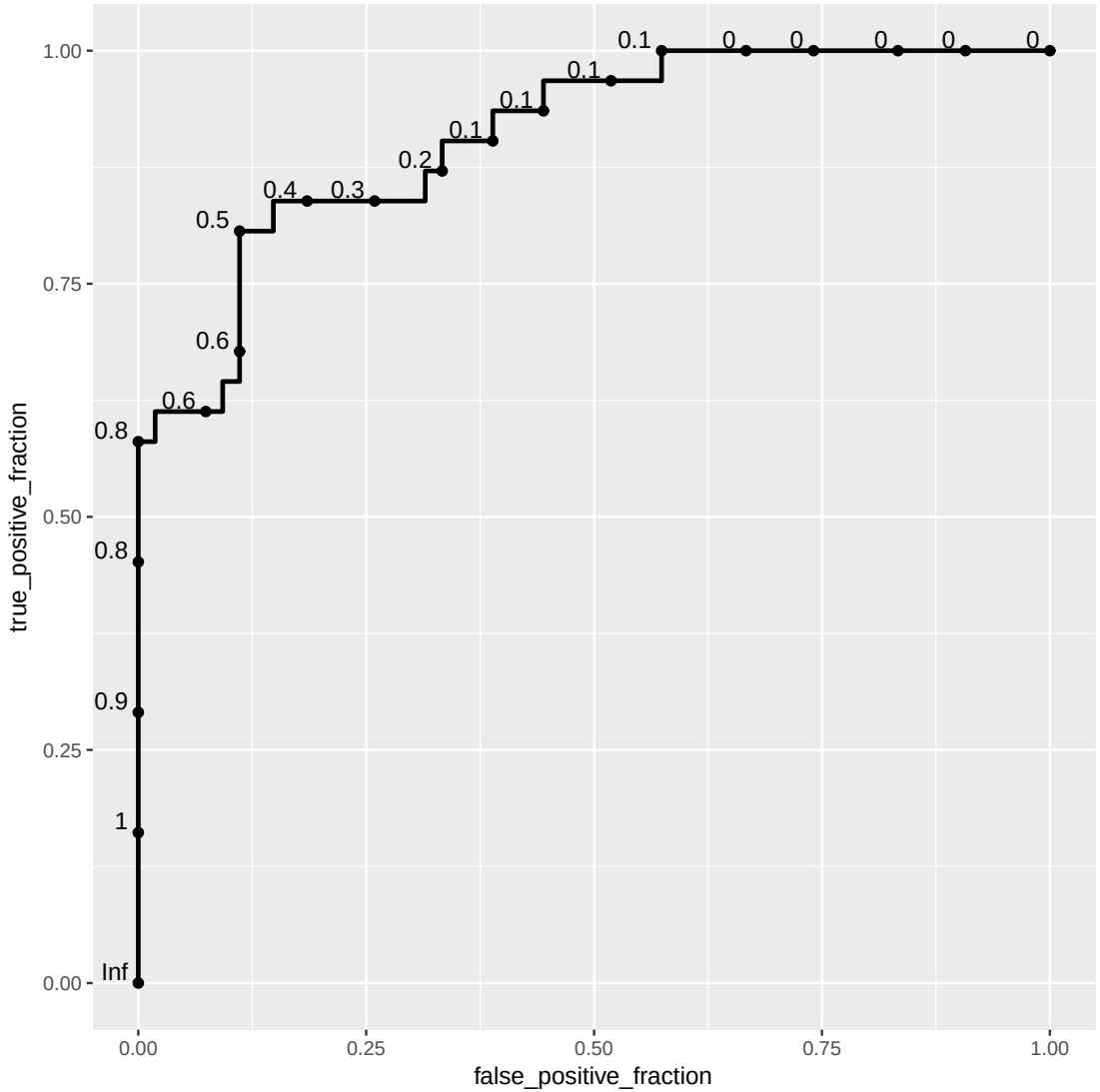
Sensitivity, Specificity & p(Correct)

- For every cut-score you will get a different result.
- Do this for lots of possible cut-scores and plot the results $\longrightarrow$ ROC curve.

ROC curve


1.00
0.75
0.50
0.25
0.00
true_positive_fraction
0.00
0.25
0.50
0.75
1.00
false_positive_fraction

ROC curve


1.00
0.75
0.50
0.25
0.00
true_positive_fraction
0.00
0.25
0.50
0.75
1.00
false_positive_fraction
0.1
0
0
0
0
0
0.1
0.1
0.1
0.2
0.4
0.3
0.5
0.6
0.6
0.8
0.8
0.9
1
Inf

Area Under ROC Curve

Concordance: Take two cases i and j where $y_i = 1$ and $y_j = 0$ $(i \neq j)$,

If $\hat{Pr}(y_i = 1) > \hat{Pr}(y_j = 1)$, then the pair is concordant

If $\hat{Pr}(y_i = 1) < \hat{Pr}(y_j = 1)$, then the pair is discordant

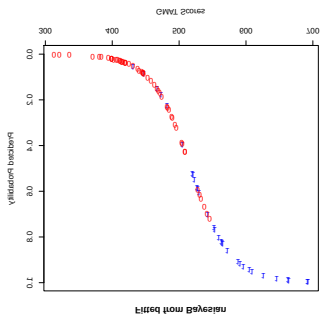
If $\hat{Pr}(y_i = 1) = \hat{Pr}(y_j = 1)$, then the pair is tie

The area under the ROC curve equals the concordance index. The concordance index is an estimate of the probability that predictions and outcomes are concordant.

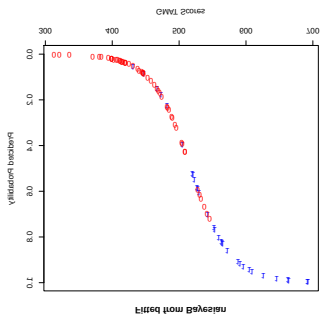
Our example, $c = 0.91$

This also provides a way to compare models, the solid dots in the next figure are from a model with more predictors. For example,. . .

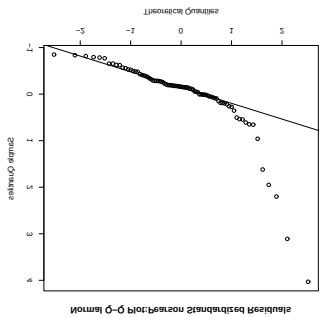
Fitted Probabilities



Fitted Probabilities & Data



QQ Plot of Standardized Residuals



admit $\sim$ GMAT + GPA

```
logreg2 ← "model {
        for (i in 1:n) {
        y[i] ~ dbern(p[i])
        p[i] ← 1/(1 + exp(-eta[i]))
        eta[i] ← b0 + b1*gmat[i] + b2*gre[i]
        }
hspace.5in b0 ~ dnorm(0,1/1000)
        b1 ~ dnorm(0,1/1000)
        b2 ~   dnorm(0,1/1000)
        }"
```

admit $\sim$ GMAT + GPA

```
logreg2.runjags ← run.jags (
        method="parallel",
        summarise=FALSE,
        plots=FALSE,
        sample=500000,
        model=logreg2,
        data=dataList,
        inits=start,
        monitor=c("b0","b1","b2"),
        n.chains=4,
        thin=20
)
```

admit $\sim$ GMAT + GPA

Calculating summary statistics...

Calculating the Gelman-Rubin statistic for 3 variables...

. JAGS model summary statistics from 2,000,000 samples (thin = 20; chains = 4; adapt+burnin = 5000):

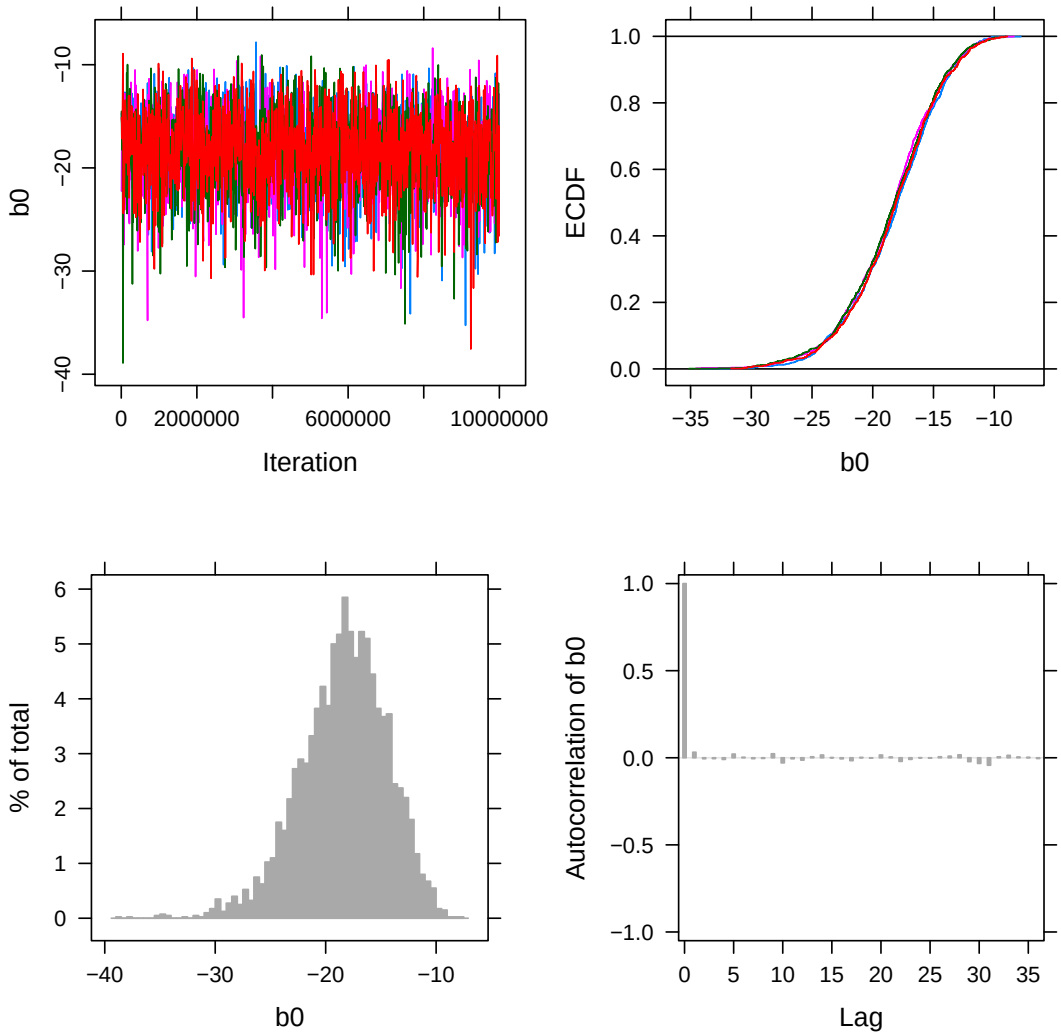
Note: Because I ran 4 chains with thin=20, the number of iterations was actually $500,000 \times 20 = 10,000,000$ per chain!

	Lower95	Median	Upper95	Mean	SD	Mode
b0	-26.101	-18.029	-11.082	-18.311	3.8781	–
b1	-3.8507	0.26955	2.9303	0.11768	1.7769	–
b2	-2.8994	-0.23269	3.8824	-0.082296	1.777	–

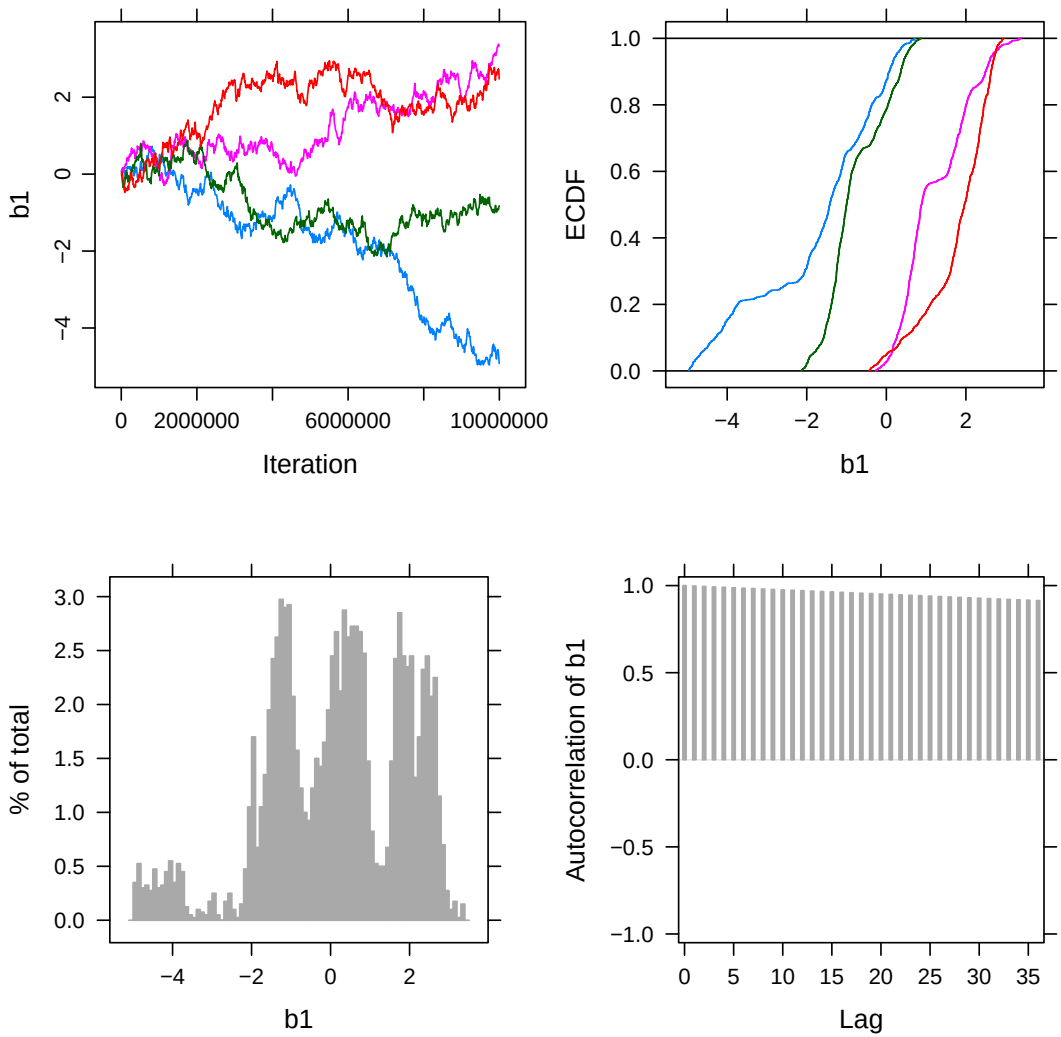
	MCerr	MC%fSD	SSeff	AC.200	psrf
b0	0.014642	0.4	70155	0.34361	1
b1	0.51228	28.8	12	0.99987	2.3686
b2	0.51484	29	12	0.99987	2.3685

Total time taken: 41.6 minutes

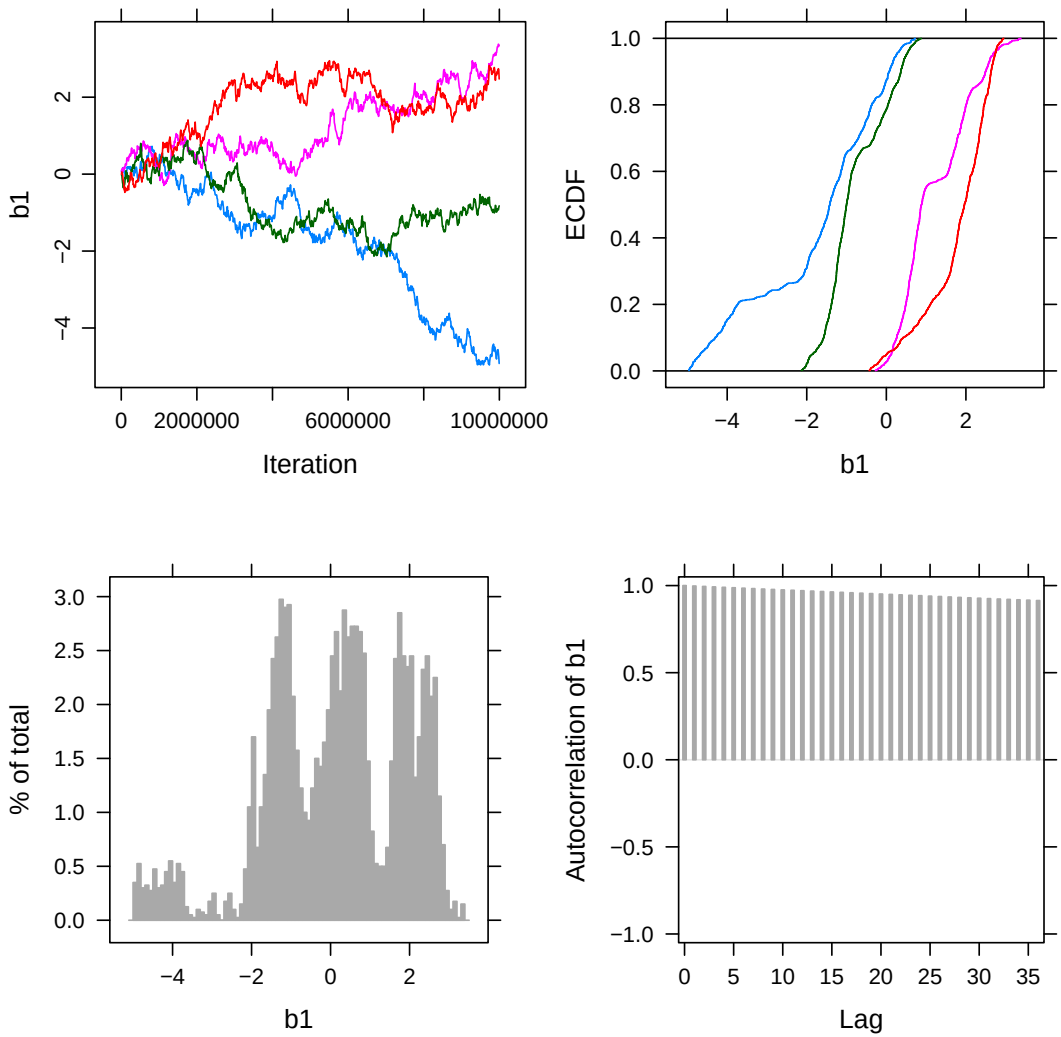
Trace, Density and Auto-Correlation



Really Bad


b1
Iteration
0
2000000
6000000
10000000
2
0
−2
−4
ECDF
b1
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
% of total
b1
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Autocorrelation of b1
Lag
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35

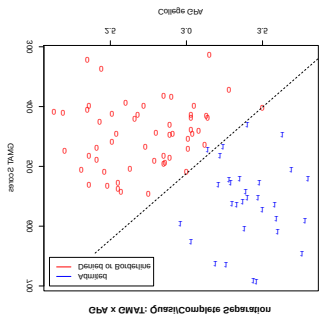
Really bad


b1
Iteration
0
2000000
6000000
10000000
2
0
−2
−4
ECDF
b1
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
% of total
b1
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Autocorrelation of b1
Lag
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35

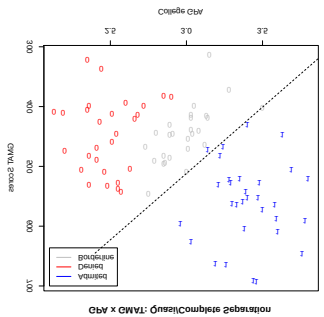
What's Going On

- $r(gmat, gpa) = .46$
- From glm:
  "Warning message:
  glm.fit: fitted probabilities numerically 0 or 1 occurred "
- Could centering help? Not here.
- Case of quasi/complete separation?

This is a "Problem"



This is a "Problem"



How Long Things Took

Note: I was typing and doing other things while models ran, so times are larger than would take if I wasn't typing and surfing.

	time	Notes
Check	6.1 sec	No errors, terrible fit
Increase iterations		Not good enough
thin=20	2.4 min	Better
+parrallel	40-45 sec	OK
+drop extra	40-43 sec	OK
2 predictors		
lots of iterations	$\approx$ 50 mintues	Terrible

- Parallel really speeds things up, but tied up computer.
- Adding iterations and emp.new really slows things down.
- Better starting values requires shorter chains.
- Expect long tmes as increase number of parameters and more complex models.
- Stan is slower than JAGS, but has other advantages.
- Illinois campus cluster: https://campuscluster.illinois.edu, which does have latest version of R.