Mixed effect probit regression

Genotypic fungal resistance

Dr. Jarad Niemi

STAT 544 - Iowa State University

April 30, 2024

Outline

- Probit regression
- Bayesian probit regression
  - Data augmentation
- Bayesian mixed effect probit regression
- Extensions
  - Ordinal categorical data
  - Nominal categorical data
  - Bayesian logistic regression

Probit regression

Consider the model

$$Y_i \overset{ind}{\sim} Ber(\theta_i)$$

where, for the ith observation,

- Y_i is binary indicating *success* and
- θ_i is the probability of success.

A probit regression model assumes

$$\theta_i = \Phi(X_i^\top \beta)$$

where

- X_i are the explanatory variables for the ith observation,
- Φ is the standard normal cumulative distribution function, and
- β is the vector of parameters to be estimated.

Low birth weight

```
      low               age              lwt              race       smoke              ptl               ht          
 Min.   :0.0000   Min.   :14.00   Min.   : 80.0   1:96   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  
 1st Qu.:0.0000   1st Qu.:19.00   1st Qu.:110.0   2:26   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000  
 Median :0.0000   Median :23.00   Median :121.0   3:67   Median :0.0000   Median :0.0000   Median :0.00000  
 Mean   :0.3122   Mean   :23.24   Mean   :129.8          Mean   :0.3915   Mean   :0.1958   Mean   :0.06349  
 3rd Qu.:1.0000   3rd Qu.:26.00   3rd Qu.:140.0          3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000  
 Max.   :1.0000   Max.   :45.00   Max.   :250.0          Max.   :1.0000   Max.   :3.0000   Max.   :1.00000  
       ui              ftv              bwt          
 Min.   :0.0000   Min.   :0.0000   Min.   : 709  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:2414  
 Median :0.0000   Median :0.0000   Median :2977  
 Mean   :0.1481   Mean   :0.7937   Mean   :2945  
 3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:3487  
 Max.   :1.0000   Max.   :6.0000   Max.   :4990  
```

```
m = glm(low~., family=binomial(link="probit"), data=birthwt[,-10]); summary(m)
```

```
Call:
glm(formula = low ~ ., family = binomial(link = "probit"), data = birthwt[,
    -10])

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.31431    0.24893  -5.280 1.29e-07 ***
age         -0.09774    0.11482  -0.851  0.39466
lwt         -0.27281    0.12217  -2.233  0.02555 *
race2        0.74961    0.31431   2.385  0.01708 *
race3        0.52183    0.25557   2.042  0.04117 *
smoke        0.56910    0.23469   2.425  0.01531 *
ptl          0.31968    0.20835   1.534  0.12495
ht           1.11161    0.41664   2.668  0.00763 **
ui           0.46517    0.27930   1.665  0.09581 .
ftv          0.02832    0.10161   0.279  0.78050
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 201.03  on 179  degrees of freedom
AIC: 221.03

Number of Fisher Scoring iterations: 5
```

Bayesian probit regression

Consider the model

$$\begin{aligned} Y_i &\overset{ind}{\sim} Ber(\theta_i) \\ \theta_i &= \Phi(X_i^\top \beta) \end{aligned}$$

with prior

$$\beta \sim N(b, B)$$

The posterior distribution is

$$\begin{aligned} p(\beta|y) &\propto p(y|\beta)p(\beta) \\ &\propto \left[\prod_{i=1}^n \Phi(X_i'\beta)^{y_i}[1-\Phi(X_i'\beta)]^{1-y_i}\right] e^{-(\beta-b)^\top B^{-1}(\beta-b)/2} \end{aligned}$$

But neither $p(\beta|y)$ nor $p(\beta_p|y, \beta_{-p})$ are a known distribution.

Data augmentation

An alternative construction of the model is

$$\begin{aligned} Y_i &= \mathrm{I}(\zeta_i > 0) \\ \zeta_i &\overset{ind}{\sim} N(X_i'\beta, 1) \end{aligned}$$

Note that

$$\begin{aligned} \theta_i &= P(Y_i = 1) \\ &= P(\zeta_i > 0) \\ &= P(X_i'\beta + \epsilon > 0) && \epsilon \sim N(0,1) \\ &= P(\epsilon > -X_i'\beta) \\ &= P(\epsilon < X_i'\beta) && \text{symmetry of standard normal} \\ &= \Phi(X_i'\beta) \end{aligned}$$

Thus, this is equivalent to the probit regression model.

Posterior distribution

Now, the likelihood is

$$p(y|\zeta) \propto \prod_{i=1}^{n} \left[\mathrm{I}(\zeta_i > 0)\mathrm{I}(y_i = 1) + \mathrm{I}(\zeta_i \leq 0)\mathrm{I}(y_i = 0) \right]$$

and

$$\zeta_i \overset{ind}{\sim} N(X_i'\beta, 1) \qquad \beta \sim N(b, B)$$

Therefore the *complete data likelihood* is

$$p(y, \zeta|\beta) \propto \prod_{i=1}^{n} N(\zeta_i|X_i'\beta, 1) \left[\mathrm{I}(\zeta_i > 0)\mathrm{I}(y_i = 1) + \mathrm{I}(\zeta_i \leq 0)\mathrm{I}(y_i = 0) \right]$$

Thus the posterior distribution is

$$p(\beta, \zeta|y) \propto p(y|\zeta, \beta)p(\zeta, \beta) = p(y|\zeta)p(\zeta|\beta)p(\beta) = p(y, \zeta|\beta)p(\beta)$$

and we will derive the full conditionals for $p(\beta|\zeta, y)$ and $p(\zeta|\beta, y)$.

Full conditional for β

The full conditional for β is

$$\begin{aligned} p(\beta|\ldots) &\propto p(y|\zeta)p(\zeta|\beta)p(\beta) \\ &\propto p(\zeta|\beta)p(\beta) \\ &= \left[\prod_{i=1}^{n} N(\zeta_i|X_i'\beta, 1)\right] N(\beta|b, B) \\ &= N(\zeta|X\beta, \mathrm{I})N(\beta|b, B) \end{aligned}$$

and thus $\beta|\ldots \sim N(\hat{\beta}, \hat{\Sigma}_\beta)$ with

$$\begin{aligned} \hat{\Sigma}_\beta &= [B^{-1} + X^\top X]^{-1} \\ \hat{\beta} &= \hat{\Sigma}_\beta[B^{-1}b + X^\top \zeta] \end{aligned}$$

Full conditional for ζ

The full conditional for ζ is

$$\begin{aligned} p(\zeta|\ldots) &\propto p(y|\zeta)p(\zeta|\beta)p(\beta) \\ &\propto p(y|\zeta)p(\zeta|\beta) \\ &= \prod_{i=1}^{n} N(\zeta_i|X_i'\beta,1)\left[\mathrm{I}(\zeta_i>0)\mathrm{I}(y_i=1)+\mathrm{I}(\zeta_i\leq 0)\mathrm{I}(y_i=0)\right] \end{aligned}$$

Thus the ζ_i are conditionally independent with distribution

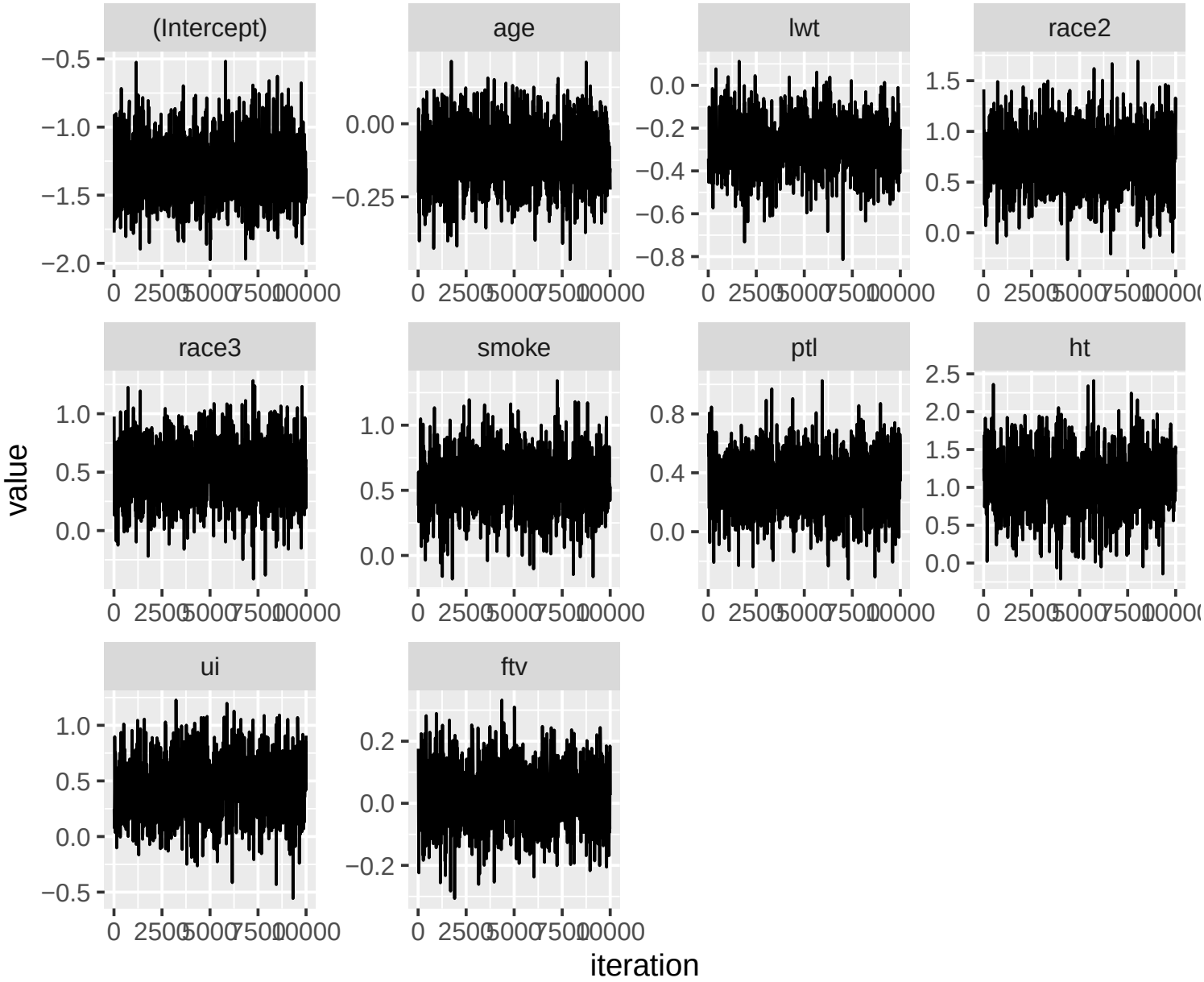
$$p(\zeta_i|y_i,\beta) = \begin{cases} N(\zeta_i|X_i'\beta,1)\mathrm{I}(\zeta_i>0) & \text{if } y_i=1 \\ N(\zeta_i|X_i'\beta,1)\mathrm{I}(\zeta_i\leq 0) & \text{if } y_i=0 \end{cases}$$

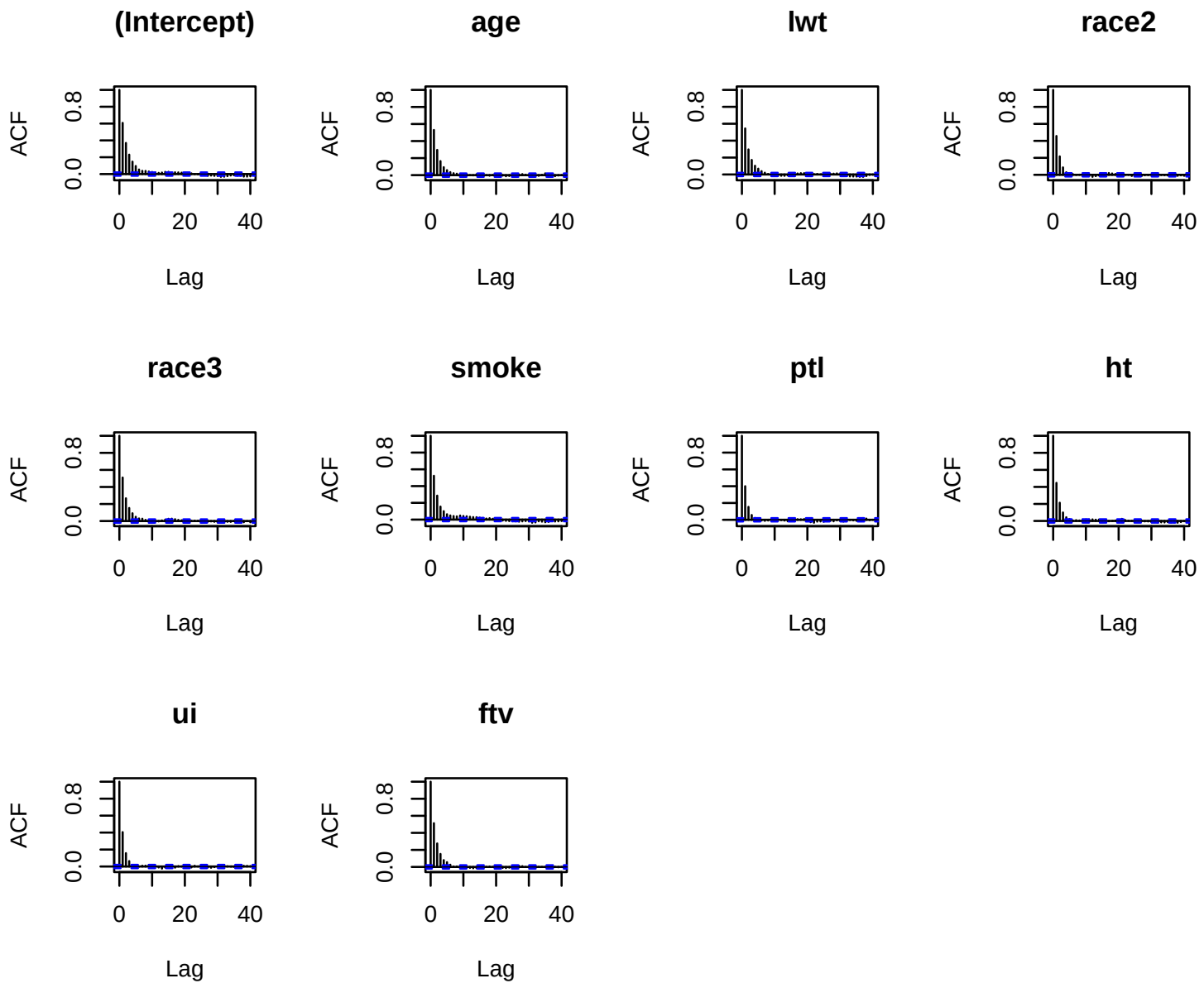
These can be drawn using the modified inverse cdf method.

```
mcmc = function(n_iter, y, X, beta0, Sigma_beta) {
  n = nrow(X)
  p = ncol(X)

  # Precalculate quantities
  y = (as.numeric(y)==1)
  n1 = sum( y)
  n0 = sum(!y)
  XX = t(X)%*%X
  Si = solve(Sigma_beta)
  Sib = Si%*%beta0

  # Saving structures
  beta_keep       = matrix(NA, n_iter, p)
  zeta_keep          = matrix(NA, n_iter, n)

  # Initial values
  m = glm(y~X-1, family=binomial("probit"))
  beta = coef(m)
  zeta = rep(NA,n)

  for (i in 1:n_iter) {
    # Sample zeta
    Xb = X%*%beta
    cut = pnorm(0,Xb)
    zeta[ y] = qnorm(runif(n1, cut[ y], 1), Xb[ y], 1)
    zeta[!y] = qnorm(runif(n0, 0, cut[!y]), Xb[!y], 1)

    # Sample beta
    S_hat = solve(Si+XX)
    b_hat = S_hat %*% (Sib+t(X)%*%zeta)
    beta = mvrnorm(1, b_hat, S_hat)

    # Record values
    beta_keep[i,] = beta
```

Run the MCMC

```
X = model.matrix(m) # Constructs the design matrix
p = ncol(X)
n_iter = 10000
system.time(out <- mcmc(n_iter, birthwt$low, X, rep(0,p), 3*diag(p)))

   user  system elapsed
  0.679   0.028   0.706
```


(Intercept)
age
lwt
race2
race3
smoke
ptl
ht
ui
ftv
value
iteration
0
2500
5000
7500
10000


(Intercept)
age
lwt
race2
race3
smoke
ptl
ht
ui
ftv
ACF
0.0
0.8
0
20
40
Lag

Credible intervals

```
# A tibble: 10 x 4
   variable      ess    lb    ub
   <fct>       <dbl> <dbl> <dbl>
 1 (Intercept)  1951 -1.76 -0.81
 2 age          2525 -0.33  0.11
 3 lwt          3012 -0.53 -0.05
 4 race2        3476  0.11  1.32
 5 race3        2472  0     0.98
 6 smoke        2110  0.09  1.01
 7 ptl          4769 -0.07  0.72
 8 ht           3293  0.26  1.87
 9 ui           4056 -0.08  0.98
10 ftv          3215 -0.19  0.22
```

Probit regression with random effects

Consider the probit regression model

$$\begin{aligned} Y_i &= \mathrm{I}(\zeta_i > 0) \\ \zeta &\sim N(\tilde{X}\tilde{\beta}, 1) \end{aligned}$$

where

$$\tilde{X} = [X \quad Zm] \qquad \tilde{\beta} = (\beta, \alpha)^\top$$

where X is the design matrix for fixed effects and Zm is the design matrix for the random effects. A common assumption is that the random effects are $\alpha \sim N(0, \sigma^2\mathrm{I})$. Thus the distribution on $\tilde{\beta}$ is

$$\tilde{\beta} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \sim N\left(\begin{bmatrix} b \\ 0 \end{bmatrix}, \begin{bmatrix} B & 0 \\ 0 & \sigma^2\mathrm{I} \end{bmatrix} \right)$$

where the precision is

$$\begin{bmatrix} B & 0 \\ 0 & \sigma^2\mathrm{I} \end{bmatrix}^{-1} = \begin{bmatrix} B^{-1} & 0 \\ 0 & \frac{1}{\sigma^2}\mathrm{I} \end{bmatrix}$$

Full posterior

The full posterior is

$$p(\zeta, \beta, \alpha, \sigma^2|y) \propto p(y|\zeta)p(\zeta|\tilde{\beta})p(\tilde{\beta}|\sigma^2)p(\sigma^2)$$

We have already derived the full conditionals

- $p(\tilde{\beta}|\ldots)$
- $p(\zeta|\ldots)$

but we need the full conditional for σ^2 to implement a Gibbs sampler.

Full conditional for σ^2

If we choose $\sigma \sim Unif(0, 10)$ and there are U random effects, then

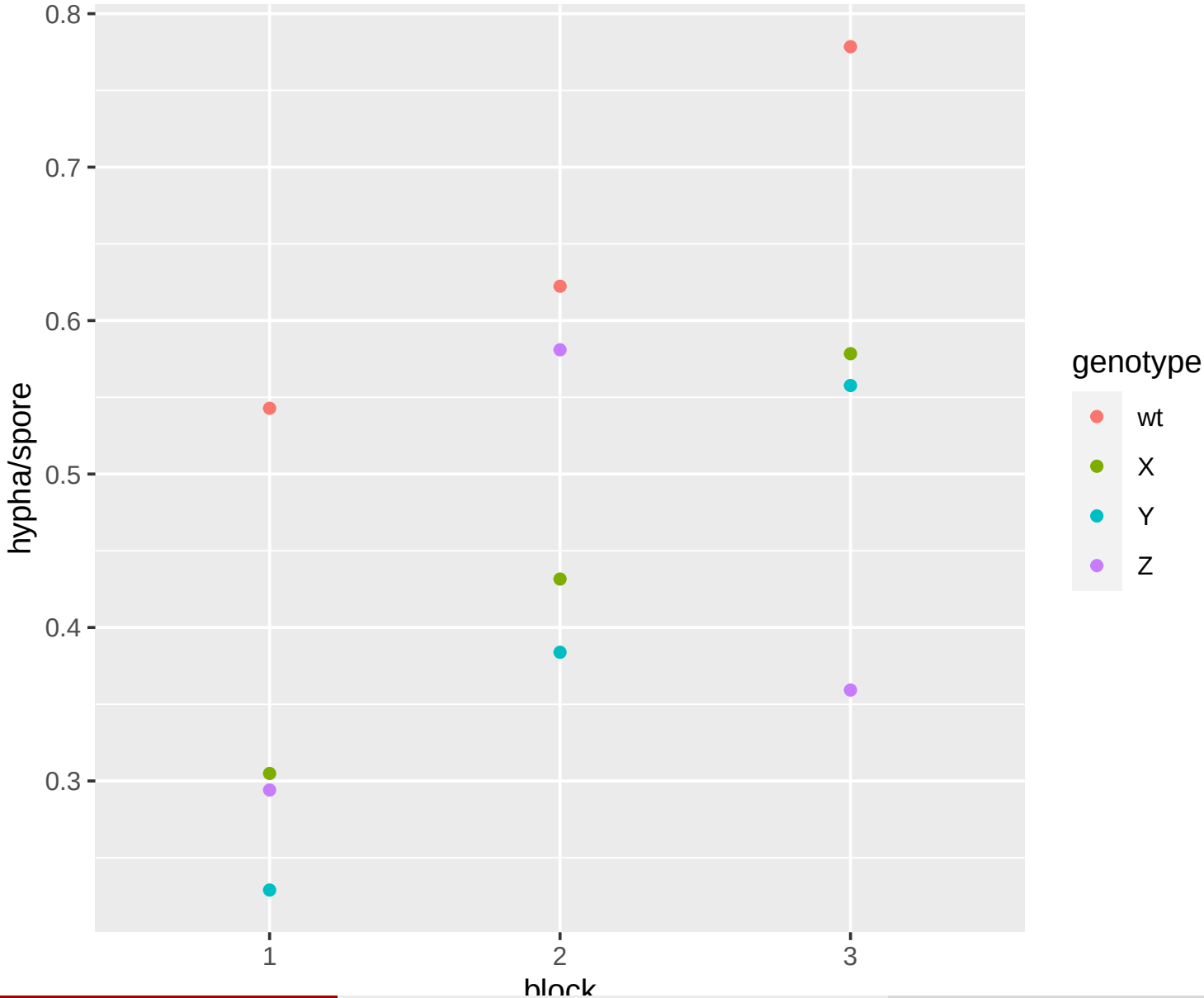
$$\begin{aligned}
p(\sigma^2|\ldots) &\propto p(y|\zeta)p(\zeta|\tilde{\beta})p(\tilde{\beta}|\sigma^2)p(\sigma^2) \\
&= p(\tilde{\beta}|\sigma^2)p(\sigma^2) \\
&\propto p(\alpha|\sigma^2)p(\sigma^2) \\
&\propto \textstyle\prod_{i=1}^{U} N(\alpha_i|0, \sigma^2)\frac{1}{\sigma}\mathrm{I}(0 < \sigma^2 < 100) \\
&\propto (\sigma^2)^{-U/2}e^{-\frac{1}{2\sigma^2}\alpha'\alpha}(\sigma^2)^{-1/2}\mathrm{I}(0 < \sigma^2 < 100) \\
&= (\sigma^2)^{-\frac{U-1}{2}-1}e^{-\frac{\alpha'\alpha}{2\sigma^2}}\mathrm{I}(0 < \sigma^2 < 100)
\end{aligned}$$

Thus $\sigma^2 \sim IG([U-1]/2, \alpha'\alpha/2)$ truncated to be smaller than 100. This can be drawn using the modified inverse cdf method.

Genotypic resistance to corn fungus

	X	genotype	block	spore	hypha	prop	pot
1	1	X	1	82	25	0.3048780	X1
6	6	X	2	95	41	0.4315789	X2
11	11	X	3	102	59	0.5784314	X3
16	16	Y	1	83	19	0.2289157	Y1
21	21	Y	2	99	38	0.3838384	Y2
26	26	Y	3	104	58	0.5576923	Y3
31	31	Z	1	102	30	0.2941176	Z1
36	36	Z	2	105	61	0.5809524	Z2
41	41	Z	3	103	37	0.3592233	Z3
46	46	wt	1	140	76	0.5428571	wt1
51	51	wt	2	143	89	0.6223776	wt2
56	56	wt	3	158	123	0.7784810	wt3

Corn fungus data set

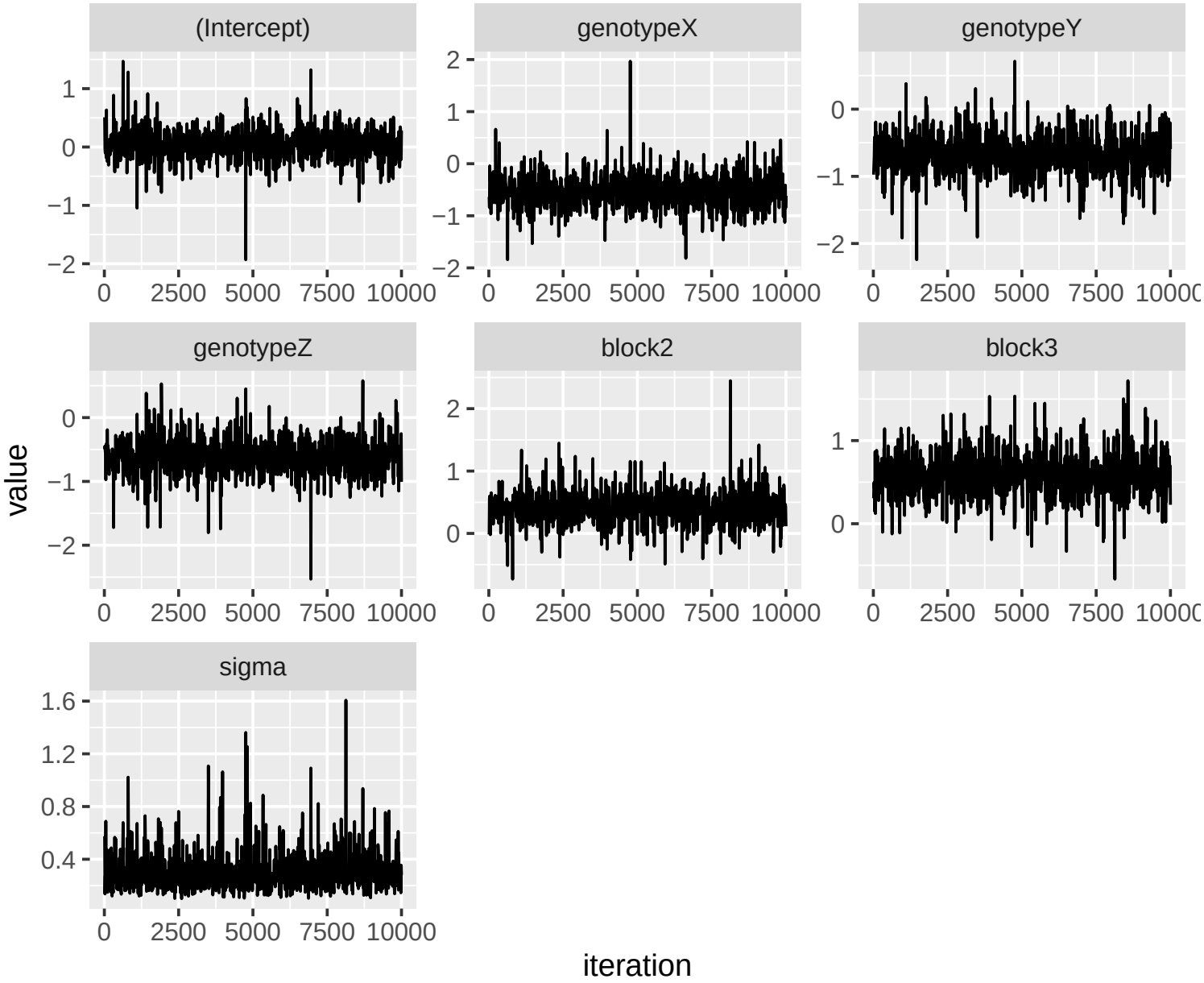


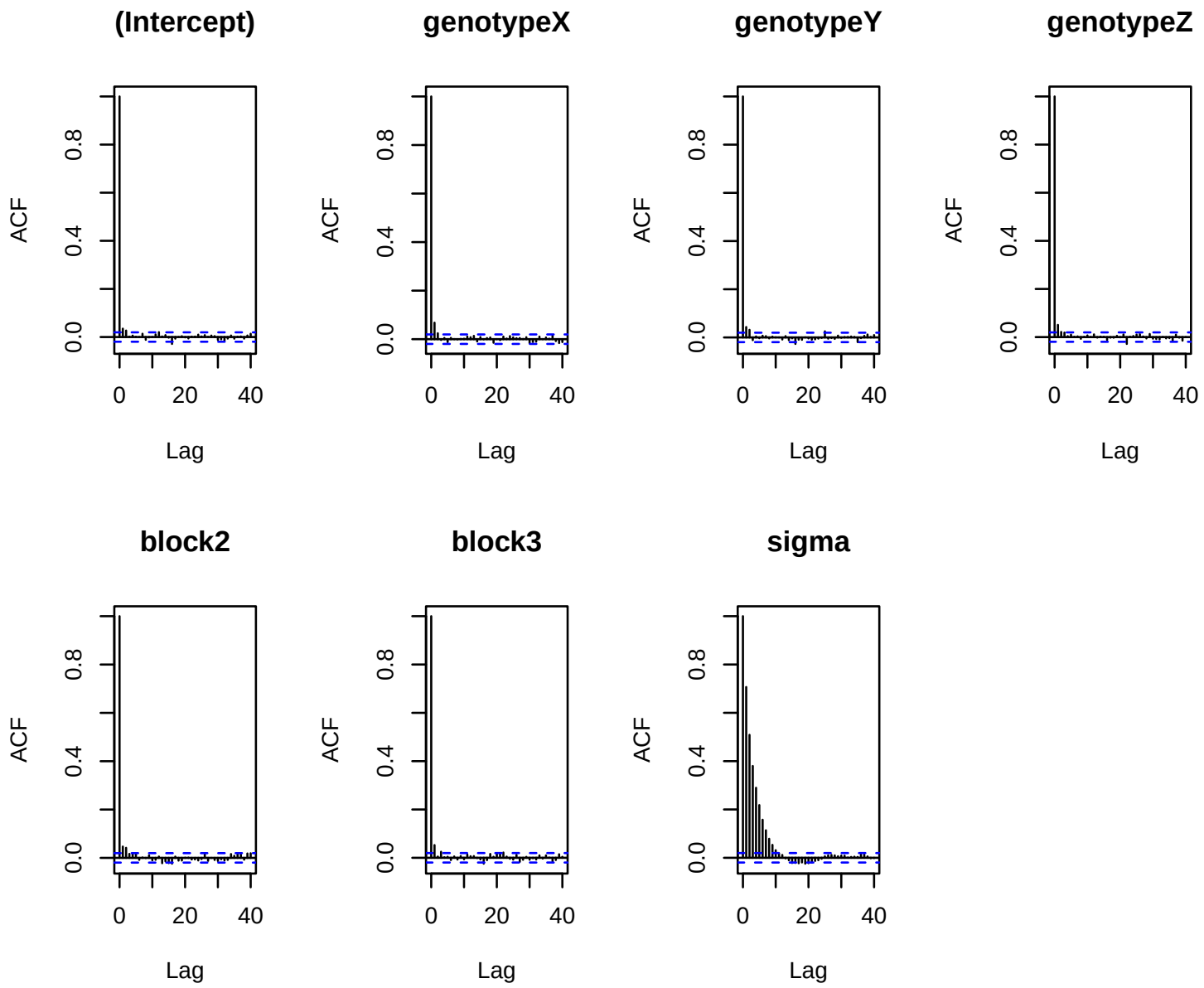
```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( probit )
Formula: cbind(hypha, spore - hypha) ~ block + genotype + (1 | pot)
   Data: d
Control: glmerControl(optimizer = "bobyqa")

     AIC      BIC   logLik deviance df.resid
    95.3     98.7    -40.6     81.3        5

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.45760 -0.35765  0.05486  0.36506  1.32376

Random effects:
 Groups Name        Variance Std.Dev.
 pot    (Intercept) 0.01773  0.1331
Number of obs: 12, groups:  pot, 12

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.05126    0.12429   0.412 0.680052
block2       0.42497    0.13027   3.262 0.001106 **
block3       0.60818    0.13006   4.676 2.92e-06 ***
genotypeX   -0.55654    0.14700  -3.786 0.000153 ***
genotypeY   -0.68630    0.14725  -4.661 3.15e-06 ***
genotypeZ   -0.62691    0.14500  -4.324 1.53e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
          (Intr) block2 block3 gntypX gntypY
block2    -0.530
block3    -0.526  0.522
genotypeX -0.520 -0.019 -0.027
genotypeY -0.514 -0.027 -0.035  0.454
```

```
mcmc = function(n_iter, y, X, Zm, beta0, Sigma_beta) {
  require(Matrix)
  n = nrow(X)
  p = ncol(X)
  q = ncol(Zm)

  # Initial values
  m = glm(y~0+X, family=binomial("probit"))
  beta = c(coef(m),rnorm(q))
  zeta = rep(NA,n)

  # Precalculate quantities
  y = (as.numeric(y)==1)
  n1 = sum( y)
  n0 = sum(!y)
  X  = cbind(X,Zm)
  XX = t(X)%*%X
  Si = solve(Sigma_beta)
  Sib = Si%*%beta0
  a = (q-1)/2

  # Saving structures
  beta_keep  = matrix(NA, n_iter, p)
  alpha_keep = matrix(NA, n_iter, q)
  sigma_keep = rep(NA, n_iter)

  for (i in 1:n_iter) {
    # Sample zeta
    Xb = X%*%beta
    cut = pnorm(0,as.numeric(Xb))
    zeta[ y] = qnorm(runif(n1, cut[ y], 1), Xb[ y], 1)
    zeta[!y] = qnorm(runif(n0, 0, cut[!y]), Xb[!y], 1)

    # Sample sigma
    alpha = beta[p+1:q]
```

```r
# Turn into binary data
dd = ddply(d, .(genotype, block, pot), function(x) {
  data.frame(y=c(rep(1, x$hypha), rep(0, x$spore - x$hypha)))
})

m = glmer(y ~ genotype + block + (1|pot), family = binomial("probit"), dd)

X = model.matrix(m)
Z = as.matrix(getME(m,"Z"))
p = ncol(X)
n_iter = 10000
system.time(out <- mcmc(n_iter, dd$y, X, Z, rep(0,p), 10*diag(p)))
```

```
   user  system elapsed
 15.596   0.348  15.944
```


(Intercept)
genotypeX
genotypeY
genotypeZ
block2
block3
sigma
value
iteration
0
2500
5000
7500
10000


(Intercept)
genotypeX
genotypeY
genotypeZ
block2
block3
sigma
ACF
0.0
0.4
0.8
0
20
40
Lag

Credible intervals

```
# A tibble: 7 x 4
  variable      ess    lb    ub
  <fct>       <dbl> <dbl> <dbl>
1 (Intercept)  9036 -0.45  0.55
2 genotypeX    8827 -1.13  0.05
3 genotypeY    9020 -1.25 -0.09
4 genotypeZ    8726 -1.17 -0.04
5 block2       8970 -0.09  0.94
6 block3       9376  0.08  1.09
7 sigma        1667  0.13  0.67
```

Contrasts to compare other genotypes

```
t(with(betas, data.frame("X-Y" = quantile(genotypeX-genotypeY, c(.025,.975)),
                         "Y-Z" = quantile(genotypeY-genotypeZ, c(.025,.975)),
                         "X-Z" = quantile(genotypeX-genotypeZ, c(.025,.975)), check.names=FALSE)))
```

```
          2.5%     97.5%
X-Y -0.4529010 0.7221782
Y-Z -0.6625842 0.5101285
X-Z -0.5228319 0.6426117
```

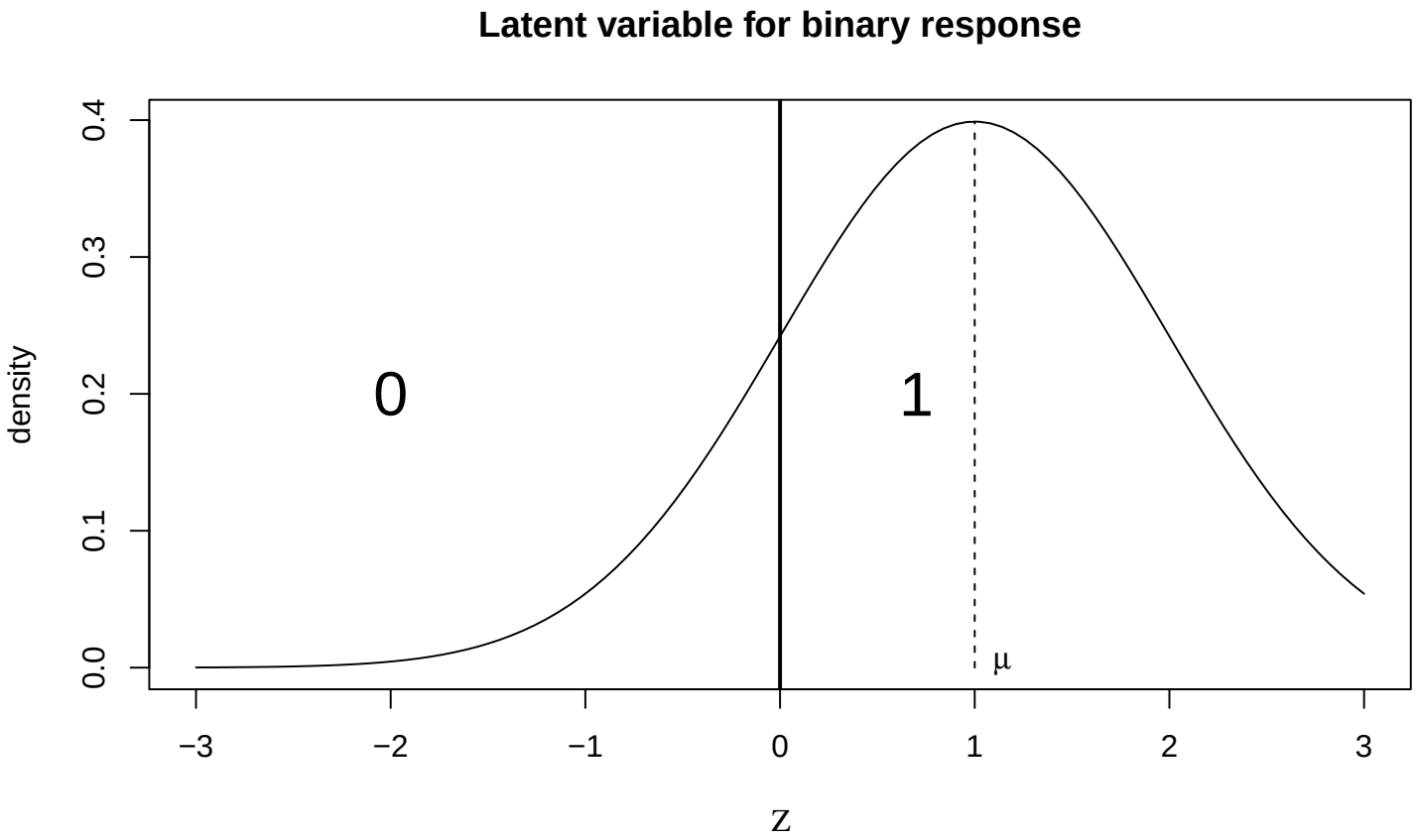
t priors

Suppose we want $\beta_j \overset{ind}{\sim} t_{v_j}(b_j, B_j)$. We can write this prior hierarchically via

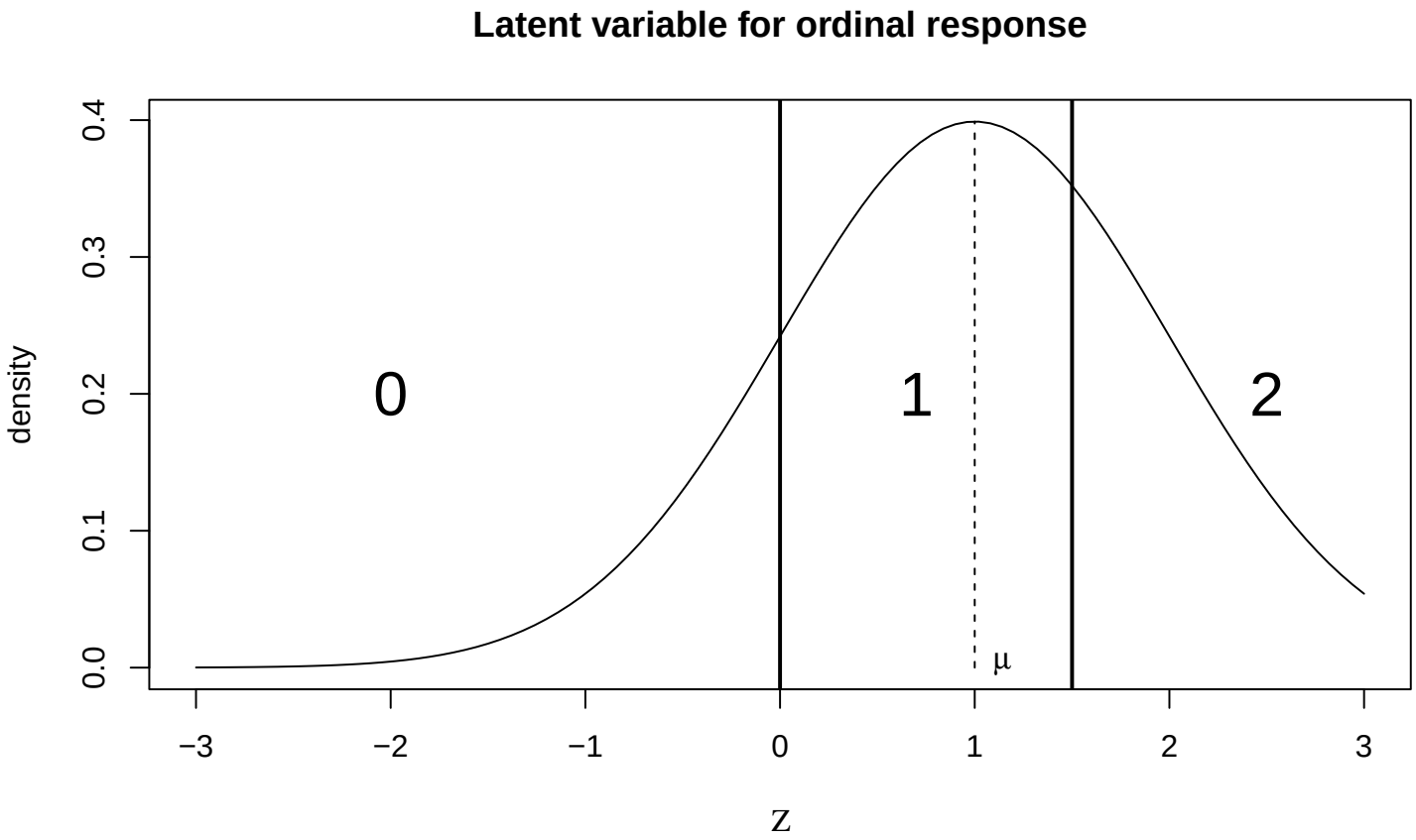
$$\beta_j|\tau_j^2 \overset{ind}{\sim} N(b_j, \tau_j^2), \qquad \tau_j^2 \sim \text{Inv} - \chi^2(v_j, B_j).$$

Now the MCMC can proceed exactly as before, but with the additional full conditional for $(\tau_1^2, \ldots, \tau_J^2)$ which will be independent inverse χ^2 distributions.

Binary response



Ordinal response with 3 categories



Unordered categorical response

Suppose Y_i is random variable with support $1, \ldots, K$ and

$$Pr(Y_i = k) = \theta_{ik}$$

where θ_{ik} may depend on explanatory variables for both i and k. For example, an individual is shopping for fruit then perhaps the age of the individual and the price of the fruits will affect the shopper's choice.

We can model this using data augmentation by introducing a latent utility ζ_{ik} for each shopper-fruit combination. Then the response is

$$Y_i = \text{argmax}_k \zeta_{ik}$$

and there is great flexibility in how the ζ_{ik} are modeled.

Bayesian logistic regression

$$
\begin{aligned}
Y_i &= \mathrm{I}(\zeta_i > 0) \\
\zeta_i &\overset{ind}{\sim} Logistic(X_i'\beta, 1)
\end{aligned}
$$

```
[1] "LogitPG: Iteration 2000"
[1] "LogitPG: Iteration 4000"
[1] "LogitPG: Iteration 6000"
[1] "LogitPG: Iteration 8000"
[1] "LogitPG: Iteration 10000"
                  2.5 %      97.5 %          X1    lb    ub
(Intercept) -3.14081600 -1.37637277 (Intercept) -3.29 -1.47
age         -0.54965424  0.22294401         age -0.58  0.23
lwt         -0.91079221 -0.07537297         lwt -0.97 -0.10
race2        0.24166064  2.32608774       race2  0.26  2.41
race3        0.02661178  1.76511921       race3  0.06  1.85
smoke        0.16158429  1.74790611       smoke  0.20  1.83
ptl         -0.12346116  1.24603059         ptl -0.13  1.30
ht           0.53239257  3.32119843          ht  0.60  3.48
ui          -0.14356295  1.67090307          ui -0.15  1.72
ftv         -0.28308378  0.39881567         ftv -0.29  0.40
```