2021 SISG Bayesian Statistics for Genetics R Notes: Generalized Linear Models

Jon Wakefield
Departments of Statistics and Biostatistics, University of Washington

2021-07-15

Overview

In this set of notes a number of generalized linear models (GLMs) and generalized linear mixed models (GLMMs) will be fitted using Bayesian methods.

The integrated nested Laplace approximation (INLA) computational technique will be illustrated

Case control example: Data

We analyze a case control example using logistic regression models, first using likelihood methods.

The data concern the numbers of cases (of the disease Leber Hereditary Optic Neuropathy) and controls as a function of genotype at a particular location (rs6767450).

```
x <- c(0, 1, 2)
# Case data for CC CT TT
y <- c(6, 8, 75)
# Control data for CC CT TT
z <- c(10, 66, 163)
```

Case control example: Likelihood analysis

We fit the logistic regression model as a generalized linear model and then examine the estimate and an asymptotic (large sample) 95% confidence interval.

```
logitmod <- glm(cbind(y, z) ~ x, family = "binomial")
thetahat <- logitmod$coeff[2]  # Log odds ratio
thetahat
##         x
## 0.4787428
exp(thetahat)  # Odds ratio
##        x
## 1.614044
V <- vcov(logitmod)[2, 2]  # standard error^2
# Asymptotic confidence interval for odds ratio
exp(thetahat - 1.96 * sqrt(V))
##         x
## 0.9879159
exp(thetahat + 1.96 * sqrt(V))
##        x
## 2.637004
```

Case control example: Likelihood analysis

Now let's look at a likelihood ratio test of $H_0 : \theta = 0$ where θ is the log odds ratio associated with the genotype (multiplicative model).

```
logitmod
##
## Call:  glm(formula = cbind(y, z) ~ x, family = "binomial")
##
## Coefficients:
## (Intercept)            x
##     -1.8077       0.4787
##
## Degrees of Freedom: 2 Total (i.e. Null);  1 Residual
## Null Deviance:       15.01
## Residual Deviance: 10.99     AIC: 27.79
dev <- logitmod$null.deviance - logitmod$deviance
dev
## [1] 4.01874
pchisq(dev, df = logitmod$df.residual, lower.tail = F)
## [1] 0.04499731
```

So just significant at the 5% level.

FTO Example: Data

We reproduce the least squares analysis of the FTO data.

The `lm` function uses MLE, which is equivalent to ordinary least squares.

```
load(url("http://faculty.washington.edu/kenrice/sisgbayes/yX_FTO.Rdata"))
liny <- yX$y
linxg <- yX$X[, "xg"]
linxa <- yX$X[, "xa"]
linxint <- yX$X[, "xg"] * yX$X[, "xa"]
ftodf <- list(liny = liny, linxg = linxg, linxa = linxa,
    linxint = linxint)
```

FTO Example: LS fit

```
ols.fit <- lm(liny ~ linxg + linxa + linxint, data = ftodf)
summary(ols.fit)
##
## Call:
## lm(formula = liny ~ linxg + linxa + linxint, data = ftodf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8008 -0.8844  0.2993  1.2270  2.4819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06822    1.42230  -0.048   0.9623
## linxg        2.94485    2.01143   1.464   0.1625
## linxa        2.84421    0.42884   6.632 5.76e-06 ***
## linxint      1.72948    0.60647   2.852   0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.918 on 16 degrees of freedom
## Multiple R-squared:  0.9393, Adjusted R-squared:  0.9279
## F-statistic: 82.55 on 3 and 16 DF,  p-value: 5.972e-10
```

INLA

Integrated nested Laplace approximation (INLA) is a technique for carrying out Bayesian computation.

It is not a standard R package and must be downloaded from the development website.

The `inla` function is the work horse.

```
# install.packages('INLA',
# repos='http://www.math.ntnu.no/inla/R/stable')
library(INLA)
# Data should be input to INLA as either a list or
# a dataframe
formula <- liny ~ linxg + linxa + linxint
lin.mod <- inla(formula, data = ftodf, family = "gaussian")
```

We might wonder, where are the priors?

We didn't specify any. . . but INLA has default choices.

FTO example via INLA: Lots of output available!

```
names(lin.mod)
##  [1] "names.fixed"                 "summary.fixed"
##  [3] "marginals.fixed"             "summary.lincomb"
##  [5] "marginals.lincomb"           "size.lincomb"
##  [7] "summary.lincomb.derived"     "marginals.lincomb.derived"
##  [9] "size.lincomb.derived"        "mlik"
## [11] "cpo"                         "po"
## [13] "waic"                        "model.random"
## [15] "summary.random"              "marginals.random"
## [17] "size.random"                 "summary.linear.predictor"
## [19] "marginals.linear.predictor"  "summary.fitted.values"
## [21] "marginals.fitted.values"     "size.linear.predictor"
## [23] "summary.hyperpar"            "marginals.hyperpar"
## [25] "internal.summary.hyperpar"   "internal.marginals.hyperpar"
## [27] "offset.linear.predictor"     "model.spde2.blc"
## [29] "summary.spde2.blc"           "marginals.spde2.blc"
## [31] "size.spde2.blc"              "model.spde3.blc"
## [33] "summary.spde3.blc"           "marginals.spde3.blc"
## [35] "size.spde3.blc"              "logfile"
## [37] "misc"                        "dic"
## [39] "mode"                        "neffp"
## [41] "joint.hyper"                 "nhyper"
## [43] "version"                     "Q"
## [45] "graph"                       "ok"
## [47] "cpu.used"                    "all.hyper"
## [49] ".args"                       "call"
## [51] "model.matrix"
```

FTO example: INLA analysis

The posterior means and posterior standard deviations are in very close agreement with the OLS fits presented earlier.

```
coef(ols.fit)
## (Intercept)        linxg        linxa      linxint
## -0.06821632  2.94485495  2.84420729  1.72947648
sqrt(diag(vcov(ols.fit)))
## (Intercept)        linxg        linxa      linxint
##   1.4222970   2.0114316   0.4288387   0.6064695
lin.mod$summary.fixed[, 1:5]
##                    mean        sd 0.025quant    0.5quant 0.975quant
## (Intercept) -0.06158122 1.4304379 -2.8994652 -0.06200624   2.774229
## linxg        2.93317509 2.0205097 -1.0787429  2.93377062   6.934649
## linxa        2.84236002 0.4313676  1.9859078  2.84245090   3.696813
## linxint      1.73264086 0.6094348  0.5236541  1.73244093   2.940860
```

The posterior means and standard deviations are in very close agreement with the OLS fits presented earlier.

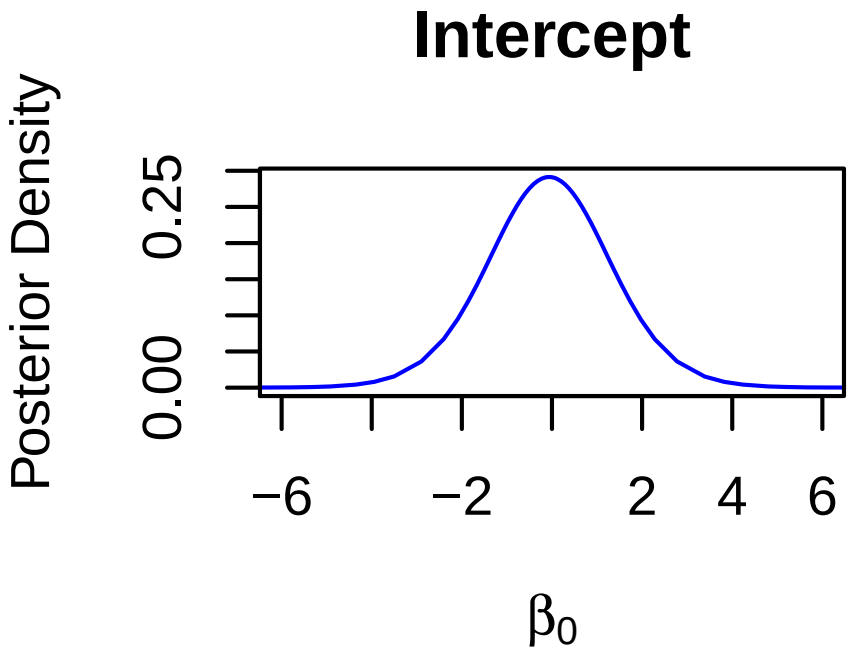
FTO Posterior marginals

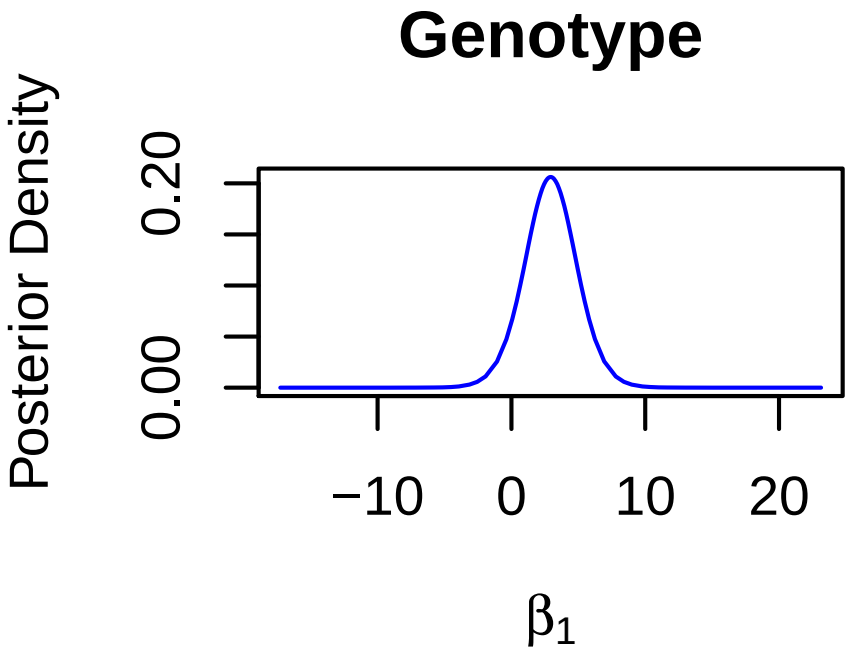
We now examine the posterior marginal distributions.

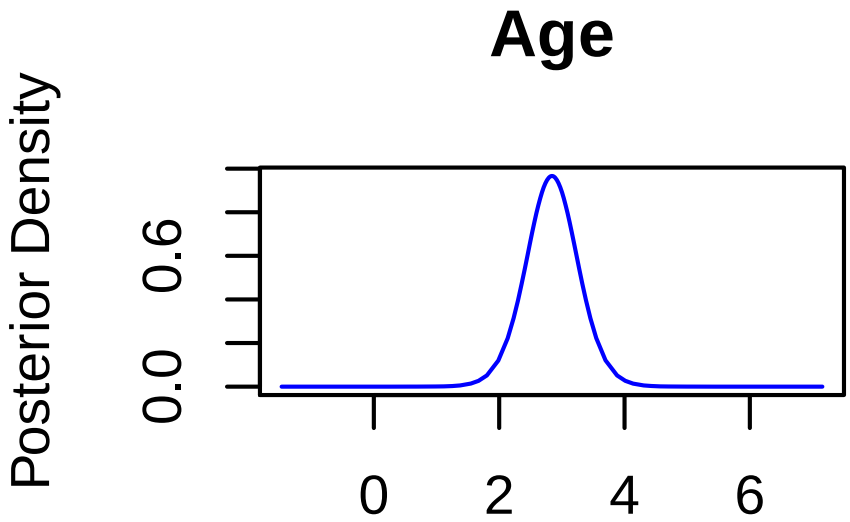
The posterior marginal distribution for the vector of regression coefficients (including the intercept) is given below, and then we examine the posterior marginal on the precision, $1/\sigma_\epsilon$.

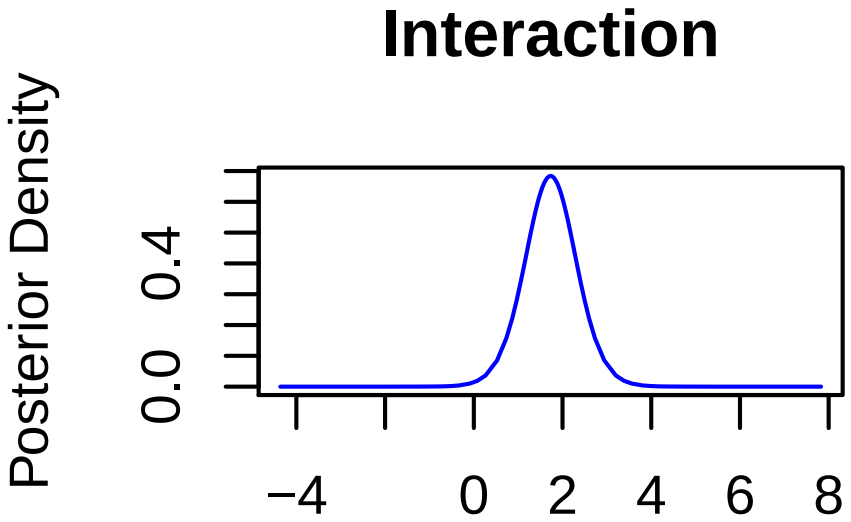
Check out the files that are written.

```
par(mfrow = c(2, 2))
plot(lin.mod$marginals.fixed$`(Intercept)`[, 2] ~ lin.mod$marginals.fixed$`(Interce
    1], xlab = expression(beta[0]), ylab = "Posterior Density",
    type = "l", col = "blue", xlim = c(-6, 6), main = "Intercept")
plot(lin.mod$marginals.fixed$linxg[, 2] ~ lin.mod$marginals.fixed$linxg[,
    1], xlab = expression(beta[1]), ylab = "Posterior Density",
    type = "l", col = "blue", main = "Genotype")
plot(lin.mod$marginals.fixed$linxa[, 2] ~ lin.mod$marginals.fixed$linxa[,
    1], xlab = expression(beta[2]), ylab = "Posterior Density",
    type = "l", col = "blue", main = "Age")
plot(lin.mod$marginals.fixed$linxint[, 2] ~ lin.mod$marginals.fixed$linxint[,
    1], xlab = expression(beta[3]), ylab = "Posterior Density",
    type = "l", col = "blue", main = "Interaction")
```


Intercept
Posterior Density
0.00
0.25
−6
−2
2
4
6
β_0


Genotype
Posterior Density
0.00
0.20
−10
0
10
20
β_1


Age
Posterior Density
0.0
0.6
0
2
4
6


Interaction
Posterior Density
0.0
0.4
−4
0
2
4
6
8

FTO example via INLA

In order to carry out model checking we rerun the analysis, but now switch on a flag to obtain fitted values.

```
lin.mod <- inla(liny ~ linxg + linxa + linxint, data = ftodf,
    family = "gaussian", control.predictor = list(compute = TRUE))
fitted <- lin.mod$summary.fitted.values[, 1]
# Now extract the posterior median of the
# measurement error sd
sigmamed <- 1/sqrt(lin.mod$summary.hyperpar[, 4])
```

FTO: Residual analysis

With the fitted values we can examine the fit of the model. In particular:

- Normality of the errors (sample size is relatively small).
- Errors have constant variance (and are uncorrelated).

FTO Residual analysis

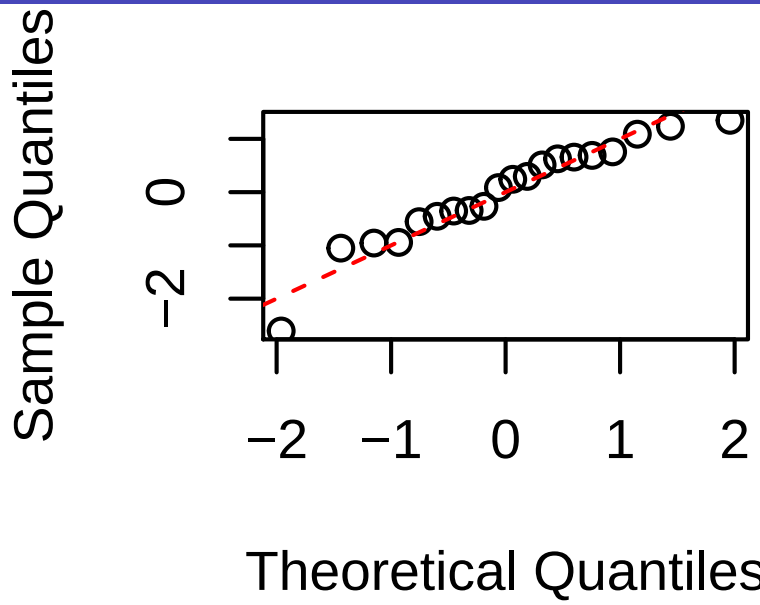
The code below forms residuals and then forms

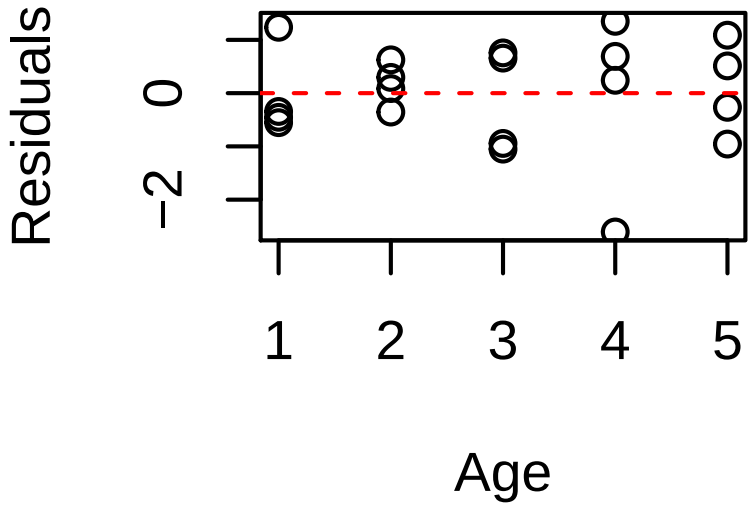
- a QQ plot to assess normality,
- a plot of residuals versus age, to assess linearity,
- a plot of residuals versus fitted values, to see if an unmodeled mean-variance relationship) and
- a plot of fitted versus observed for an overall assessment of fit.

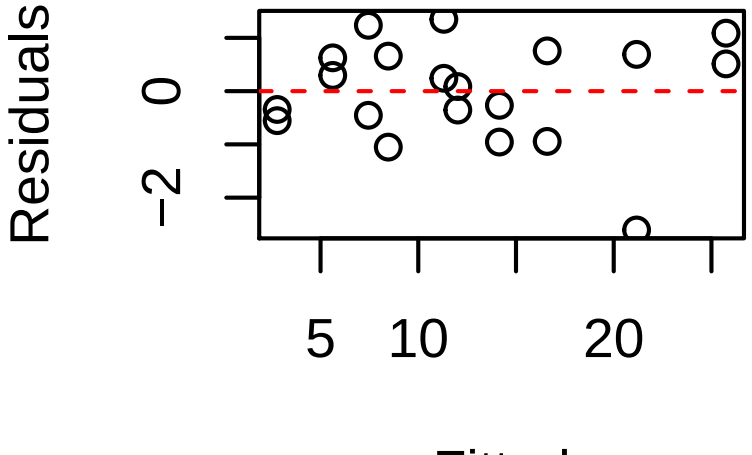
FTO: Residual analysis

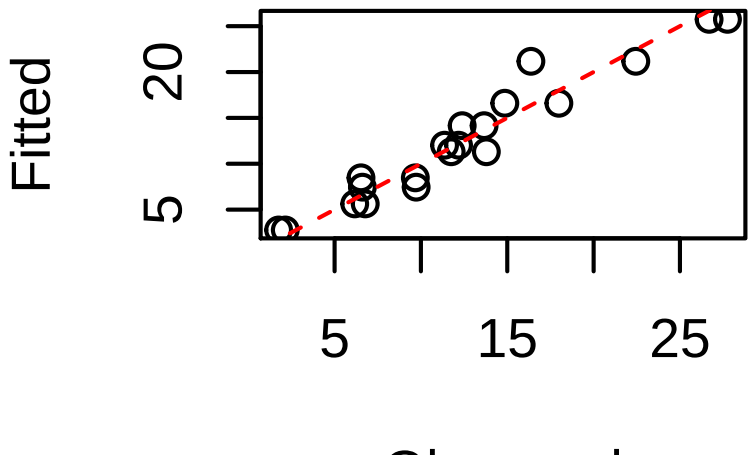
```
residuals <- (liny - fitted)/sigmamed
par(mfrow = c(2, 2))
qqnorm(residuals, main = "")
abline(0, 1, lty = 2, col = "red")
plot(residuals ~ linxa, ylab = "Residuals", xlab = "Age")
abline(h = 0, lty = 2, col = "red")
plot(residuals ~ fitted, ylab = "Residuals", xlab = "Fitted")
abline(h = 0, lty = 2, col = "red")
plot(fitted ~ liny, xlab = "Observed", ylab = "Fitted")
abline(0, 1, lty = 2, col = "red")
```

The model assumptions do not appear to be greatly invalidated here.


Sample Quantiles
−2
0
−2
−1
0
1
2
Theoretical Quantiles


Residuals
−2
0
1
2
3
4
5
Age


Residuals
−2
0
5
10
20


Fitted
5
20
5
15
25

Case-Control Example: INLA Analysis

We perform two analyses.

The first analysis uses the default priors in INLA (which are relatively flat).

```
x <- c(0, 1, 2)
y <- c(6, 8, 75)
z <- c(10, 66, 163)
cc.dat <- as.data.frame(rbind(y, z, x))
cc.mod <- inla(y ~ x, family = "binomial", data = cc.dat,
    Ntrials = y + z)
summary(cc.mod)
##
## Call:
##    c("inla(formula = y ~ x, family = \"binomial\", data = cc.dat, Ntrials
##    = y + ", " z)")
## Time used:
##     Pre = 4.82, Running = 0.328, Post = 0.305, Total = 5.45
## Fixed effects:
##                mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) -1.808 0.455      -2.75   -1.791     -0.963 -1.757   0
## x            0.480 0.250       0.01    0.473      0.994  0.458   0
##
```

Prior choice

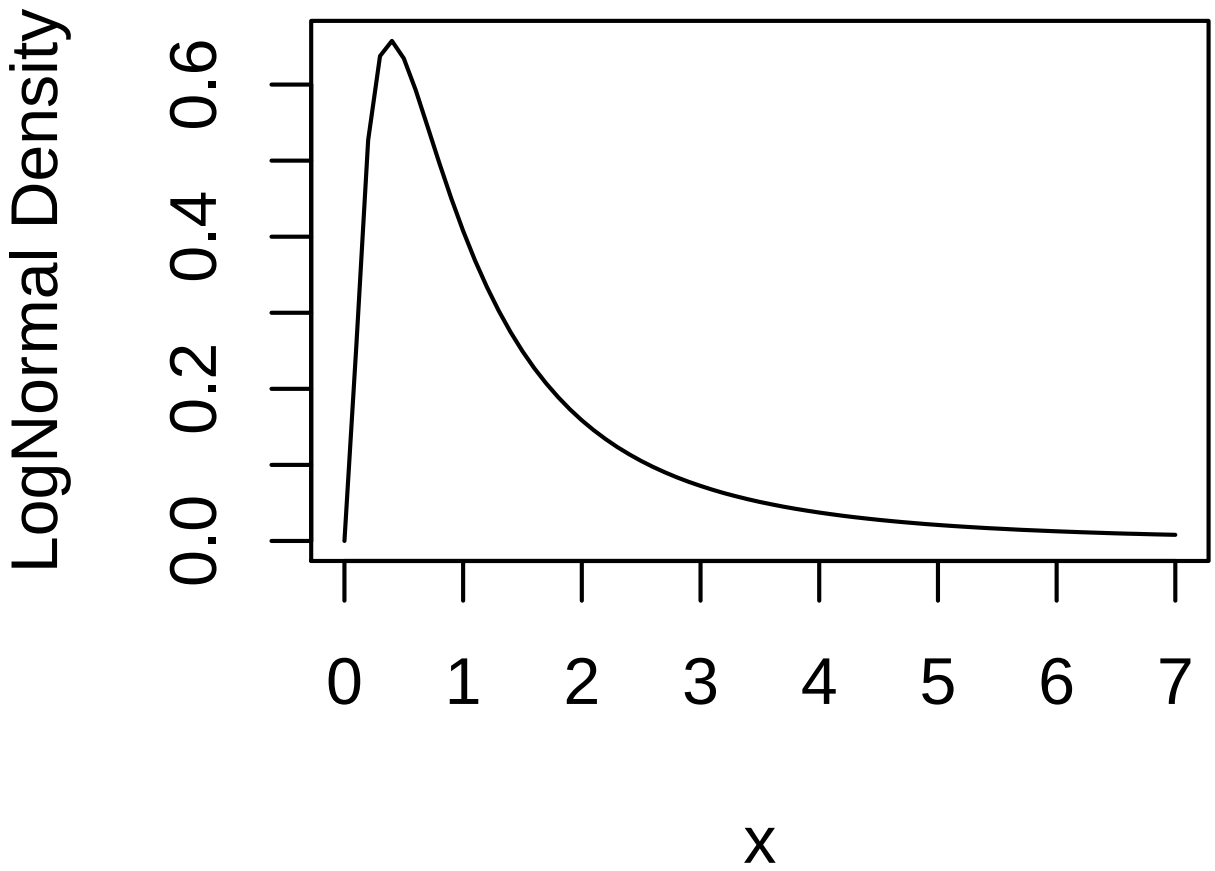
Suppose that for the odds ratio e^{β} we believe there is a 50% chance that the odds ratio is less than 1 and a 95% chance that it is less than 5; with $q_1 = 0.5, \theta_1 = 1.0$ and $q_2 = 0.95, \theta_2 = 5.0$, we obtain lognormal parameters $\mu = 0$ and $\sigma = (\log 5)/1.645 = 0.98$.

There is a function in the `SpatialEpi` package to find the parameters, as we illustrate.

```
library(SpatialEpi)
lnprior <- LogNormalPriorCh(1, 5, 0.5, 0.95)
lnprior
## $mu
## [1] 0
##
## $sigma
## [1] 0.9784688
```

Prior choice

```
plot(seq(0, 7, 0.1), dlnorm(seq(0, 7, 0.1), meanlog = lnprior$mu,
    sdlog = lnprior$sigma), type = "l", xlab = "x",
    ylab = "LogNormal Density")
```



Case-Control Example: INLA

```
# Now with informative priors
W <- LogNormalPriorCh(1, 1.5, 0.5, 0.975)$sigma^2
cc.mod2 <- inla(y ~ x, family = "binomial", data = cc.dat,
    Ntrials = y + z, control.fixed = list(mean.intercept = c(0),
        prec.intercept = c(0.1), mean = c(0), prec = c(1/W)))
summary(cc.mod2)
##
## Call:
##    c("inla(formula = y ~ x, family = \"binomial\", data = cc.dat, Ntrials
##    = y + ", " z, control.fixed = list(mean.intercept = c(0),
##    prec.intercept = c(0.1), ", " mean = c(0), prec = c(1/W)))")
## Time used:
##     Pre = 5.06, Running = 0.295, Post = 0.303, Total = 5.66
## Fixed effects:
##                mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) -1.323 0.290     -1.901   -1.319     -0.764 -1.312   0
## x            0.199 0.154     -0.100    0.198      0.503  0.195   0
##
## Expected number of effective parameters(stdev): 1.44(0.00)
## Number of equivalent replicates : 2.08
##
## Marginal log-Likelihood:  -16.64
```

The quantiles for θ can be translated to odds ratios by exponentiating

Longitudinal Example

Longitinal Example

We follow Wang et al.~(2018, Section 5.3) and analyze longitudinal data on reading scores (PIAT = Peabody Individual Achievement Test) measured on $n = 89$ children at ages 6.5, 8.5 and 10.5 years.

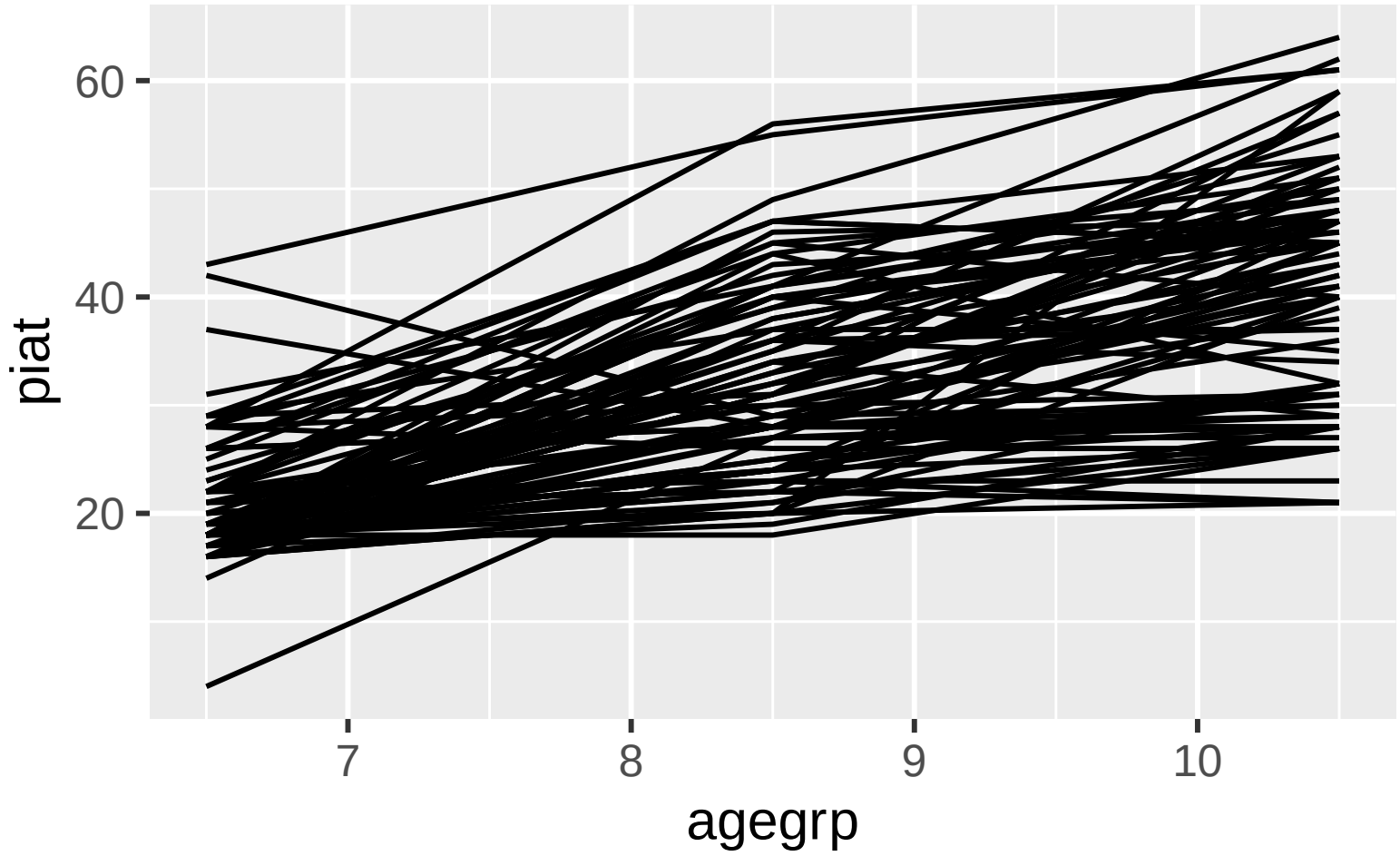
We expect scores on the same children to be correlated, and one way of acknowledging this is to include a random intercept in the model.

We let y_{ij} represent the reading score of child i at age t_j, with $t_1 = 6.5$, $t_2 = 8.5$ and $t_3 = 10.5$.

```
library(ggplot2)
library(brinla)
data(reading, package = "brinla")
```

Longitudinal Data

```
ggplot(reading, aes(agegrp, piat, group = id)) + geom_line()
```



A linear mixed effects model (LMEM)

We see a clear improvement in scores, which looks linear, but obvious between-child variartion.

We fit the model

$$\begin{aligned} Y_{ij} &= \beta_0 + \alpha_i + \beta_1 t_j + \epsilon_{ij} \\ \alpha_i &\sim_{iid} N(0, \sigma^2_\alpha) \\ \epsilon_{ij} &\sim_{iid} N(0, \sigma^2_\epsilon) \end{aligned}$$

where β_0, β_1 are fixed effects, α_i is a child-specific random effect and ϵ_{ij} is measurement error.

LMEM: Frequentist Analysis

We center age to provide an easier interpretation of the intercept, and (later) it will be easier to specify priors.

```
library(lme4)
reading$cagegrp <- reading$agegrp - 8.5
lmod <- lmer(piat ~ cagegrp + (1 | id), reading)
fixef(lmod)
## (Intercept)     cagegrp
##   31.224719    5.030899
sqrt(as.vector(diag(as.matrix(vcov(lmod)))))
## [1] 0.7090057 0.2510793
VarCorr(lmod)
##  Groups   Name        Std.Dev.
##  id       (Intercept) 5.4569
##  Residual             6.6996
```

Frequentist Analysis

```
summary(lmod)
## Linear mixed model fit by REML ['lmerMod']
## Formula: piat ~ agegrp + (1 | id)
##    Data: reading
##
## REML criterion at convergence: 1868.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.94599 -0.59452 -0.02802  0.50717  2.86759
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 29.78    5.457
##  Residual             44.89    6.700
## Number of obs: 267, groups:  id, 89
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -11.5379     2.2489  -5.131
## agegrp        5.0309     0.2511  20.037
##
## Correlation of Fixed Effects:
##        (Intr)
## agegrp -0.949
```

Bayesian Analysis

For the Bayesian analysis we have priors on the fixed effects:
$\beta_0 \sim N(40, 10^2)$, $\beta_1 \sim N(0, 4^2)$.

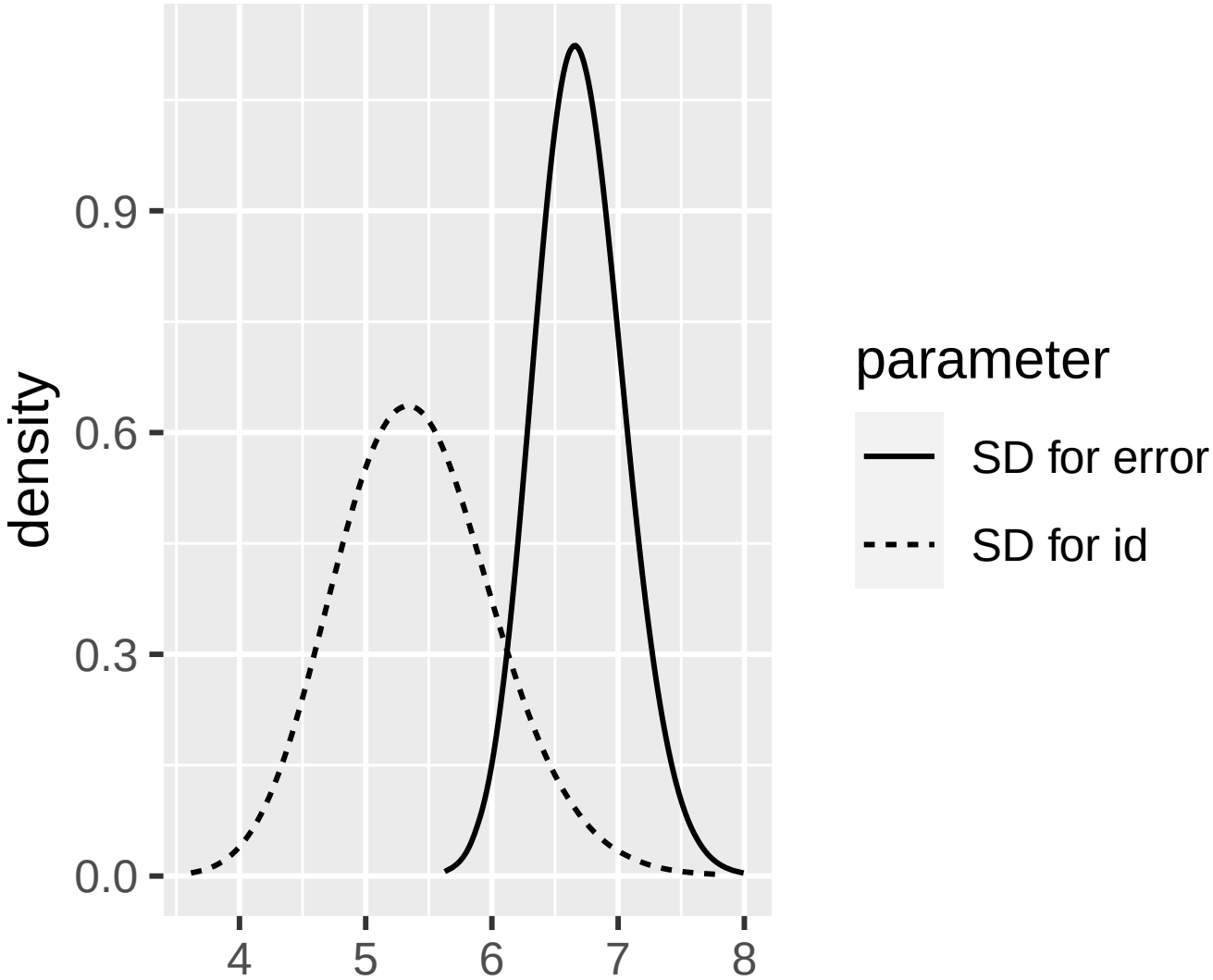
For the measurement error, the default relatively uninformative prior usually suffices (there is a lot of information in the data on the measurement error).

For the random effects variance we use a penalized complexity (PC) prior (Simpson et al.~2017) in which we say that $\Pr(\sigma_\alpha > 10) = 0.05$.

Bayesian Analysis

```
prior.fixed <- list(mean.intercept = 40, prec.intercept = 1/10^2,
    mean = 0, prec = 1/4^2)
pcprior <- list(prec = list(prior = "pc.prec", param = c(10,
    0.05)))
formula <- piat ~ cagegrp + f(id, model = "iid", hyper = pcprior)
imod <- inla(formula, family = "gaussian", data = reading,
    control.fixed = prior.fixed)
imod$summary.fixed[, 1:5]
##                  mean        sd 0.025quant  0.5quant 0.975quant
## (Intercept) 31.268518 0.7065240  29.879551 31.268012  32.658816
## cagegrp      5.010983 0.2516788   4.515954  5.011086   5.504963
bri.hyperpar.summary(imod)
##                                          mean        sd   q0.025     q0.5   q0.975
## SD for the Gaussian observations 6.698059 0.3553508 6.033866 6.685126 7.428613
## SD for id                        5.403906 0.6263179 4.254674 5.376843 6.709986
##                                      mode
## SD for the Gaussian observations 6.657952
## SD for id                        5.334598
```

```
bri.hyperpar.plot(imod)
```



Prediction

We now suppose we have a new child with measurements of 18 and 25 at years 6.5 and 8.5, and we want to predict their outcome at 10.5, under the assumption that they are exchangeable with the individuals in the dataset.

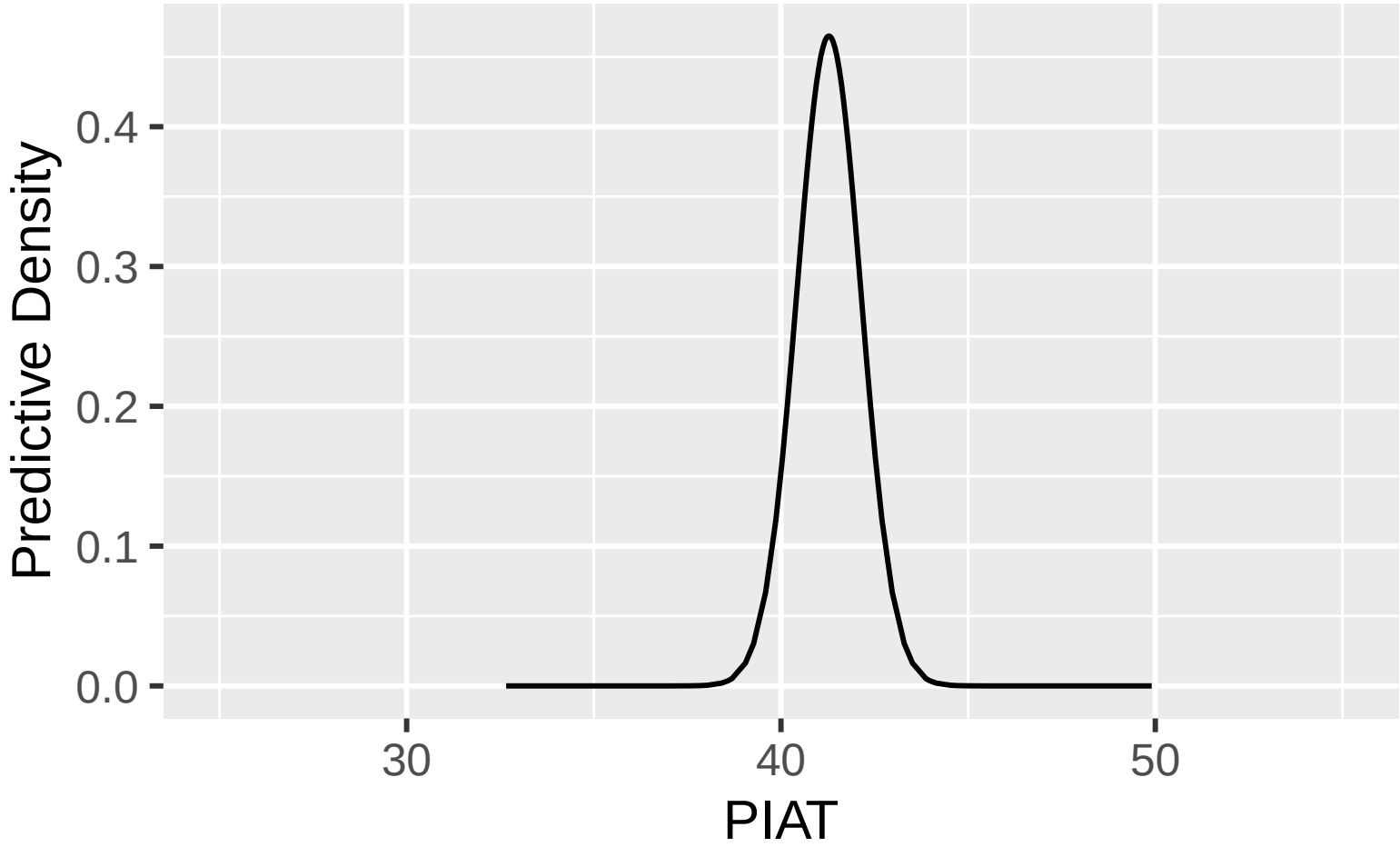
We code the missing value as `NA` and add the individual to the dataset.

Then we obtain the fitted value for this point (which is the 270th in the dataset).

Prediction

```
newsub <- data.frame(id = 90, agegrp = c(6.5, 8.5,
    10.5), cagegrp = c(-2, 0, 2), piat = c(20, 31,
    NA))
nreading <- rbind(reading, newsub)
prior.fixed <- list(mean.intercept = 40, prec.intercept = 1/10^2,
    mean = 0, prec = 1/4^2)
pcprior <- list(prec = list(prior = "pc.prec", param = c(10,
    0.05)))
formula <- piat ~ cagegrp + f(id, model = "iid", hyper = pcprior)
imodp <- inla(formula, family = "gaussian", data = nreading,
    control.fixed = prior.fixed, control.predictor = list(compute = TRUE))
pm90 <- imodp$marginals.fitted.values[[270]]
```

```
ggplot(data.frame(pm90), aes(x, y)) + geom_line() +
    xlim(c(25, 55)) + xlab("PIAT") + ylab("Predictive Density")
```



Approximate Bayes

Approximate Bayes

We return to the case control example seen earlier.

Below we construct the posterior by hand

```
x <- c(0, 1, 2)
y <- c(6, 8, 75)
z <- c(10, 66, 163)
logitmod <- glm(cbind(y, z) ~ x, family = "binomial")
thetahat <- logitmod$coef[2]
V <- vcov(logitmod)[2, 2]
# 97.5 point of prior is log(1.5) so that we with
# prob 0.95 we think theta lies in (2/3,1.5)
W <- LogNormalPriorCh(1, 1.5, 0.5, 0.975)$sigma^2
```

Approximate Bayes: estimation

```
r <- W/(V + W)
r
## [1] 0.4055539
# Not so much data here, so weight on prior is
# high.  Bayesian posterior median
exp(r * thetahat)
##        x
## 1.214286
# Shrunk towards prior median of 1 Note: INLA
# estimate (with same prior) is 1.22 and
# approximate posterior SD here is
# sqrt(rV)=0.159, INLA version is 0.154.
# Bayesian approximate 95% credible interval
exp(r * thetahat - 1.96 * sqrt(r * V))
##         x
## 0.8882832
exp(r * thetahat + 1.96 * sqrt(r * V))
##        x
## 1.659932
```

Approximate Bayes: hypothesis testing

Now we turn to testing using Bayes factors.

We examine the sensitivity to the prior on the alternative, π_1.

```
pi1 <- c(1/2, 1/100, 1/1000, 1/10000, 1/1e+05)  # 5 prior probs on the null
source("http://faculty.washington.edu/jonno/BFDP.R")
BFcall <- BFDPfunV(thetahat, V, W, pi1)
BFcall
## $BF
##         x
## 0.6182773
##
## $pH0
##         x
## 0.256323
##
## $pH1
##         x
## 0.4145761
##
## $BFDP
## [1] 0.3820589 0.9839253 0.9983836 0.9998383 0.9999838
```

Exercises

- For the case-control data, suppose we wish to specify a prior with a 5% point for the odds ratio of 0.2 and a 95% point for the odds ratio of 5:
  - Use the `LogNormalPriorCh` to find the appropriate normal distribution for the log odds ratio
  - Use this prior within INLA and report the posterior median and a 95% interval for the log odds ratio
  - Are these summaries very different from the INLA fit with default priors?