570 2021 Advanced Regression Modeling R Notes: INLA and Stan

Jon Wakefield
Departments of Statistics and Biostatistics, University of Washington

2021-10-26

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.

Two primary computational techniques will be illustrated:

– The integrated nested Laplace approximation (INLA) method using `INLA`

– Markov chain Monte Carlo (MCMC) using `Stan`

Linear Model Example

We consider a linear model example with the response Y being weight and two covariates:

- fto heterozygote, $x_g \in \{0, 1\}$
- age in weeks $x_a \in \{1, 2, 3, 4, 5\}$

We will examine the fit of the model

$$E[Y|x_g, x_a] = \beta_0 + \beta_g x_g + \beta_a x_a + \beta_{int} x_g x_a,$$

with independent normal errors, and compare with a Bayesian analysis.

Linear Model Example: Data

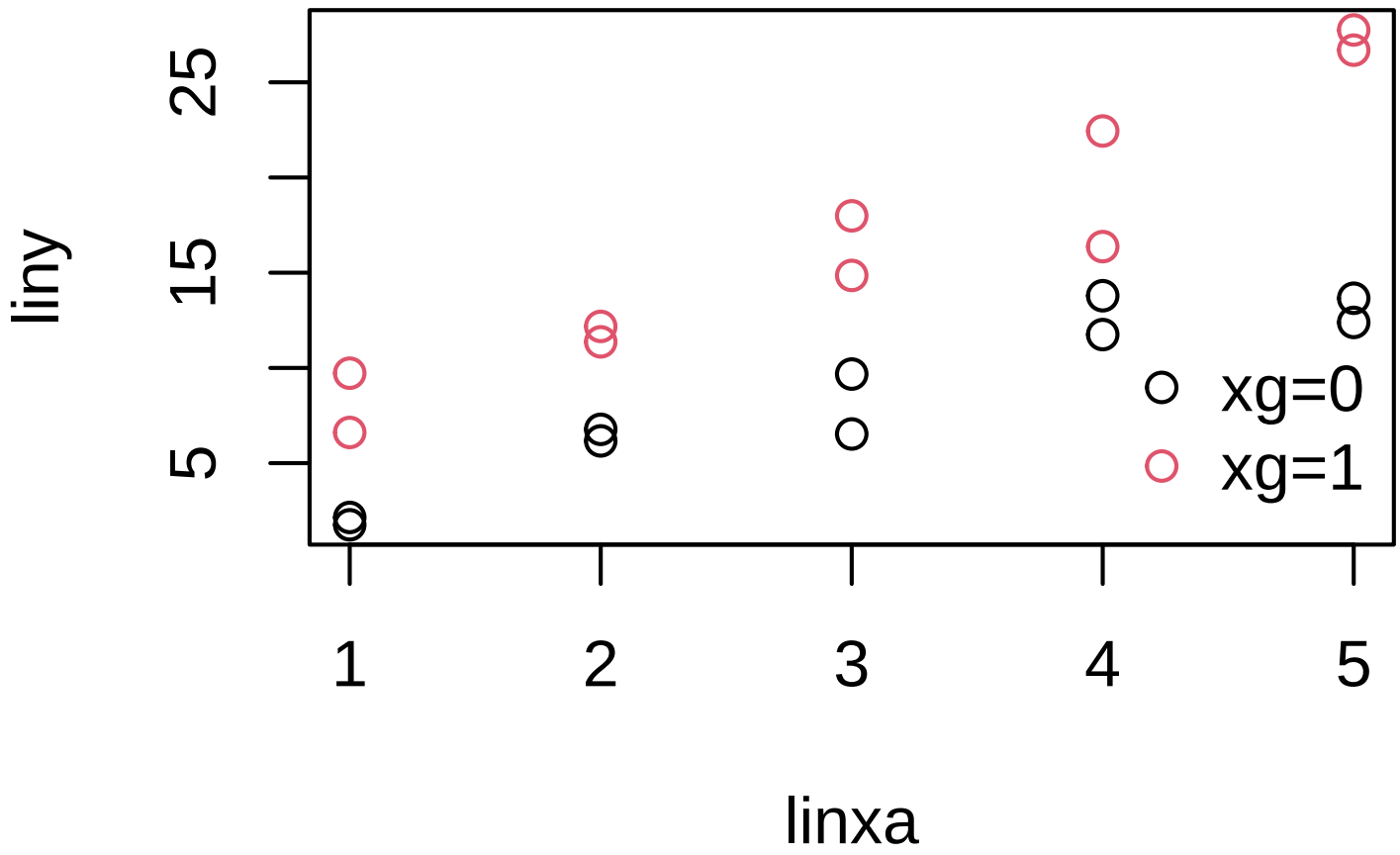
We first obtain the least squares analysis of the FTO data.

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)
```

Linear Model Example: Data

```
plot(liny ~ linxa, col = as.factor(linxg))
legend("bottomright", legend = c("xg=0", "xg=1"), col = 1:2,
    pch = 1, bty = "n")
```



Linear Model 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.

Linear Model example: 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
##                    mean        sd 0.025quant    0.5quant 0.975quant        mode
## (Intercept) -0.06158122 1.4304379 -2.8994652 -0.06200624   2.774229 -0.06259288
## linxg        2.93317509 2.0205097 -1.0787429  2.93377062   6.934649  2.93495202
## linxa        2.84236002 0.4313676  1.9859078  2.84245090   3.696813  2.84264183
## linxint      1.73264086 0.6094348  0.5236541  1.73244093   2.940860  1.73215926
##                     kld
## (Intercept) 2.811124e-08
## linxg       2.196343e-08
## linxa       2.904548e-08
## linxint     2.378588e-08
```

Linear Model example: INLA analysis

Posterior univariate marginal summaries:

```
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
```

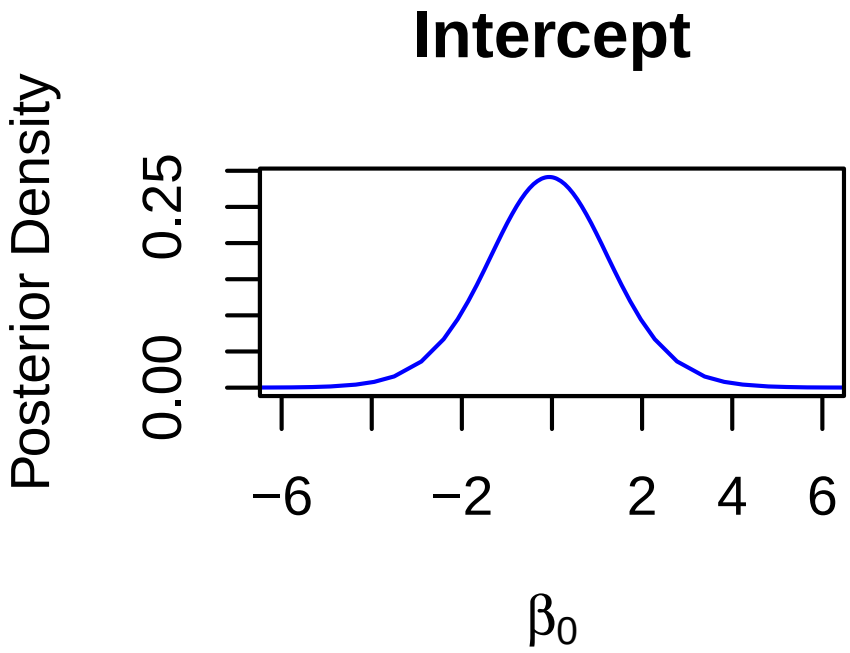
The posterior means and standard deviations are in very close agreement with the OLS fits presented earlier.

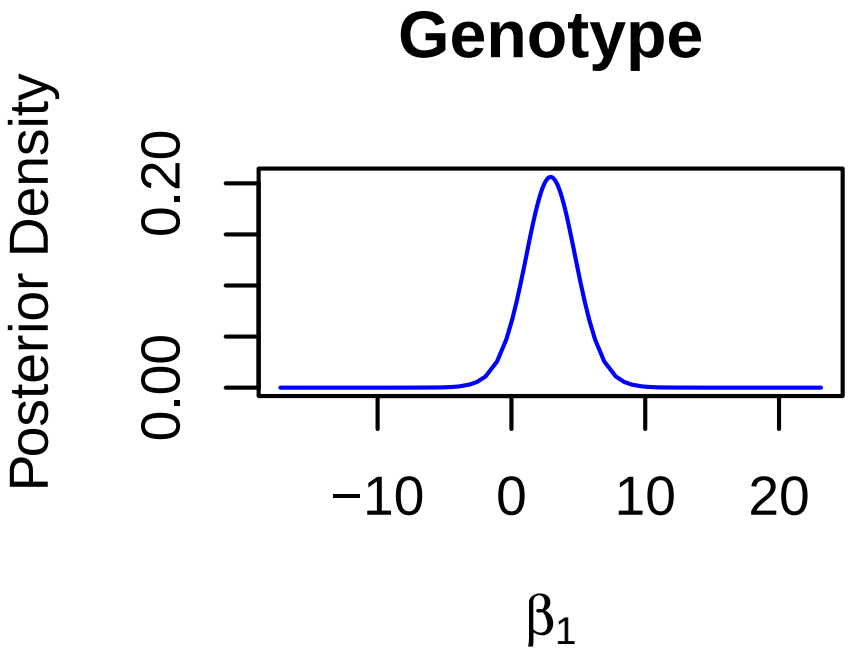
Linear Model Posterior marginals

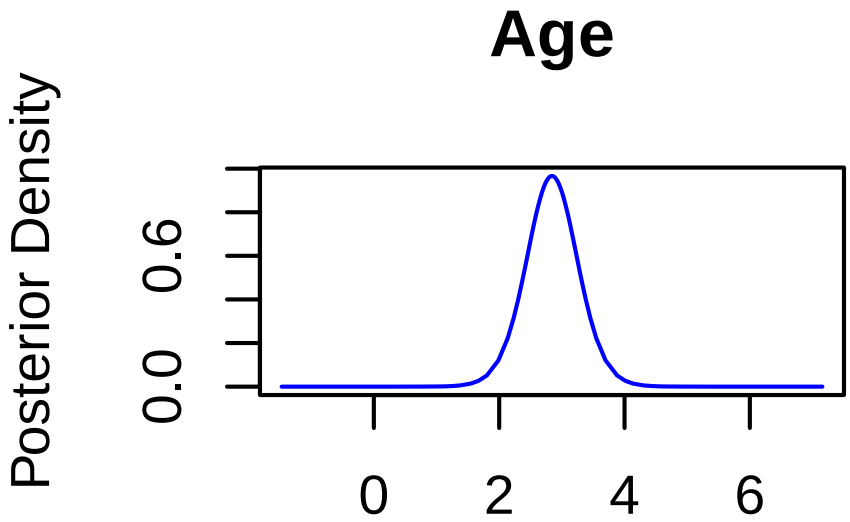
We now examine the posterior marginal distributions.

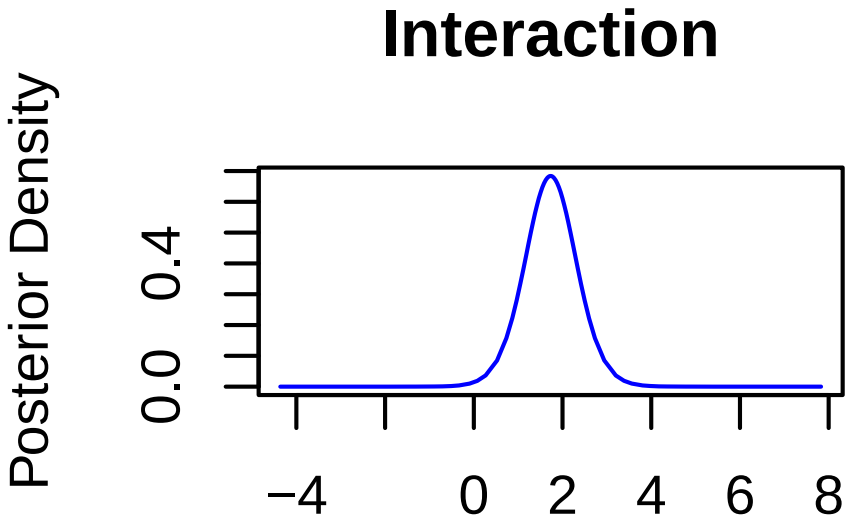
The posterior marginal distribution for the vector of regression coefficients (including the intercept) is given below.

```
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

Linear Model 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).

Linear Model Residual analysis

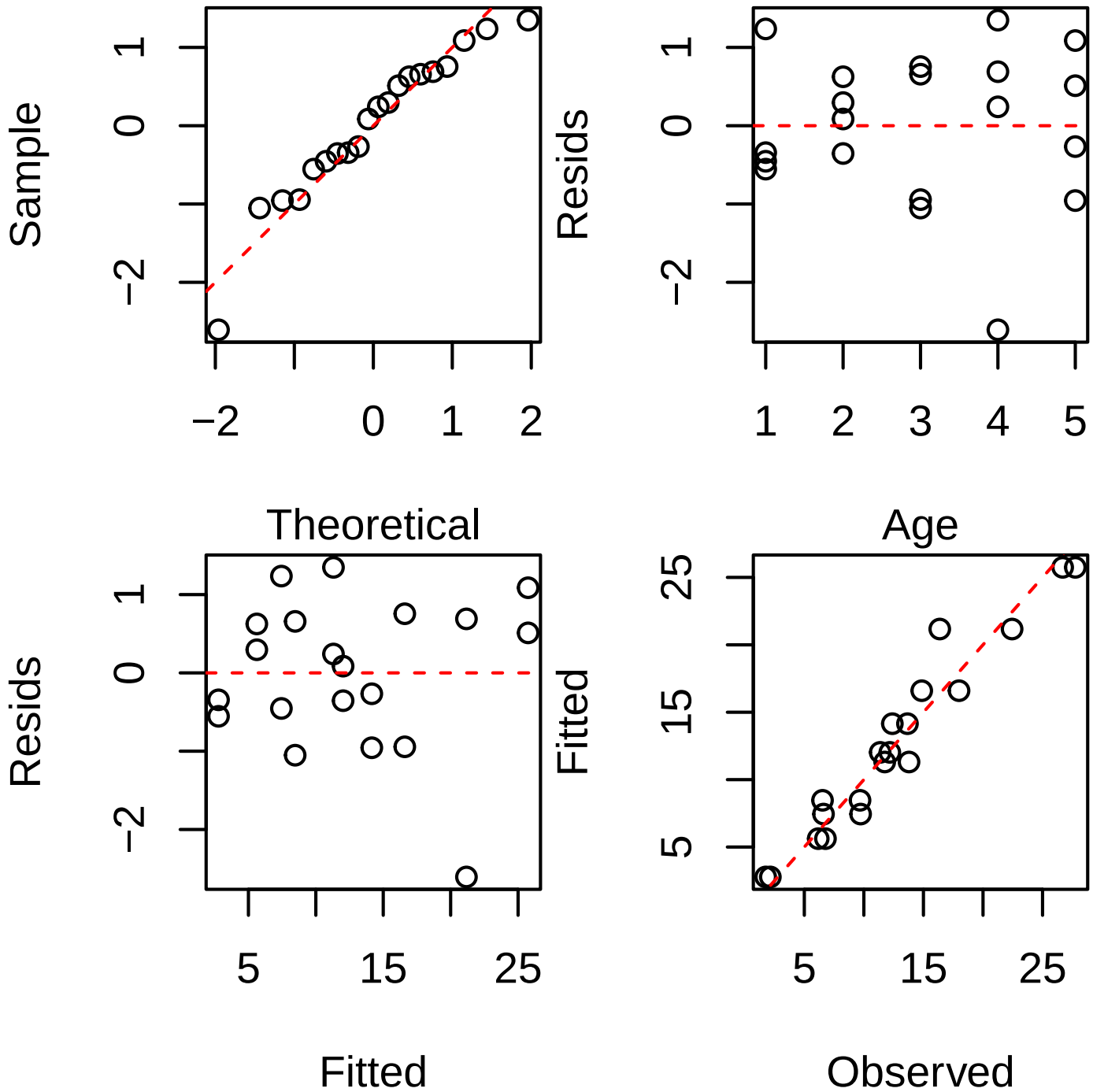
The code below forms residuals and then forms

- 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.

Linear Model: Residual analysis

```
residuals <- (liny - fitted)/sigmamed
par(mfrow = c(2, 2), mar = c(4, 4, 0.1, 0.1))
qqnorm(residuals, main = "", xlab = "Theoretical",
    ylab = "Sample")
abline(0, 1, lty = 2, col = "red")
plot(residuals ~ linxa, ylab = "Resids", xlab = "Age")
abline(h = 0, lty = 2, col = "red")
plot(residuals ~ fitted, ylab = "Resids", 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
Theoretical
Resids
Fitted
Resids
Age
Fitted
Observed

Case Control Example: Data

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

The case-control data are for the disease Leber Hereditary Optic Neuropathy (LHON) disease with genotype data for marker rs6767450:

	CC $x=0$	CT $x=1$	TT $x=2$	Total
Cases	6	8	75	89
Controls	10	66	163	239
Total	16	74	238	328

Let $x = 0, 1, 2$ represent the number of T alleles, and $p(x)$ the probability of being a case, given x copies of the T allele.

Case Control Example

For such case-control data one may fit the multiplicative odds model:

$$\frac{p(x)}{1-p(x)} = \exp(\alpha) \times \exp(\theta x),$$

with a binomial likelihood.

Interpretation:

– $\exp(\alpha)$ is of little interest given the case-control sampling.

– $\exp(\theta)$ is the odds ratio describing the multiplicative change in risk for one T allele versus zero T alleles.

– $\exp(2\theta)$ is the odds ratio describing the multiplicative change in risk for two T alleles versus zero T alleles.

– Odds ratios approximate the relative risk for a rare disease.

A Bayesian analysis adds a prior on α and θ.

Case contol example

```
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                # standard error^2
##        x
## 1.614044
exp(confint(logitmod))
##                   2.5 %    97.5 %
## (Intercept) 0.06293774 0.3801326
## x           1.01029266 2.7133859
```

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).

```
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.

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)
cc.mod$summary.fixed[, 1:5]
##                    mean        sd   0.025quant  0.5quant 0.975quant
## (Intercept) -1.8075742 0.4549020 -2.749656886 -1.790518 -0.9629009
## x            0.4802943 0.2502529  0.009944392  0.472736  0.9936563
```

Prior choice

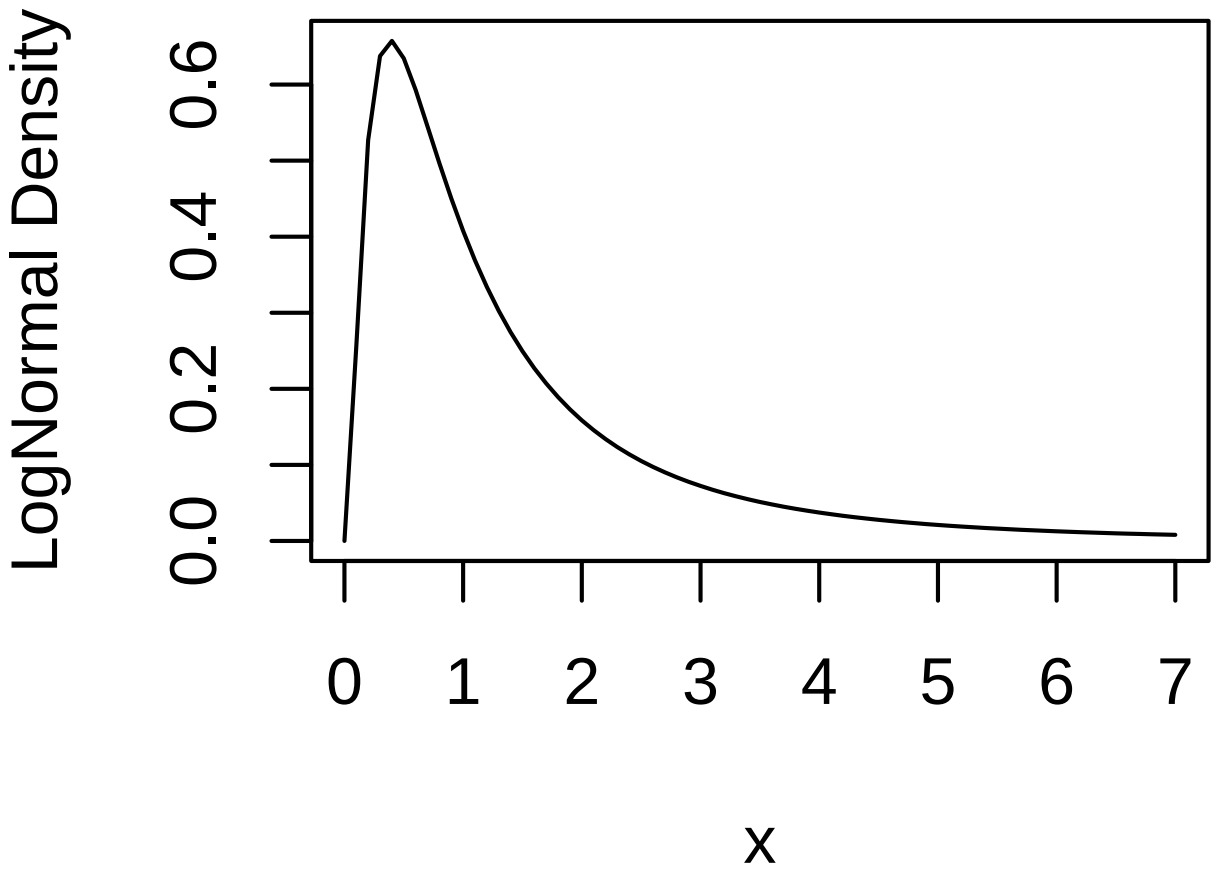
Suppose that for the odds ratio e^{β} we believe there is a 50% chance that the odds ratio is less than 1 and a 95% chance that it is less than 5; with $q_1 = 0.5, \theta_1 = 1.0$ and $q_2 = 0.95, \theta_2 = 5.0$, we obtain lognormal parameters $\mu = 0$ and $\sigma = (\log 5)/1.645 = 0.98$.

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)$sigma^2
cc.mod2 <- inla(y ~ x, family = "binomial", data = cc.dat,
    Ntrials = y + z, control.fixed = list(mean.intercept = c(0),
        prec.intercept = c(0.1), mean = c(0), prec = c(1/W)))
cc.mod2$summary.fixed[, 1:5]
##                    mean        sd  0.025quant   0.5quant 0.975quant
## (Intercept) -1.3227216 0.2895789 -1.90129454 -1.3192537 -0.7639026
## x            0.1985648 0.1536026 -0.09997018  0.1975346  0.5026117
```

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

Case-Control Example: Stan Analysis

Analysis with default priors: uses code in file `LogisticExample.stan`

```
/*
 * Logistic regresssion example
 */
data {
    int y[3];
    int n[3];
    int x[3];
}
parameters {
    real beta0;
    real beta1;
}
model {
for (i in 1:3)
    y[i] ~ binomial(n[i],inv_logit(beta0+beta1*x[i]));
}
```

Case-Control Example: Stan Analysis

```
library(rstan)
stanlogist <- stan("LogisticExample.stan",
    data = list(x = c(0, 1, 2), y = c(6,
        8, 75), n = c(16, 74, 238)), iter = 1000,
    chains = 3, seed = 1234)
```

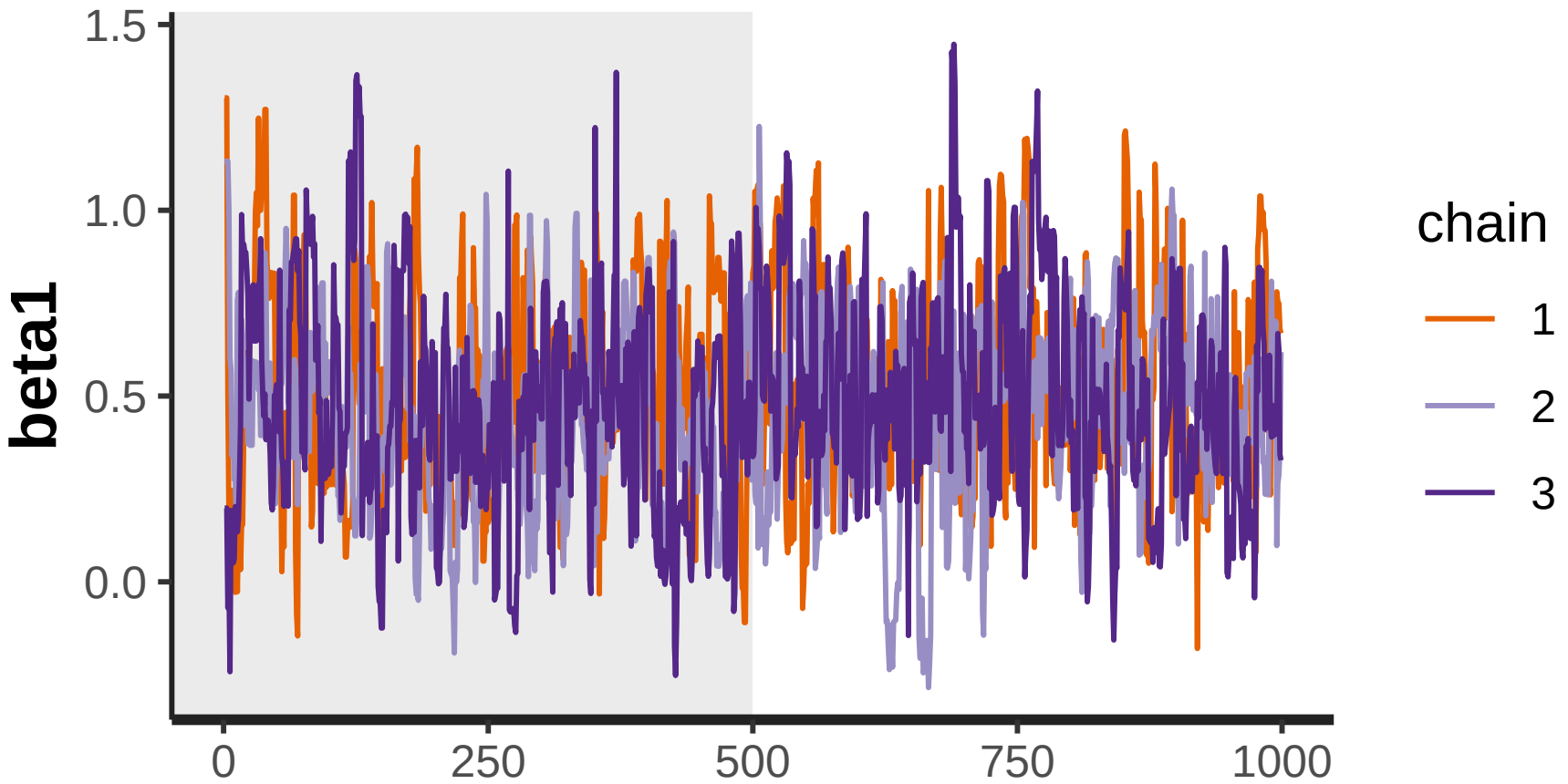
Case-Control Example: Stan Analysis

Close agreement with INLA analysis

```
summary(stanlogist)$summary
##                mean    se_mean        sd          2.5%          25%          50%
## beta0   -1.8666721 0.03811829 0.4953491 -2.874139e+00   -2.2106786   -1.8396933
## beta1    0.5064828 0.01977361 0.2706508 -1.096452e-03    0.3218775    0.4907956
## lp__  -190.8098050 0.06134907 1.1260321 -1.939657e+02 -191.2131638 -190.4401877
##                75%        97.5%    n_eff     Rhat
## beta0   -1.532068   -0.9416814 168.8714 1.022386
## beta1    0.687265    1.0522181 187.3470 1.020515
## lp__  -190.026376 -189.7647712 336.8880 1.008563
```

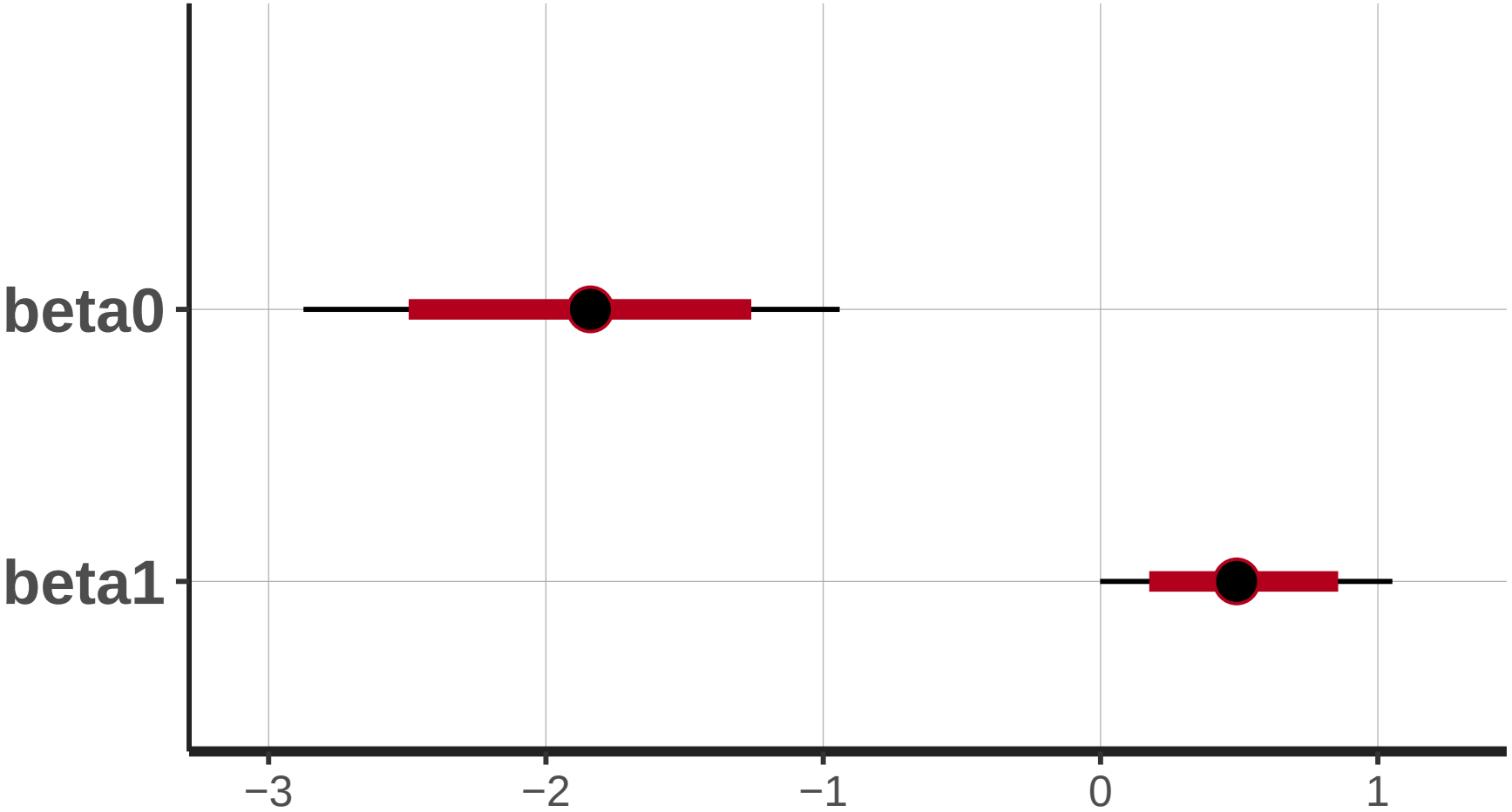
Case-Control Example: Stan Analysis

```
traceplot(stanlogist, pars = c("beta1"), inc_warmup = TRUE)
```



Case-Control Example: Stan Analysis

```
plot(stanlogist, color = "green")
```



Case-Control Example: Stan Analysis

Analysis with informative prior: `LogisticExamplePriors.stan`

```
data {
    int y[3];
    int n[3];
    int x[3];
}
parameters {
    real beta0;
    real beta1;
}
transformed parameters {
    real<lower=0> theta;
    theta = exp(beta1);
}
model {
beta0 ~ normal(0,3.162278);
beta1 ~ normal(0,0.2068738);
for (i in 1:3)
    y[i] ~ binomial(n[i],inv_logit(beta0+beta1*x[i]));
}
```

Case-Control Example

Stan Analysis with Informative Prior

```
library(rstan)
stanlogist2 <- stan("LogisticExamplePriors.stan",
    data = list(x = c(0, 1, 2), y = c(6,
        8, 75), n = c(16, 74, 238)), iter = 1000,
    chains = 3, seed = 2345)
```

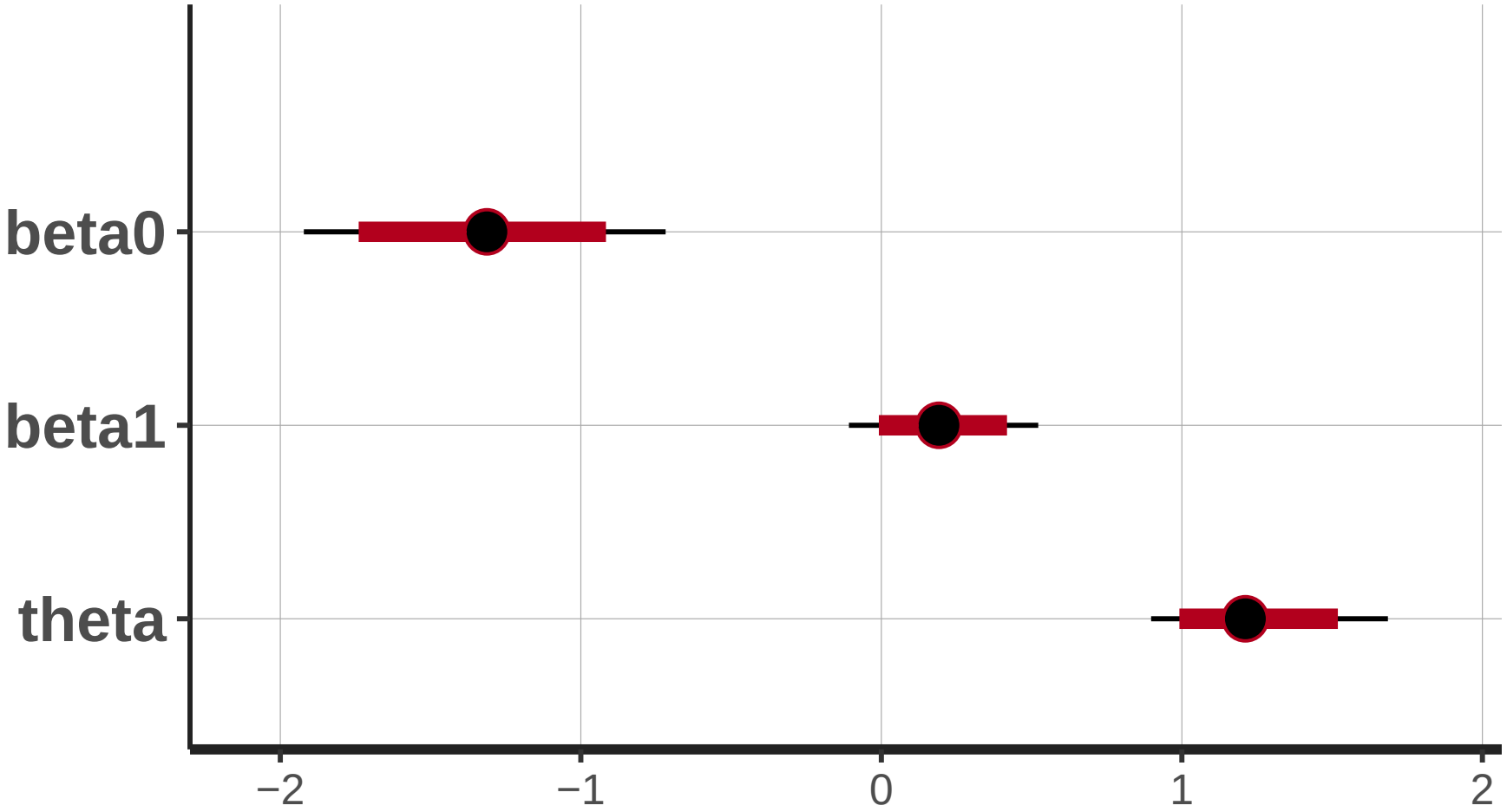
Case-Control Example: Stan Analysis with Informative Prior

Again close agreement with INLA analysis

```
summary(stanlogist2)$summary
##               mean     se_mean        sd        2.5%         25%         50%
## beta0   -1.322365 0.018641945 0.3080748   -1.9216211   -1.53105113   -1.3128418
## beta1    0.197279 0.009847078 0.1639286   -0.1078181    0.07897217    0.1917935
## theta    1.234691 0.012286508 0.2060639    0.8977912    1.08217423    1.2114203
## lp__  -192.040846 0.049638152 1.0594150 -194.7192545 -192.51606325 -191.7040471
##                75%        97.5%    n_eff     Rhat
## beta0   -1.1079035   -0.7180333 273.1052 1.006858
## beta1    0.3078097    0.5220129 277.1370 1.006086
## theta    1.3604421    1.6854170 281.2851 1.006068
## lp__  -191.2540306 -190.9664644 455.5133 1.000624
```

Case-Control Example: Stan Analysis with Informative Prior

```
plot(stanlogist2, color = "green", parameter = "theta")
```



Case-Control Example: Stan Analysis with Informative Prior

```
stan_dens(stanlogist2)
```