HW2

1 (a).

My stan model:

```
data {
  int<lower=0> N;
  int<lower=0> G;
  int<lower=0> C;
  vector[N] post;
  vector[N] treatment;
  vector[N] pre;
  int grade[N];
  vector[N] city;
}
parameters {
  vector[G] a;
  vector[G] b;
  vector[G] c;
  vector[G] d;
  real mu_a;
  real mu_b;
  real mu_c;
  real mu_d;
  real<lower=0,upper=100> sigma_a;
  real<lower=0,upper=100> sigma_b;
  real<lower=0,upper=100> sigma_c;
  real<lower=0,upper=100> sigma_d;
  real<lower=0,upper=100> sigma_y[G];
}
transformed parameters {
  vector[N] y_hat;
  vector[N] sigma_y_hat;

  for (i in 1:N) {
    y_hat[i] <- a[grade[i]] + b[grade[i]] * treatment[i] + c[grade[i]] * pre[i] + d[grade[i]] * city[i]
    sigma_y_hat[i] <- sigma_y[grade[i]];
  }
}
model {
  a ~ normal(mu_a, sigma_a);
  b ~ normal(mu_b, sigma_b);
  c ~ normal(mu_c, sigma_c);
  d ~ normal(mu_d, sigma_d);
  post ~ normal(y_hat, sigma_y_hat);
}
```

And code for running it:

```
library ("rstan")
setwd("~/Documents/BDA/Homework 2")
```

```
electric_data <- read.table ("electric_data.txt", header=TRUE)
city <- c(electric_data$City, electric_data$City)
grade <- as.integer(c(electric_data$Grade, electric_data$Grade))
pre <- c(electric_data$T_Pre, electric_data$C_Pre)
post <- c(electric_data$T_Post, electric_data$C_Post)
control <- array(0, length(electric_data$C_Pre))
treat <- array(1, length(electric_data$T_Pre))
treatment <- c(treat, control)
N <- length(pre)
G <- 4
C <- 2

fit <- stan("electric_model.stan")
```

Treatment effect by grade and average

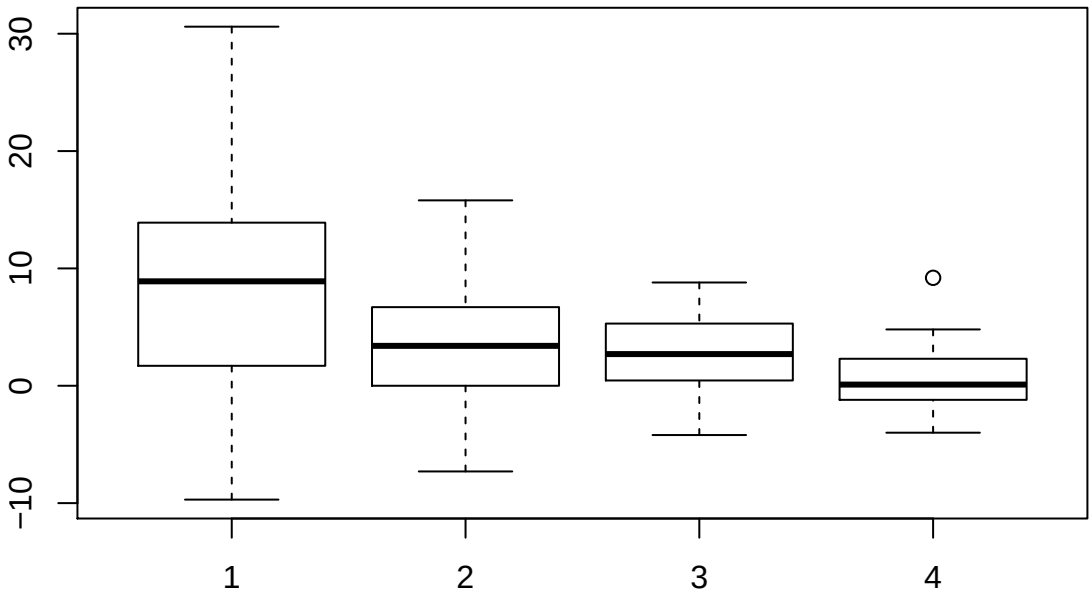
```
print(fit, pars = c("b", "mu_b"))
```

```
## Inference for Stan model: electric_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean   sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
## b[1] 5.38    0.35 2.76  1.47 3.15 4.91 7.19 11.63    63 1.06
## b[2] 4.05    0.24 1.35  1.55 3.02 4.06 5.22  6.39    31 1.11
## b[3] 1.92    0.15 0.81  0.54 1.24 1.94 2.46  3.51    28 1.14
## b[4] 1.74    0.01 0.64  0.38 1.39 1.71 2.13  3.03  2475 1.00
## mu_b 3.30    0.07 2.52 -1.28 2.25 3.14 4.10  9.07  1254 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Sep 25 09:50:36 2015.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

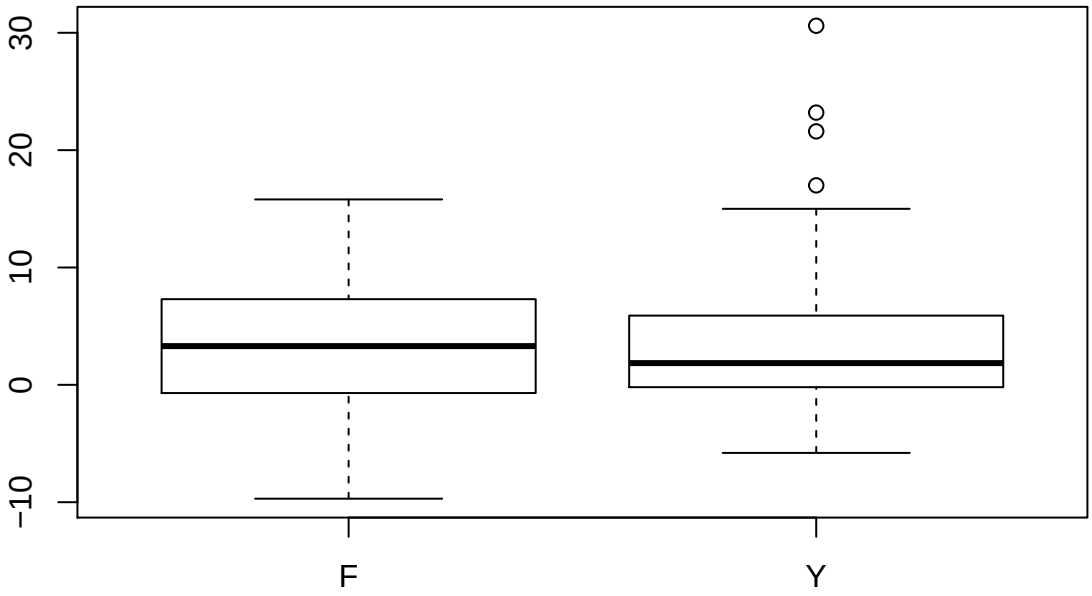
(b) If we look at point estimates of the treatment effect, we see differences by grade but not city

```
tot <- (electric_data$T_Post - electric_data$T_Pre)
baseline <- (electric_data$C_Post - electric_data$C_Pre)
electric_data$Treatment_eff <- tot - baseline

#grade
grade1 <- subset(electric_data, electric_data$Grade == 1)
grade2 <- subset(electric_data, electric_data$Grade == 2)
grade3 <- subset(electric_data, electric_data$Grade == 3)
grade4 <- subset(electric_data, electric_data$Grade == 4)
boxplot(grade1$Treatment_eff, grade2$Treatment_eff, grade3$Treatment_eff, grade4$Treatment_eff, names =
```




```
#city
fresno <- subset(electric_data, as.numeric(electric_data$City) == 1)
youngstown <- subset(electric_data, as.numeric(electric_data$City) == 2)
boxplot(fresno$Treatment_eff, youngstown$Treatment_eff, names = c("F", "Y"))
```



However this might be due to the lower variance as the grades get higher. The first graders have low variance on the pretest compared to the 4th graders because they have very low scores:

```
var(grade1$T_Pre) + var(grade1$C_Pre)
```

```
## [1] 11.84305
```

```
var(grade4$T_Pre) + var(grade4$C_Pre)
```

```
## [1] 143.0665
```

And the opposite is true at the upper end

```
var(grade1$T_Post) + var(grade1$C_Post)
```

```
## [1] 448.6678
```

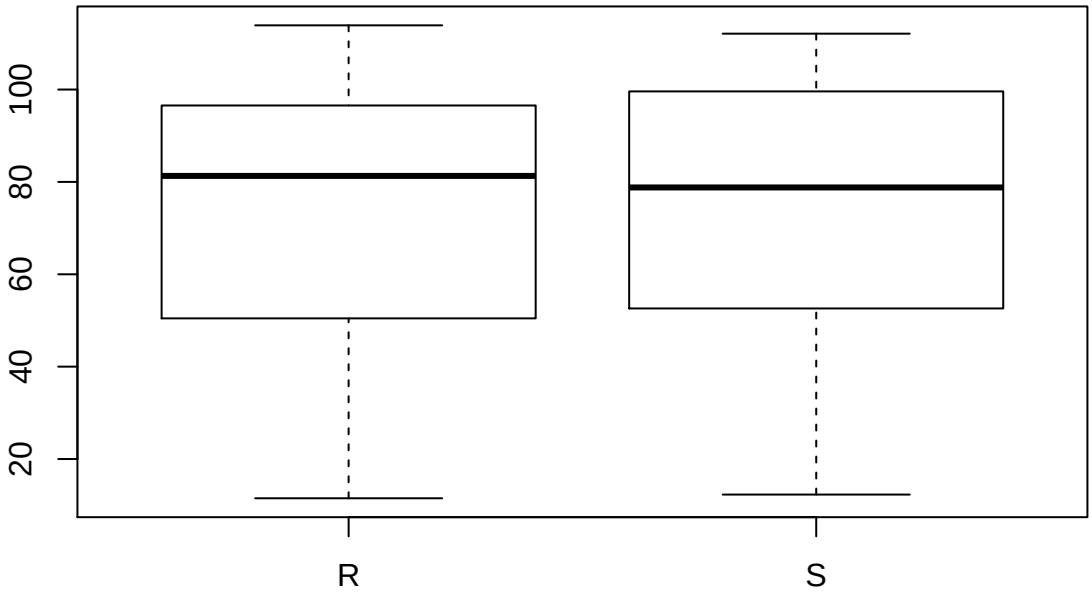
```
var(grade4$T_Post) + var(grade4$C_Post)
```

```
## [1] 70.8319
```

This is beacuse grades are bounded in 0-100 so a linear model for grades is probably not the right model

(c) The biggest concern is that they were selecting on how bad the class was at the beginning. This does not seem to be the case:

```
R <- subset(electric_data, as.numeric(electric_data$Replacement) == 1)
S <- subset(electric_data, as.numeric(electric_data$Replacement) == 2)
boxplot(R$C_Pre, S$C_Pre, names = c("R", "S"))
```



In addition, we can regress everything and get no real coefficients

```
replacement <- electric_data$Replacement
city <- electric_data$City
grade <- electric_data$Grade
pre_test <- electric_data$T_Pre
lm(replacement ~ city + grade + pre_test)
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a
## factor response will be ignored
```

```
## Warning in Ops.factor(y, z$residuals): - not meaningful for factors
```

```
##
## Call:
## lm(formula = replacement ~ city + grade + pre_test)
##
## Coefficients:
## (Intercept)        cityY        grade     pre_test
##   1.6541283    0.0114392   -0.0015031   -0.0002871
```

2. Fitting model

```
data {
  int J;
  int n[J];
  real x[J];
  int y[J];
  vector[2] mu;
  cov_matrix[2] Sigma;
}
parameters {
  vector[2] pars;
}
transformed parameters {
  real alpha;
  real beta;
  real theta[J];
  alpha <- pars[1];
  beta <- pars[2];
  for (j in 1:J)
    theta[j] <- inv_logit(alpha + beta * x[j]);
}
model {
  pars ~ multi_normal(mu, Sigma);
  y ~ binomial(n, theta);
}
generated quantities {
  real LD50;
  LD50 <- -alpha/beta;
}
```

```
setwd("~/Documents/BDA/Homework 2")
bioassay <- read.table("bioassay_data.txt", header=TRUE)
x <- bioassay$x
y <- bioassay$y
n <- bioassay$n
J <- length(x)

sd_a <- 4
sd_b <- 10
var_a <- sd_a * sd_a
var_b <- sd_b * sd_b
corr <- .5
cov <- corr * sd_a * sd_b
Sigma <- matrix(c(var_a,cov,
```

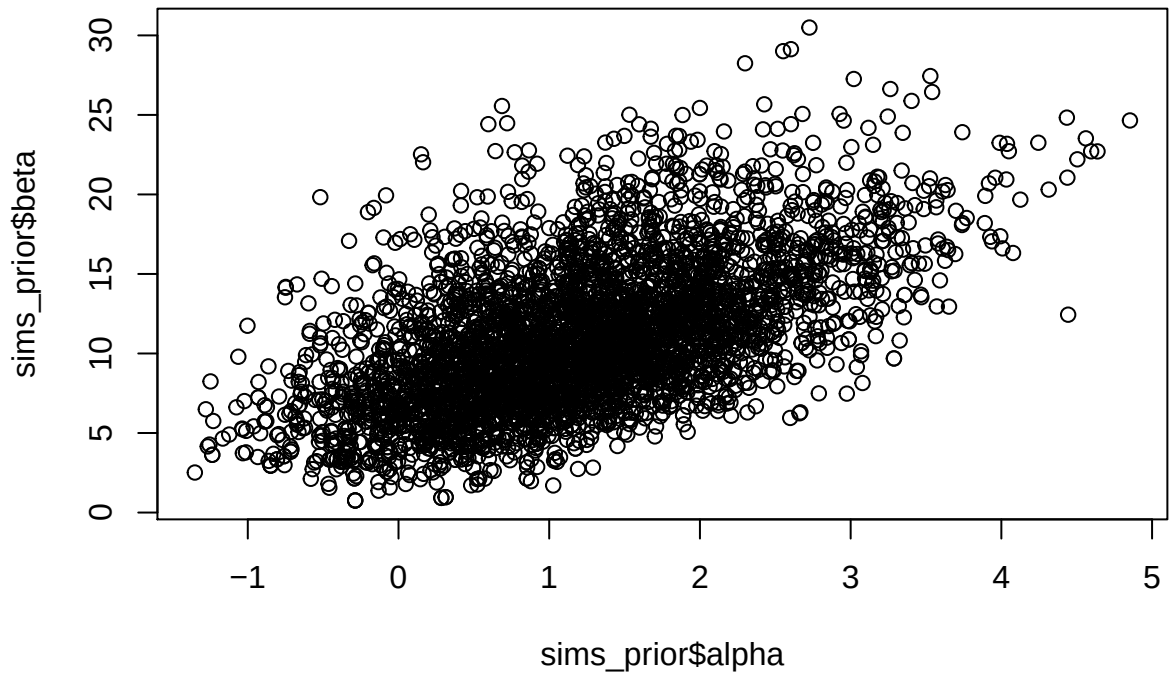
```
                  cov,var_b),nrow=2)
mu <- c(0, 10)

fit_prior <- stan("bioassay_model_prior.stan")
```

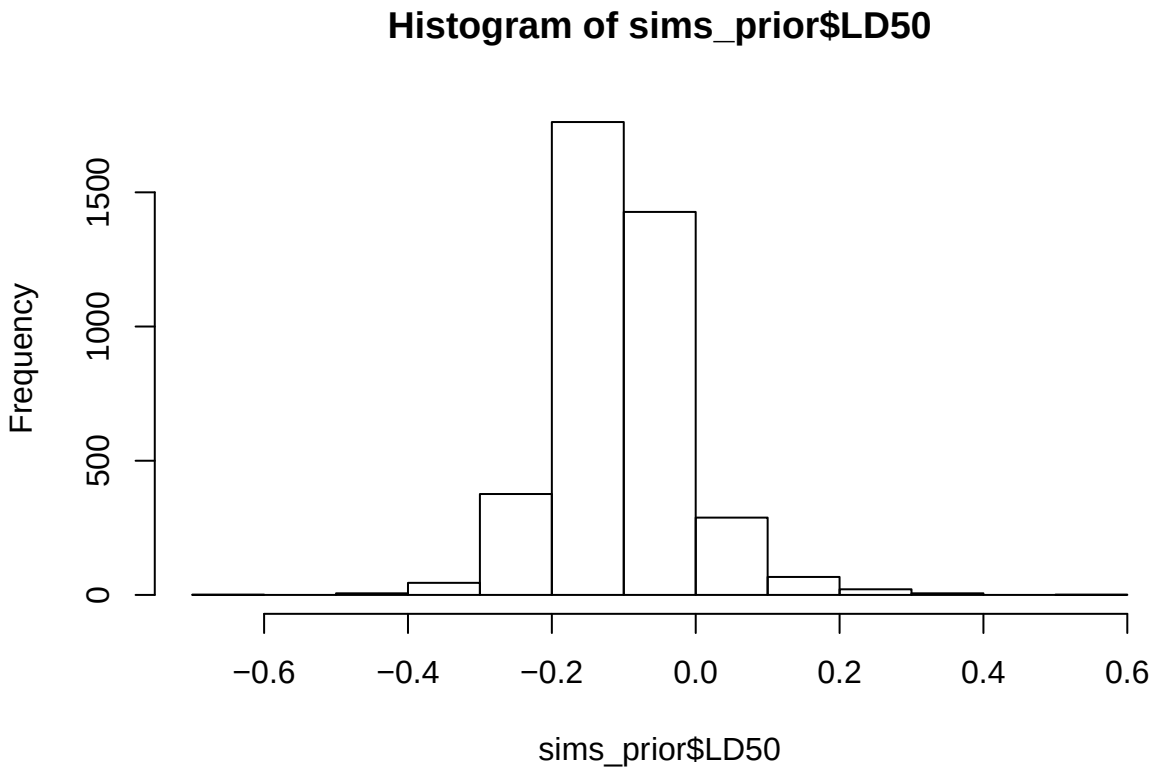
```
print(fit_prior, pars = c("alpha", "beta", "LD50"))
```

```
## Inference for Stan model: bioassay_model_prior.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##        mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## alpha  1.19    0.03 0.94 -0.52  0.52  1.16  1.80  3.20   906 1.00
## beta  10.86    0.16 4.55  3.43  7.50 10.34 13.60 21.13   815 1.01
## LD50  -0.10    0.00 0.09 -0.27 -0.16 -0.11 -0.06  0.10  1406 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Sep 25 09:51:16 2015.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
## Post-processing
sims_prior <- extract(fit_prior)
plot(sims_prior$alpha, sims_prior$beta)
```




```
hist(sims_prior$LD50)
```



(b) Compared to the model with the noninformative prior, the estimates are pulled in slightly towards the prior

```
data {
  int J;
  int n[J];
  real x[J];
  int y[J];
}
parameters {
  real alpha;
  real beta;
}
transformed parameters {
  real theta[J];
  for (j in 1:J)
    theta[j] <- inv_logit(alpha + beta * x[j]);
}
model {
  y ~ binomial(n, theta);
}
generated quantities {
  real LD50;
  LD50 <- -alpha/beta;
}
```

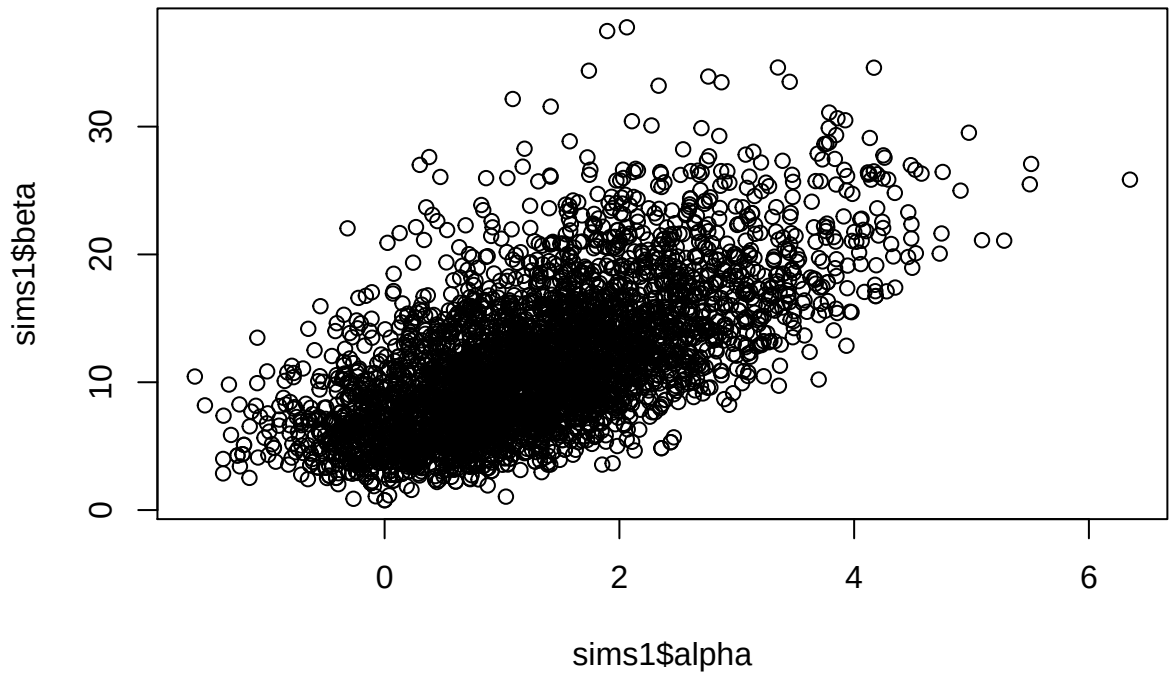
```
fit1 <- stan("bioassay_model.stan")
```

```
print(fit1, pars = c("alpha", "beta", "LD50"))
```

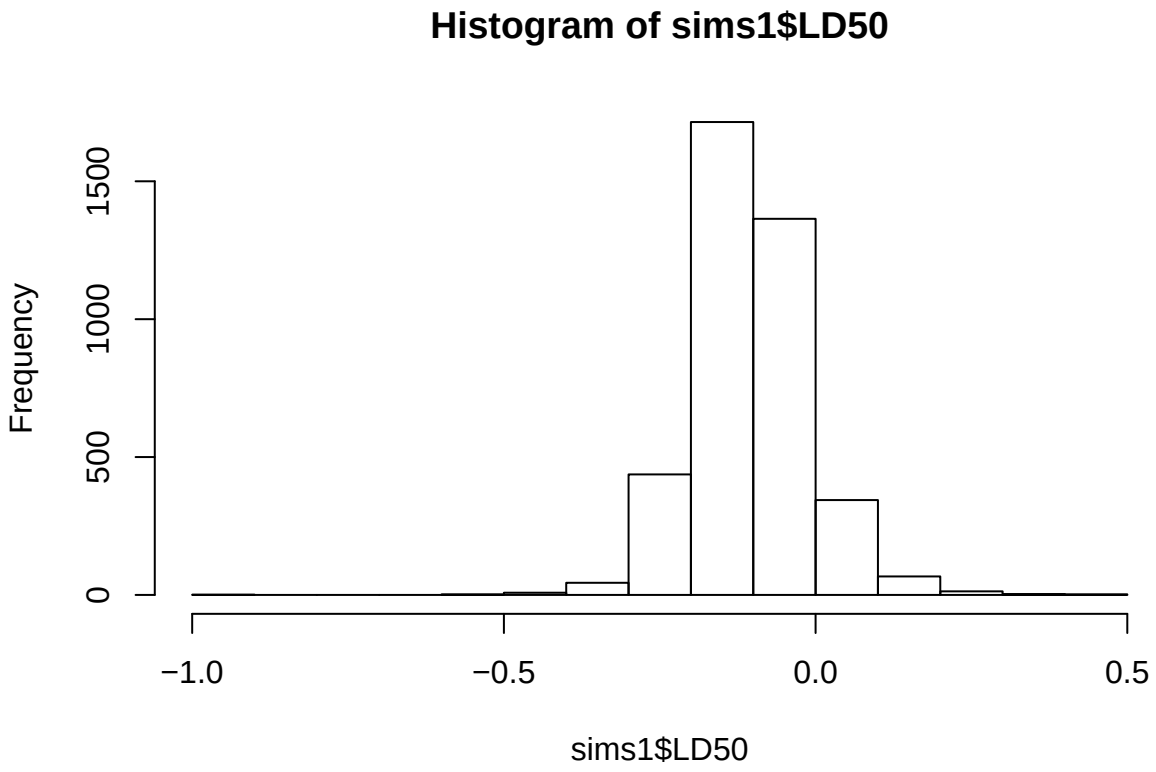
```
## Inference for Stan model: bioassay_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##        mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## alpha  1.26    0.04 1.06 -0.51  0.53  1.19  1.90  3.70   902    1
## beta  11.33    0.20 5.57  3.40  7.21 10.45 14.45 25.00   785    1
## LD50  -0.11    0.00 0.09 -0.27 -0.16 -0.11 -0.06  0.09  1194    1
##
## Samples were drawn using NUTS(diag_e) at Fri Sep 25 09:51:50 2015.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
## Post-processing

sims1 <- extract(fit1)
plot(sims1$alpha, sims1$beta)
```




```
hist(sims1$LD50)
```


Histogram of sims1$LD50
Frequency
0
500
1000
1500
−1.0
−0.5
0.0
0.5
sims1$LD50