# 6.2 probit regression

*Chris Parrish*

*July 3, 2016*

## Contents



6.2 probit regressionn

reference:
- ARM chapter 06, github

```
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
library(ggplot2)
```

## Poisson

### data

```
source("wells.data.R", echo = TRUE)
```

```
##
## > "switc" <- c(1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
## +     1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
## +     1, 0, 0, 1, 0, 1, 1, 0, 1 .... [TRUNCATED]
##
## > "arsenic" <- c(2.36, 0.71, 2.07, 1.15, 1.1, 3.9, 2.97,
## +     3.24, 3.28, 2.52, 3.13, 3.04, 2.91, 3.21, 1.7, 1.8, 1.44,
## +     1.43, 2.33, 2.83, 1.7 .... [TRUNCATED]
##
## > "dist" <- c(16.826000213623, 47.3219985961914, 20.9669990539551,
## +     21.4860000610352, 40.8740005493164, 69.5179977416992, 80.7109985351563,
## +   .... [TRUNCATED]
##
## > "assoc" <- c(0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1,
## +     1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
## +     1, 0, 0, 1, 1, 0, 1, 0, 0 .... [TRUNCATED]
##
## > "educ" <- c(0, 0, 10, 12, 14, 9, 4, 10, 0, 0, 5, 0,
## +     0, 0, 0, 7, 7, 7, 0, 10, 7, 0, 5, 0, 8, 8, 10, 16, 10, 10,
## +     10, 10, 0, 0, 0, 3, 0,  .... [TRUNCATED]
##
## > "N" <- 3020
```

## model

### wells_probit.stan

```
data {
  int<lower=0> N;
  vector[N] dist;
  int<lower=0,upper=1> switc[N];
}
transformed data {
  vector[N] dist100;

  dist100 = dist / 100.0;
}
parameters {
  vector[2] beta;
}
model {
  for (n in 1:N)
    switc[n] ~ bernoulli(Phi(beta[1] + beta[2] * dist100[n]));
}
```

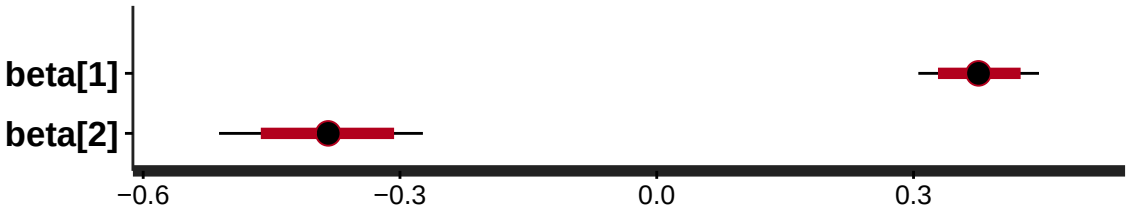
## fit

```
## Probit or logit (wells_probit.stan)
## glm (switch ~ dist100, family=binomial(link="probit"))
dataList <- c("N","switc","dist")
wells_probit.sf1 <- stan(file='wells_probit.stan', data=dataList,
                         iter=1000, chains=4)
```

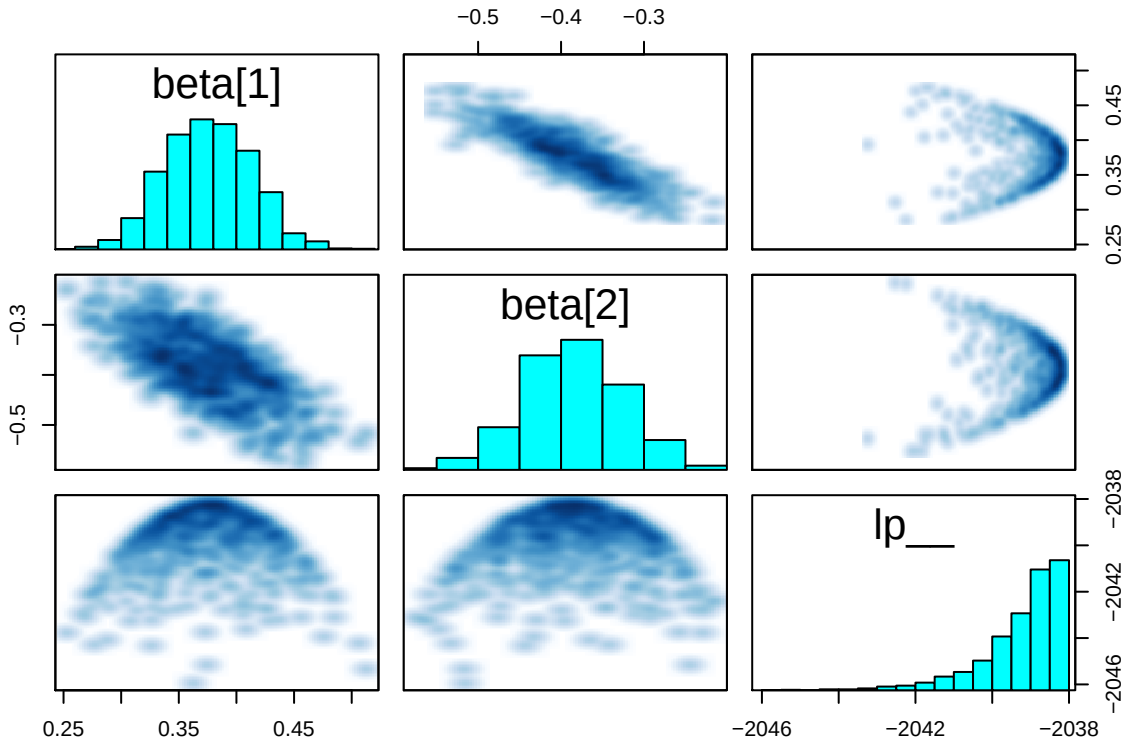
```
plot(wells_probit.sf1)
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```




```
pairs(wells_probit.sf1)
```




```
print(wells_probit.sf1)
```

```
## Inference for Stan model: wells_probit.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##             mean se_mean   sd     2.5%      25%      50%      75%    97.5%
## beta[1]     0.38    0.00 0.04     0.31     0.35     0.38     0.40     0.45
## beta[2]    -0.38    0.00 0.06    -0.51    -0.42    -0.38    -0.35    -0.27
## lp__    -2039.22    0.03 1.05 -2042.09 -2039.65 -2038.90 -2038.46 -2038.18
##         n_eff Rhat
## beta[1]   673    1
## beta[2]   689    1
## lp__     1039    1
##
## Samples were drawn using NUTS(diag_e) at Wed Jul  6 02:24:54 2016.
## 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).
##  The estimated Bayesian Fraction of Missing Information is a measure of
##  the efficiency of the sampler with values close to 1 being ideal.
##  For each chain, these estimates are
##  0.9 0.9 0.9 0.9
```

**figure 6.2**

```
 # Figure 6.2
p1 <- ggplot(data.frame(x = c(0, 2)), aes(x)) +
      stat_function(geom="line", fun=dnorm, args=list(mean=0, sd=1.6),
```

```
                    color = "darkred") +
      scale_x_continuous(limits=c(-6,6)) +
      theme_gray() +
      ggtitle("Normal(mean=0, sd=1.6)")
print(p1)
```