

RSTAN - Logistic Regression

Bernoulli Logistic

Load the packages

```
library(rstan)
```

Model

$$Y \sim \text{Bernoulli}(p)$$

$$\text{logit}E[Y] = \beta * X + \alpha$$

Data

survival package 안에 있는 colon 데이터 사용.

```
library(survival)

##data
str(colon)
colon <- na.omit(colon)
colon[, -1]
selected_cols <- c('status', 'sex', 'age', 'obstruct',
  'perfor', 'adhere', 'nodes', 'differ', 'extent', 'surg')
colon <- colon[, selected_cols]

str(colon)
colon$differ <- as.ordered(colon$differ)
colon$extent <- as.ordered(colon$extent)

design_matrix <- model.matrix(status ~., data = colon)

##list data
data1 <- list(
  N = nrow(design_matrix),
  x = design_matrix[, -1],
  K = ncol(design_matrix) - 1,
  y = colon$status
)
```

STAN Code

```
data {  
  int<lower=0> N; // number of obs  
  int<lower=0> K; // number of predictors  
  matrix[N, K] x; // predictor design matrix  
  int y[N]; // outcome //different from normal or linear regression.  
}  
  
parameters {  
  real alpha; // intercept  
  vector[K] beta; // coefficients for predictors  
}  
  
transformed parameters{ // we need this step for bernoulli_logit  
  vector[N] eta;  
  eta = alpha + x * beta;  
}  
  
model {  
  y ~ bernoulli_logit(eta) ; // likelihood  
}
```

Fitting

```
fit1 <- stan(file = 'logistic_reg.stan', data = data1)
```

Trace Plot

```
traceplot(fit1, pars = 'alpha', inc_warmup = TRUE)  
traceplot(fit1, pars = 'beta[1]', inc_warmup = TRUE)
```

Estimated Parameters

```
plot(fit1, pars = 'alpha', show_density=TRUE)  
plot(fit1, pars = 'beta[1]', show_density = TRUE)  
plot(fit1, pars = 'beta[2]', show_density = TRUE)
```

Multinomial Logistic Regression

Model

종속변수 Y 가 범주 j 에 속할 확률을 $P(Y=j)$ 라고 하자.

$$P(Y = j) = \frac{e^{\alpha_j + \beta_j x}}{1 + \sum_{j=1}^{J-1} e^{\alpha_j + \beta_j x}}$$

기준 범주를 J 라고 할 때

$$P(Y = j) = \frac{1}{1 + \sum_{j=1}^{J-1} e^{\alpha_j + \beta_j x}}$$

Data

출처: <https://stats.idre.ucla.edu/stat/data/>

```
##data
library(foreign)
m1 <- read.dta("https://stats.idre.ucla.edu/stat/data/hsbdemo.dta")

mldata <- m1[,c('prog', 'ses', 'write')]
str(mldata)

mldata$ses <- as.ordered(mldata$ses)
str(mldata)

mldata$prog <- as.numeric(mldata$prog)
design_matrix2 <- model.matrix(prog ~ ., data = mldata)

##list data
data2 <- list(
  K = length(unique(mldata$prog)),
  N = nrow(design_matrix2),
  D = ncol(design_matrix2),
  y = mldata$prog,
  x = design_matrix2
)
```

STAN Code

```
data {
  int K; // possible outcomes
  int N; // number of obs
  int D; // number of predictors(including intercept)
  int y[N];
  matrix[N, D] x;
}

parameters {
  matrix[D, K] beta;
}
```

```

model {
  matrix[N,K] x_beta = x * beta;

  to_vector(beta) ~ normal(0,5); //prior for beta

  //As of Stan 2.18, the categorical-logit distribution is not vectorized for
  parameter arguments,
  //so the loop is required.
  for (n in 1:N)
    y[n] ~ categorical_logit(x_beta[n]'); // x_beta[n]'is the transpose of
    x_beta[n]
}

```

Fitting

```

fit2 <- stan(file = 'multi_logit.stan', data = data2)
fit2

```

Trace Plot

```

traceplot(fit2, inc_warmup = TRUE)

```

Estimated Parameters

```

plot(fit2, pars = 'beta[1,1]', show_density=TRUE)
plot(fit2, pars = 'beta[1,2]', show_density = TRUE)
plot(fit2, pars = 'beta[4,3]', show_density = TRUE)

```

Summary

```

summary(fit2, par = 'beta', prob = 0.5)$summary

```