A simple Weibull model in Stan

Leah Comment

10/26/2017

```
# Packages
suppressPackageStartupMessages(library("rstan"))
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores() - 1)
```

```
# Read in Weibull simulated data
dat_wei <- readRDS("dat_wei.Rdata")

# Package for Stan
stan_dat <-  list(N = nrow(dat_wei),
                  P_1 = 2,
                  X_1 = cbind(1, dat_wei$x_c),
                  Y = dat_wei$Y,
                  dY= dat_wei$delta_Y)
```

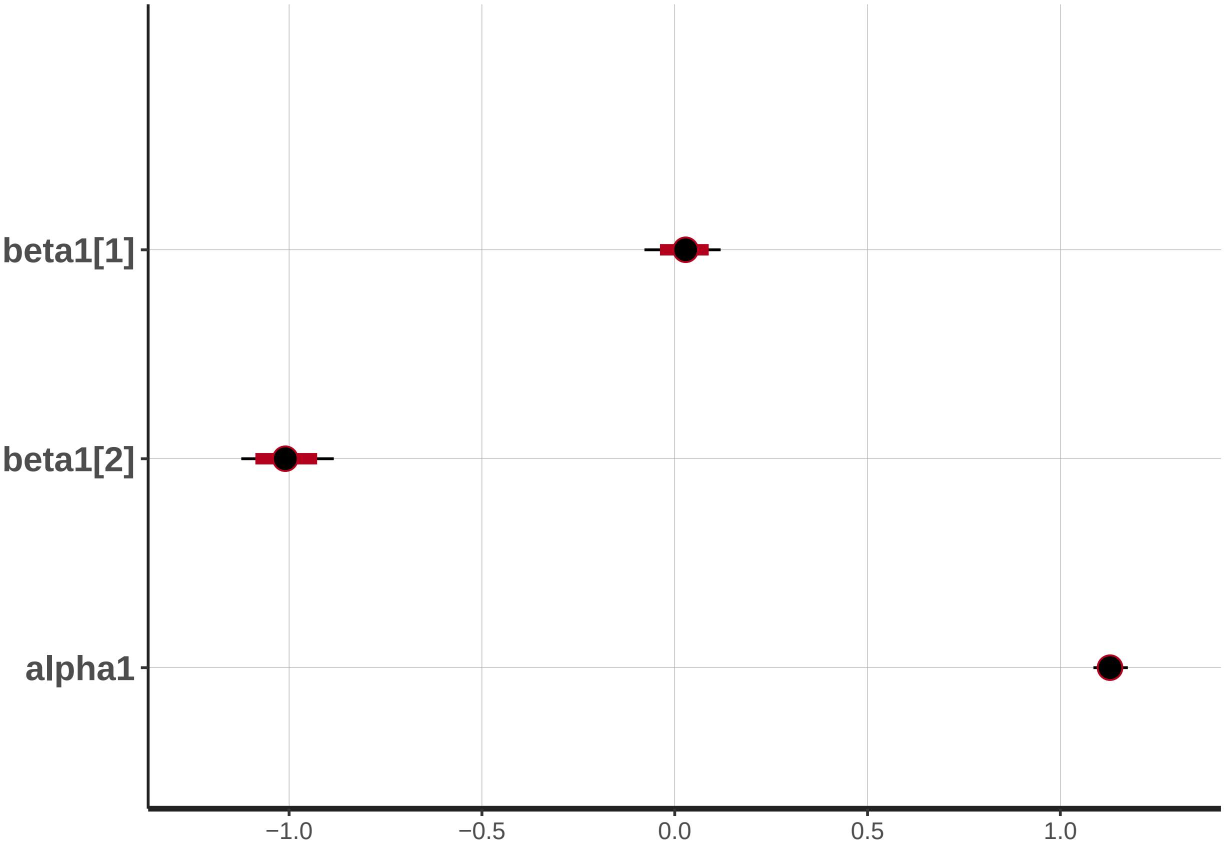
The Stan code to fit a simple Weibull regression model is shown below.

```
data {
  // number of observations
  int<lower=0> N;
  // number of columns in design matrix, including intercept
  int<lower=1> P_1;
  // design matrix
  matrix[N, P_1] X_1;
  // observed event or censoring time
  real<lower=0> Y[N];
  // indicator of event observation
  int<lower=0,upper=1> dY[N];
}

parameters {
  // vector of regression parameters
  vector[P_1] beta1;

  // shape parameters (the one in exponent of time)
  // alpha > 1 -> hazard increases over time, more clumping
  real<lower=0> alpha1;
}

model {
  // linear predictors
  vector[N] lp1;
  lp1 = X_1 * beta1;

  // likelihood
  for (n in 1:N){
    if (dY[n] == 1) {
       Y[n] ~ weibull(alpha1, exp(-(lp1[n])/alpha1));
    } else {
      target += weibull_lccdf(Y[n] | alpha1, exp(-(lp1[n])/alpha1));
    }
  }

}
```

```
# Fit Weibull model
fit1 <- stan(file = "simple_weibull.stan", data = stan_dat, iter = 1000, chains = 4)
b_ests <- summary(fit1, params = c("beta1"))[["summary"]][1:3,"50%"]
plot(fit1)
```

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

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



```
# Compare to frequentist
library("survival")
fit2 <- survreg(Surv(Y, delta_Y) ~ x_c, data = dat_wei, dist = "weibull")
print(fit2)
```

```
## Call:
## survreg(formula = Surv(Y, delta_Y) ~ x_c, data = dat_wei, dist = "weibull")
##
## Coefficients:
## (Intercept)         x_c
## -0.02312934  0.89185366
##
```

```
## Scale= 0.8861882
##
## Loglik(model)= -2170.7   Loglik(intercept only)= -2289.9
##  Chisq= 238.43 on 1 degrees of freedom, p= 0
## n= 1500
```

```
f_ests <- c(-coef(fit2)/fit2$scale, 1/fit2$scale)

# Side-by side comparison of model results
library("knitr")
kable(data.frame(Parameter = c("beta1", "beta2", "alpha"),
                 Bayesian = b_ests, Frequentist = f_ests),
      row.names = FALSE)
```

Parameter	Bayesian	Frequentist
beta1	0.0282756	0.0260998
beta2	-1.0096705	-1.0063930
alpha	1.1288041	1.1284284