# Rigdon Basu (1989) Time truncated data example

replication in Stan

*Miao Cai miao.cai@slu.edu*

*1/29/2019*

## Time truncated data - 115 kV Transmission line example

This data is presented by Martz (1975). It gives the failure times, or interrpution times, of the 115 kV transmission circuit from Cunningham Generating Station, located near Hobbs, New Mexico, to Eddy County Interchange, located near Artesia, New Mexico. We assume that data collection was terminated on December 31, 1971.

Data collected in this manner, with testing terminated at a predetermined time, are called time truncated. It is important to distinguish between these approches to data collection because statistical inference precedures are different for the two situations.

```
library(rstan)
```

```
## Warning: package 'rstan' was built under R version 3.4.4
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.4.4
## Loading required package: StanHeaders
## Warning: package 'StanHeaders' was built under R version 3.4.4
## rstan (Version 2.17.3, GitRev: 2e1f913d3ca3)
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
t = c(0.129, 0.151, 0.762, 0.869, 2.937, 3.077, 3.841, 3.964, 4.802, 4.898, 7.868, 8.430)
trunc_time = 8.463

datstan = list(
  n = length(t),
  tau = trunc_time,
  t = t)
```

MLE estimates of $\beta$ and *theta* are:

$$\hat{\beta} = \frac{12}{\sum_{i=1}^{12} \ln(8.463/t_i)} = 0.678 \hat{\theta} = \frac{8.463}{12^{1/0.678}} = 0.217$$

### Diffuse priors for beta and theta - GAMMMA(1, 1)

The log likelihood function $l$ for time truncated non-homogeneous Poisson process is:

$$
\begin{aligned}
l &= \log\Big(\prod_{i=1}^{n}\frac{\beta}{\theta}(\frac{t_i}{\theta})^{\beta-1}\Big)e^{-(\tau/\theta)^{\beta}} \\
&= \sum_{i=1}^{n}\log\Big(\frac{\beta}{\theta}(\frac{t_i}{\theta})^{\beta-1}\Big) - (\frac{\tau}{\theta})^{\beta} \\
&= n\log\beta - n\beta\log\theta + (\beta-1)\sum_{i=1}^{n}\log t_i - (\frac{\tau}{\theta})^{\beta}
\end{aligned}
$$

```
plpstan = '
functions{
  real nhpp_log(vector t, real beta, real theta, real tau){
    vector[num_elements(t)] loglik_part;
    real loglikelihood;
    for (i in 1:num_elements(t)){
      loglik_part[i] = log(beta) - beta*log(theta) + (beta - 1)*log(t[i]);
    }
    loglikelihood = sum(loglik_part) - (tau/theta)^beta;
    return loglikelihood;
  }
}
data {
  int<lower=0> n; //total # of obs
  real<lower=0> tau;//truncated time
  vector<lower=0>[n] t; //failure time
}
parameters{
  real<lower=0> beta;
  real<lower=0> theta;
}
model{
  t ~ nhpp(beta, theta, tau);
//PRIORS
beta ~ gamma(1, 1);
theta ~ gamma(1, 1);
}
'

fitplp <- stan(
  model_code=plpstan, model_name="NHPP", data=datstan,
  iter=2000,warmup = 1000, chains=1, seed = 123
)
```

```
## In file included from D:/R-3.4.1/library/BH/include/boost/config.hpp:39:0,
##                  from D:/R-3.4.1/library/BH/include/boost/math/tools/config.hpp:13,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/core/var.hpp:7,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/core/gevv_vvv_vari.hpp:5,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/core.hpp:12,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/mat.hpp:4,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math.hpp:4,
##                  from D:/R-3.4.1/library/StanHeaders/include/src/stan/model/model_header.hpp:4,
##                  from file23485b62379b.cpp:8:
## D:/R-3.4.1/library/BH/include/boost/config/compiler/gcc.hpp:186:0: warning: "BOOST_NO_CXX11_RVALUE_R
```

```
## #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##  ^
## <command-line>:0:0: note: this is the location of the previous definition
##
## SAMPLING FOR MODEL 'NHPP' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.097 seconds (Warm-up)
##                0.1 seconds (Sampling)
##                0.197 seconds (Total)
```

```r
print(fitplp)
```

```
## Inference for Stan model: NHPP.
## 1 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=1000.
##
##         mean se_mean   sd   2.5%    25%    50%   75% 97.5% n_eff Rhat
## beta    0.77    0.02 0.19   0.43   0.64   0.75  0.89  1.19   152    1
## theta   0.45    0.02 0.32   0.04   0.21   0.37  0.61  1.27   177    1
## lp__  -10.43    0.12 1.29 -13.84 -10.77 -10.02 -9.61 -9.33   121    1
##
## Samples were drawn using NUTS(diag_e) at Tue Jan 29 00:55:46 2019.
## 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).
```

Compared to MLE estimate of $\hat{\beta} = 0.678, \hat{\theta} = 0.217$.

### Weakly informative priors for beta and theta - GAMMMA(1, 5)

```r
plpstan15 = '
functions{
  real nhpp_log(vector t, real beta, real theta, real tau){
    vector[num_elements(t)] loglik_part;
    real loglikelihood;
    for (i in 1:num_elements(t)){
```

```
      loglik_part[i] = log(beta) - beta*log(theta) + (beta - 1)*log(t[i]);
    }
    loglikelihood = sum(loglik_part) - (tau/theta)^beta;
    return loglikelihood;
  }
}
data {
  int<lower=0> n; //total # of obs
  real<lower=0> tau;//truncated time
  vector<lower=0>[n] t; //failure time
}
parameters{
  real<lower=0> beta;
  real<lower=0> theta;
}
model{
  t ~ nhpp(beta, theta, tau);
//PRIORS
beta ~ gamma(1, 5);
theta ~ gamma(1, 5);
}
'

fitplp15 <- stan(
  model_code=plpstan15, model_name="NHPP15", data=datstan,
  iter=2000,warmup = 1000, chains=1, seed = 123
)
```

```
## In file included from D:/R-3.4.1/library/BH/include/boost/config.hpp:39:0,
##                  from D:/R-3.4.1/library/BH/include/boost/math/tools/config.hpp:13,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/core/var.hpp:7,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/core/gevv_vvv_vari.hpp:5,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/core.hpp:12,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math/rev/mat.hpp:4,
##                  from D:/R-3.4.1/library/StanHeaders/include/stan/math.hpp:4,
##                  from D:/R-3.4.1/library/StanHeaders/include/src/stan/model/model_header.hpp:4,
##                  from file2348779c33d2.cpp:8:
## D:/R-3.4.1/library/BH/include/boost/config/compiler/gcc.hpp:186:0: warning: "BOOST_NO_CXX11_RVALUE_RE
##  #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##  ^
## <command-line>:0:0: note: this is the location of the previous definition
##
## SAMPLING FOR MODEL 'NHPP15' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.107 seconds (Warm-up)
##                0.08 seconds (Sampling)
##                0.187 seconds (Total)
```

```r
print(fitplp15)
```

```
## Inference for Stan model: NHPP15.
## 1 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=1000.
##
##         mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## beta    0.57    0.01 0.12   0.37   0.49   0.56   0.65   0.80   267    1
## theta   0.15    0.01 0.12   0.01   0.06   0.13   0.22   0.46   309    1
## lp__  -14.13    0.06 1.05 -16.89 -14.61 -13.82 -13.37 -13.06   289    1
##
## Samples were drawn using NUTS(diag_e) at Tue Jan 29 00:58:15 2019.
## 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).
```

Compared to MLE estimate of $\hat{\beta} = 0.678, \hat{\theta} = 0.217$.