

Gamma with fix shape(k)

2022-10-13

Simulate Data

Hazard rate for Lomax(pareto 2) distribution is $S(x) = \frac{k}{s+x}$, where s is a vector of scale parameters and k is a vector of shape parameters.

```
knitr::opts_chunk$set(echo = TRUE)
#install.packages("truncdist")
library(survival)
library(truncdist)
```

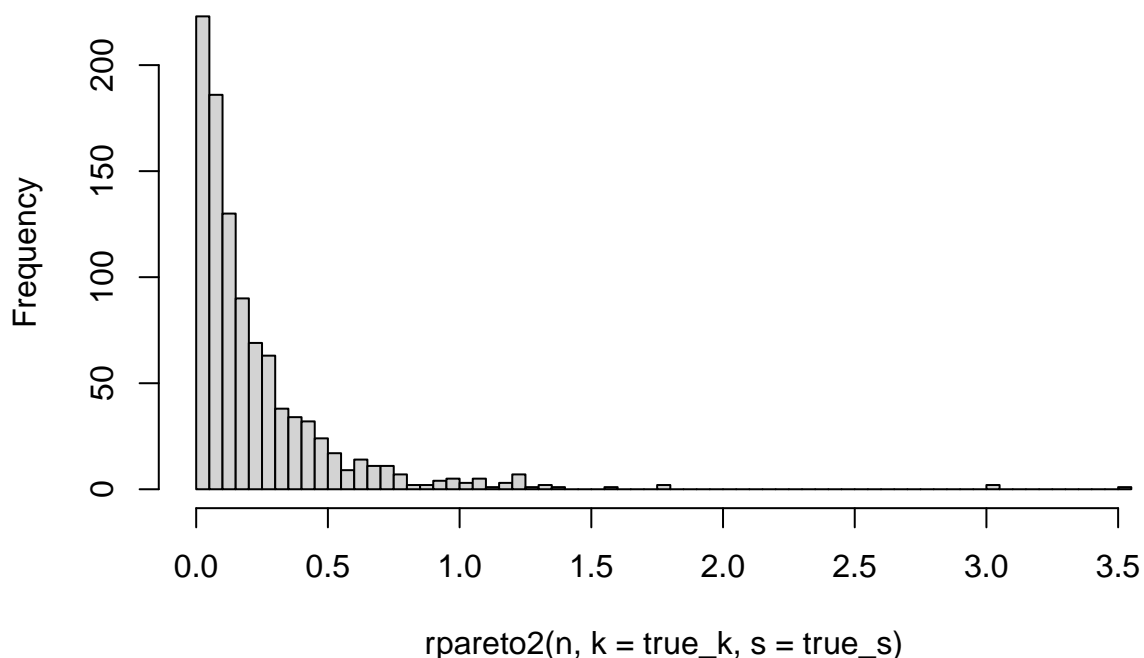
```
## Loading required package: stats4
```

```
## Loading required package: evd
```

```
set.seed(3)
#install.packages("agop")
# type-II Pareto is the true distribution (the one that generate data)
library("agop")

n <- 1000
A <- rbinom(n, 1, .5)
X <- model.matrix(~ A)
true_beta <- matrix(c(-2, .1), ncol=1)
true_mu <- X %*% true_beta
true_s = .7
true_k <- exp(-1*true_mu*true_s)
hist(rpareto2(n, k=true_k, s=true_s), breaks=100)
```

Histogram of rpareto2(n, k = true_k, s = true_s)



```
# simulate censoring and survival times
survt = rpareto2(n, k=true_k, s = true_s)
cent = rpareto2(n, k=true_k, s = true_s)

## observed data:
#censoring indicator
delta <- cent < survt
survt[delta==1] <- cent[delta==1] # censor survival time.

# survt_all will combine observed and imputed survival times.
survt_all <- survt

# count number of missing/censored survival times
n_miss <- sum(delta)
row_miss <- c(1:n)[delta] # index for which rows are censored
```

Functions

```
log_post_beta <- function(beta, log_s, X, survt){ # shape
  s <- exp(log_s)
  mu <- X %*% beta
  k <- exp(-1*mu*s)

  lik <- sum(dgamma(survt, shape = k, scale = s, log = T))
  pr <- dexp(x = s, rate = 1, log = T)
  return(lik + pr)
```

```

}
log_post_s <- function(log_s, beta, X, survt){ # scale
  s <- exp(log_s)

  mu <- X %*% beta
  k <- exp(-1*mu*s)

  lik <- sum(dgamma(survt, shape = k, scale = s, log = T))
  pr <- dexp(x = s, rate = 1, log = T)
  return(lik + pr)
}
metrop_hastings<-function(x_0, iter=1, log_post_density,
                           proposal_dist = function(x, prop_sigma){
                             MASS::mvrnorm(1, mu = x, Sigma = prop_sigma )
                           },
                           lower=-Inf, upper=Inf, prop_sigma,
                           ... ){
  for(i in 1:iter){
    # draw from proposal distribution
    x_star <- proposal_dist(x_0, prop_sigma)

    # calculate ratio of conditional posterior densities
    r_num <- do.call(log_post_density, c(list(x_star), list(...)) )
    r_denom <- do.call(log_post_density, c(list(x_0), list(...)) )
    r <- exp(r_num - r_denom)
    rmin<-min(r,1)
    if(is.na(rmin)) next
    # accept / reject proposal
    if(rbinom(1,1,rmin)==1){
      x_0<-x_star
    }
  }
}

res<-list(x_0 = x_0, accept_prob = rmin )
return(res)
}
rgamma_trunc <- function(n, a, k, s) {

  # range is a vector of two values

  F.a <- pgamma(a, shape=k, scale = s)
  F.b <- 1

  u <- runif(n, min = F.a, max = F.b)

  qgamma(u, shape=k, scale=s)
}

```

Run Augmented Sampler Accounting for Censoring

```
iter <- 10000 # number of gibbs iterations
burnin <- 9000 # burn-in iterations

# shells for storing parameters
hazard_ratio <- numeric(iter - burnin)

# initial values
beta_shell <- matrix(c(0,0), ncol=1)
l_s_shell <- c(0)

prop_covar <- diag(c(.01,.01))
# plot stratified Kaplan-Meier
par(mfrow=c(1,1))
plot(survfit(Surv(survt, 1-delta) ~ A), conf.int = F, col=c('blue','red'),
      xlab=c('Time'), ylab='Survival Probability',
      main = 'Data augmentation with all subjects')
for(i in 2:iter){
  ## sample from posterior of parameters,
  ## conditional on observed and missing survival times

  # metrop_hastings() is a custom function for generating a draw
  # from conditional posterior of beta: log_post_beta
  beta_shell <- metrop_hastings(x_0 = beta_shell,
                               iter = 1,
                               log_post_density = log_post_beta,
                               prop_sigma = prop_covar,
                               X=X, survt=survt_all,
                               log_s=l_s_shell )$x_0

  # sample from conditional posterior of s: log_post_s
  l_s_shell <- metrop_hastings(x_0 = l_s_shell,
                              iter = 1,
                              log_post_density = log_post_s,
                              prop_sigma = matrix(.002),
                              X=X, survt=survt_all,
                              beta=beta_shell)$x_0

  ## sample from conditional posterior of missing survival times
  mu_curr <- X %>% beta_shell
  s_curr <- exp(l_s_shell)

  for(m in row_miss){
    k_curr <- exp(-1*mu_curr[m]*s_curr)

    survt_all[m] <- rgamma_trunc(1, survt[m], k_curr, s_curr)
  }

  if(i>burnin){
    # plot 500 posterior survival curve draws for treated and placebo
    mu_trt <- sum(beta_shell)
```

```

mu_pbo <- beta_shell[1]

post_draw <- rgamma(n, shape = exp(-1*mu_trt*s_curr), scale = s_curr )
post_ecdf <- ecdf(post_draw)
curve(1-post_ecdf(x), add=T, from=0, to=1.2, col='lightblue')

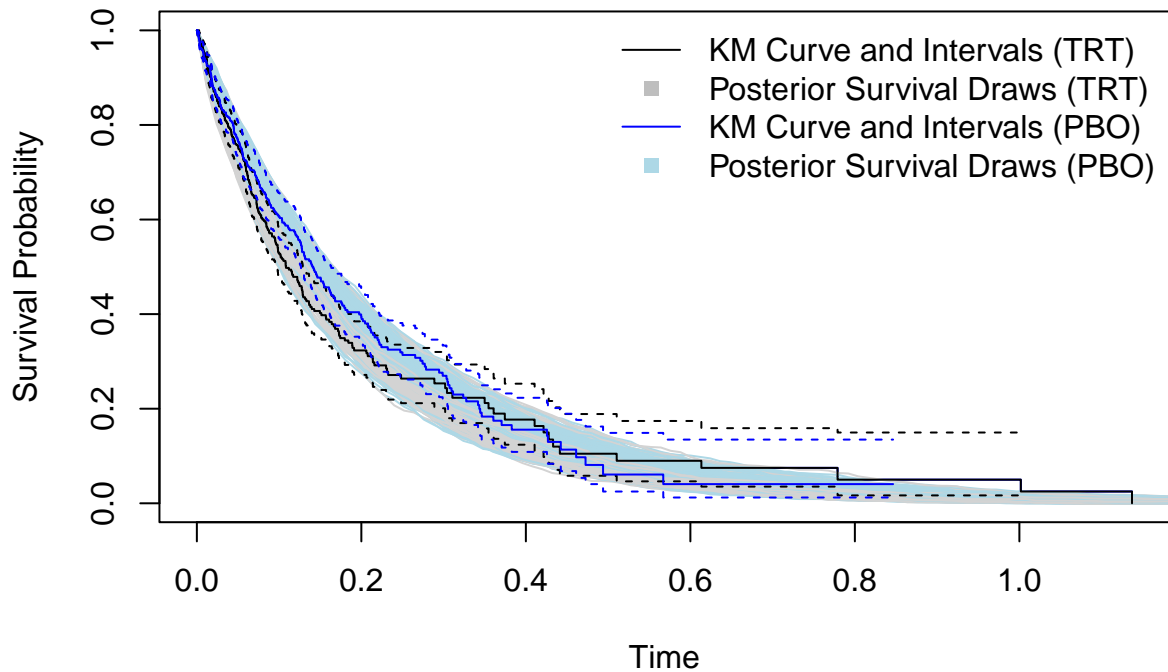
post_draw <- rgamma(n, shape = exp(-1*mu_pbo*s_curr), scale = s_curr )
post_ecdf <- ecdf(post_draw)
curve(1-post_ecdf(x), add=T, from=0, to=1.2, col='lightgray')

# store hazard ratio
hazard_ratio[i-burnin] <- exp(-beta_shell[2]*s_curr)
}

}
# overlay KM curve and plot legend
lines(survfit(Surv(survtime, 1-delta) ~ A), conf.int = T, col=c('black','blue'))
legend('topright',
      legend = c('KM Curve and Intervals (TRT)',
                  'Posterior Survival Draws (TRT)',
                  'KM Curve and Intervals (PBO)',
                  'Posterior Survival Draws (PBO)'),
      col=c('black','gray','blue','lightblue'),
      lty=c(1,0,1,0), pch=c(NA,15,NA,15), bty='n')

```

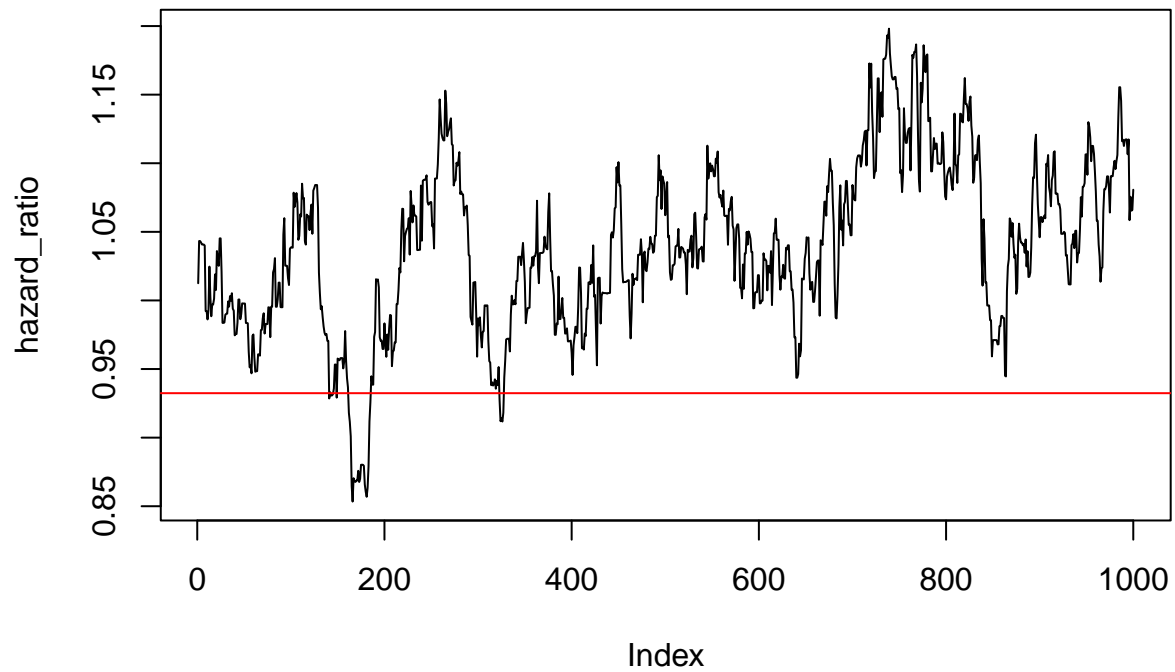
Data augmentation with all subjects



```

plot(hazard_ratio, type='l')
abline(h=exp(-true_beta[2]*true_s), col='red')

```



```
hist(hazard_ratio)
abline(v=exp(-true_beta[2]*true_s), col='red')
```

Histogram of hazard_ratio

