Bayesian confidence interval exercise

1) The data in `MTH396_CI_assignment_data1.csv` is from a gamma distribution.

- Use uniform priors on $(0, 8)$ for α and σ
- Find point estimates for the parameters α and σ
- Find a 95% confidence interval for α
- Find a 95% confidence interval for σ
- Find a 95% confidence interval for the median of the distribution
- Find a 95% confidence interval for the 90^{th} percentile of the distribution

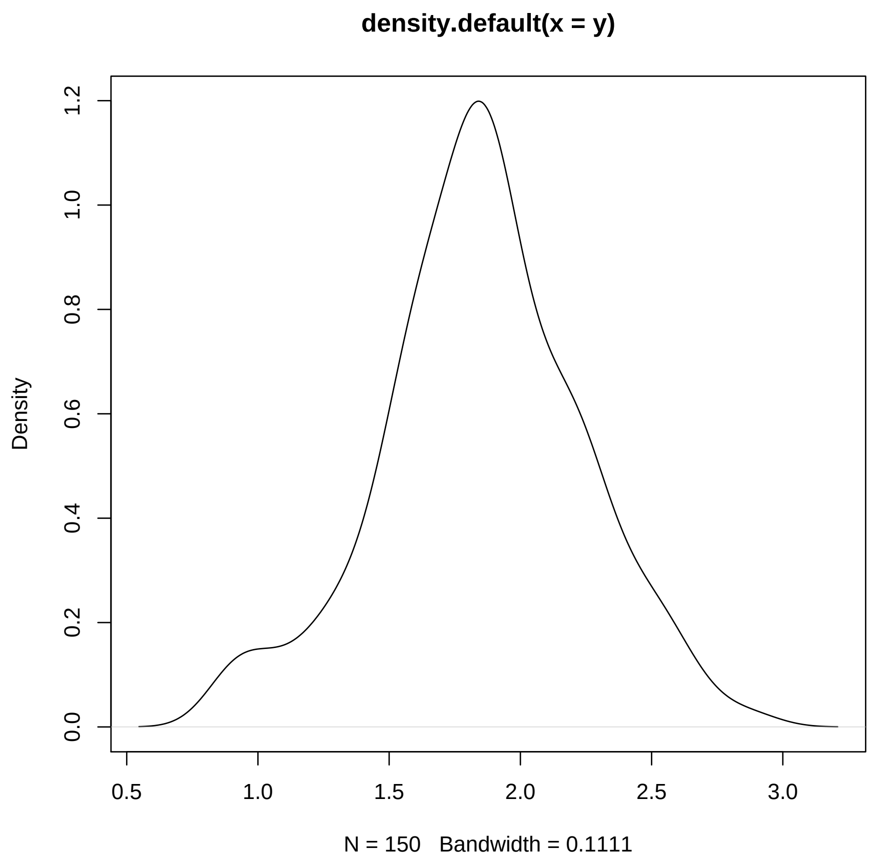
Weibull

```
library(rstan)

## Loading required package:  ggplot2
## Loading required package:  StanHeaders
## rstan (Version 2.14.2, packaged:  2017-03-19 00:42:29 UTC, GitRev:
5fa1e80eb817)
## For execution on a local, multicore CPU with excess RAM we recommend
calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())

rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
set.seed(301)

N<-150
y <- rweibull(N,5,2)
df<-data.frame(y)
write.csv(df,file='MTH396_CI_assignment_data1.csv')
plot(density(y))
```




```
stanfit<-stan("Weibull.stan")

print(stanfit)

## Inference for Stan model: Weibull.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean   sd   2.5%    25%   50%    75%  97.5% n_eff Rhat
## alpha    5.41    0.01 0.33   4.77   5.18   5.4   5.64   6.05  2859    1
## sigma    2.00    0.00 0.03   1.94   1.98   2.0   2.03   2.06  2790    1
## lp__   -67.08    0.02 0.94 -69.48 -67.48 -66.8 -66.40 -66.14  1849    1
##
## Samples were drawn using NUTS(diag_e) at Fri Mar 24 18:05:31 2017.
## 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).
```

```r
pd <- extract(stanfit)
str(pd)
```

```
## List of 3
##  $ alpha: num [1:4000(1d)] 5.34 5.63 5.79 5.69 5.92 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL
##  $ sigma: num [1:4000(1d)] 2.01 2.07 2 1.99 2.02 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL
##  $ lp__ : num [1:4000(1d)] -66.2 -68.5 -66.8 -66.7 -67.1 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL
```

```r
q50<-qweibull(.5,pd$alpha,pd$sigma)
quantile(q50,c(.025,.5,.975))
```

```
##     2.5%      50%    97.5%
## 1.808079 1.872339 1.934061
```

```r
q90<-qweibull(.9,pd$alpha,pd$sigma)
quantile(q90,c(.025,.5,.975))
```

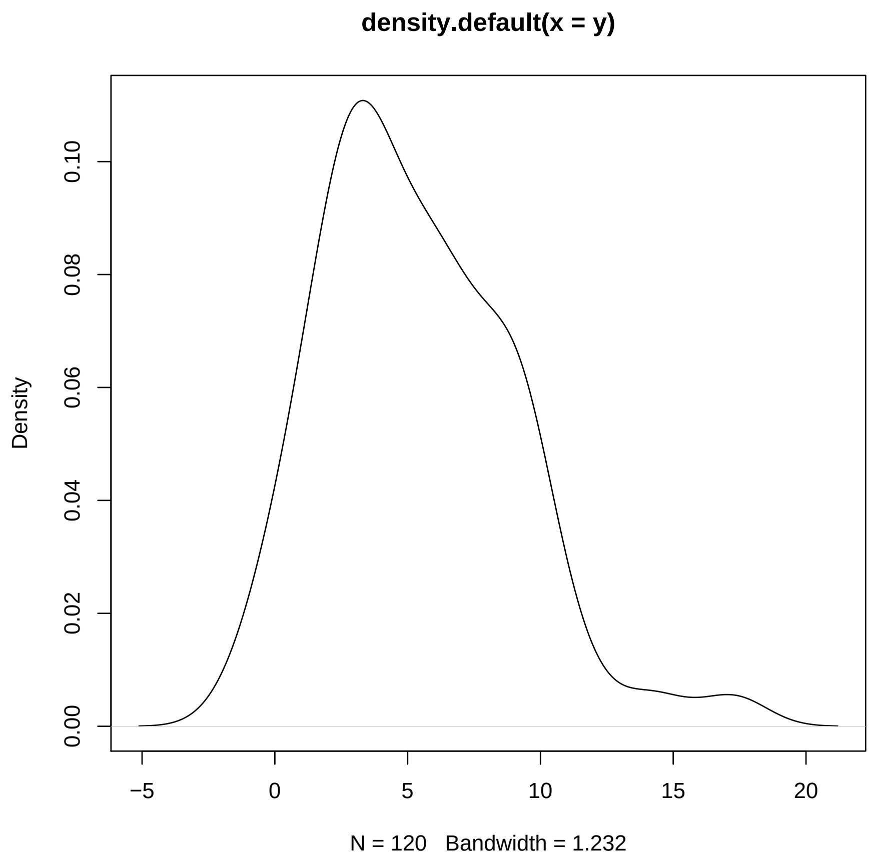
```
##     2.5%      50%    97.5%
## 2.271771 2.338535 2.414144
```

2) The data in `MTH396_CI_assigmnemt_data2.csv` is from a logistic distribution.

- Use a normal(0,20) prior for μ
- Use a uniform prior on $(0, 10)$ for σ
- Find point estimates for the parameters μ and σ
- Find a 95% confidence interval for μ
- Find a 95% confidence interval for σ
- Find a 95% confidence interval for the median of the distribution
- Find a 95% confidence interval for the 25^{th} percentile of the distribution

logistic

```r
N<-120
y <- rlogis(N,5,2)
df<-data.frame(y)
write.csv(df,file='MTH396_CI_assignment_data2.csv')
plot(density(y))
```




```r
stanfit<-stan("logistic.stan")

print(stanfit)

## Inference for Stan model: logistic.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff
## mu        5.13    0.01 0.33   4.48   4.90   5.13   5.35   5.76  3326
```

```
## sigma    2.07    0.00 0.16    1.77    1.95    2.06    2.17    2.40  3221
## lp__  -323.87    0.02 0.98 -326.40 -324.29 -323.58 -323.16 -322.88  1777
##       Rhat
## mu       1
## sigma    1
## lp__     1
##
## Samples were drawn using NUTS(diag_e) at Fri Mar 24 18:05:35 2017.
## 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).
```

```
pd <- extract(stanfit)
str(pd)
```

```
## List of 3
##  $ mu   : num [1:4000(1d)] 4.82 4.91 4.81 5.41 5.16 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL
##  $ sigma: num [1:4000(1d)] 2.26 2.47 2.46 2.27 2 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL
##  $ lp__ : num [1:4000(1d)] -324 -326 -326 -324 -323 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL
```

```
q50<-qlogis(.5,pd$mu,pd$sigma)
quantile(q50,c(.025,.5,.975))
```

```
##     2.5%      50%    97.5%
## 4.479289 5.125708 5.760965
```

```
q25<-qlogis(.25,pd$mu,pd$sigma)
quantile(q25,c(.025,.5,.975))
```

```
##     2.5%      50%    97.5%
## 2.130391 2.864487 3.542688
```